AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/1

Łukasz Jopek*, Laurent Babout**, Marcin Janaszewski***, Michał Postolski*

A New Method to Segment X-ray Tomography Images of Lamellar Microstructure in Titanium Alloys Based on Gray Level Gradient

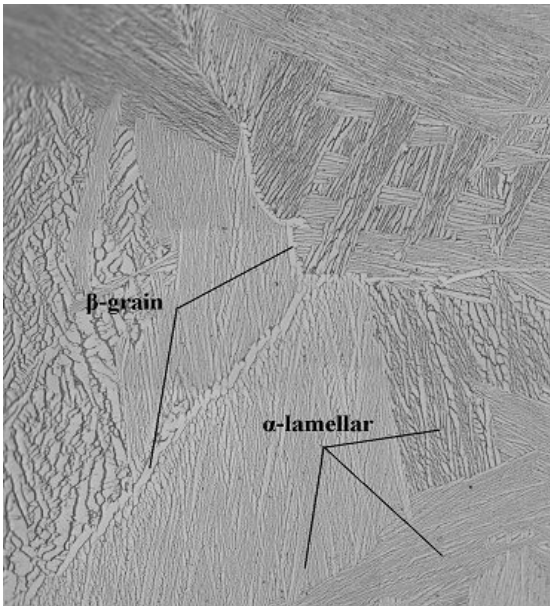
1. Introduction

Titanium alloys are widely used in many industrial applications such as in power generation, aeronautical, and biomedical industry because of excellent mechanical and corrosion properties combined with a relatively low density. The mechanical properties of these alloys are strongly dependent on variations in the microstructure. One of important aspect is the understanding of how crack is growing in the material and correlate this to the microstructure, because propagation direction is probably dependent on space organization of two types of microstructural features in such alloy when the microstructure if fully lamellar [1, 3]: β grain Boundary and α-lamellae. α-lamellar colonies can be described as group of lines (or parallel planes in 3D), which have a direction orientation. The presence of β grain Boundary is revealed by the growth of so-called α-layer along the previous mentioned boundary and corresponds in X-ray tomography images to single, large, dark surfaces (while in microscope image β grain Boundaries correspond to large bright lines) (see Fig. 1). From an image processing point of view, the task to segment the different lamellar colonies in x-ray tomography images is preferably based on local statistical properties of pixel intensities, because this type of lamellar microstructure is a good example of textured image. Texture images can be described by more features than normal images. For example, in texture images, each pixel (or voxel in 3D images) can be characterized by properties such as smoothness, direction orientation, coarseness and regularity of patterns. Applications such as MaZda software [13] proposes more than 300 different features that can be used to characterise/classify patterns. Classical segmentation methods based on intensity

* Computer Engineering Department, Technical University of Lodz, Poland, Division of Expert Systems & Artificial Intelligence, The College of Computer Science, Poland
** Computer Engineering Department, Technical University of Lodz
*** Computer Engineering Department, Technical University of Lodz, Poland, Division of Expert Systems & Artificial Intelligence, The College of Computer Science, Poland

histogram cannot be used to segment different α-lamellae colonies, because it does not recognize directionality of colonies. Previous attempts of segmentation of these images concentrated on two methods: wavelet decomposition and co-occurrence matrix.

a)



b)

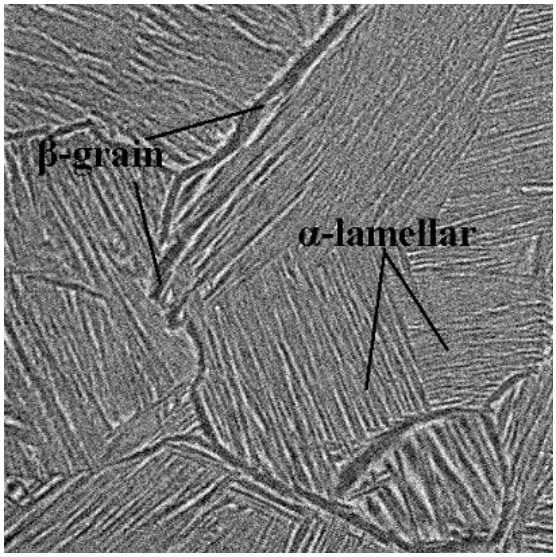


Fig. 1. Example of microstructure of fully lamellar titanium alloys:
a) obtained from optical microscope; b) obtained from X-ray tomography

Methods which are based on wavelet decomposition [2, 5] gave promising results for images obtained from optical microscope. However the method worked satisfyingly to separate only colonies about vertical, horizontal directions, and oblique directions . In real, this is not enough and colonies can present different orientations and a minimum of 6 different directions should be achieved for the classification. Moreover, for images obtained from x-ray tomography, results are less satisfying than for optical images. Some areas were uncorrectly assign to these 3 classes. One of the main reason is the level of noise of the 3D image, and irregularities in á-lamellae structure. Details about this method is included in [2]. The second tested method is based on co-occurrence matrix. The method is sensitive to rotation and particularly sensitive to regular patterns. Co-occurrence matrix can be created for 4 directions : 0, 45, 90 and 135 degrees, so theoretically should detect regular patterns in these directions. From this matrix 11 features [4, 13] can be calculated. Using feature selection algorithms [6] a feature 'contrast' was chosen, because it has a high discrimination power factor. In practice, the four directions were properly detected for artificial images that simulated lamellar microstructure. However, for real images a lot of α-lamellae colonies were incorrectly identified. In some cases, single colony was sub-divided into few others. The main reason is similar to the one of the previous approach: noise and irregularities in the pattern. The high noise level made impossible to correctly recognize and assign pixels to the appropriate class. Not only noise but also small contrast reducing the perception of lamellar colonies altered the classification.

Using the gradient of the image is an alternative option. The gradient of the image is sensitive to the local contrast, and allows to detect local orientation. In the method [5] they have used the gradient to detect local orientation structures similar to α-lamellae. For each point a local matrix of orientation is calculated (co-variance matrix). From this matrix principal component analysis is applied to calculate eigenvectors and eigenvalues. The local orientation is given by the angle between the eigenvector corresponding to the largest eigenvalue and a reference vector. The authors also use standard watershed algorithm [9] for segmentation. The proposed methodology was tested on our set of 3D data, but results weren't satisfactory, again because of the high level of noise. This paper presents a new approach to segment heavily 3D textured images such as the one of lamellar titanium alloys obtained from X-ray tomography. The presented method considers gradient from gray-level value and using it for segmentation textures, using the method of the pattern recognition, as cluterization and classification. The paper is organised as follows. Section 2 presents the algorithm of the presented method. In section 3, results are presented and discussed. Conclusion which presents a list of perspective works closes the paper.

2. Algorithm

2.1. Local image orientation based on its gradient

An image is defined as a function A:{0, 1, ..., N}×{0, 1, ..., M} –> {0, 1, ..., 255}, *which arguments represent spatial coordinates of a pixel and its value represents grey level of the*

pixel. The gradient from gray-level value is being calculated for each pixel. The components of the gradient image in point (x, y) of image A are estimated by

$$\frac{\partial A(x, y)}{x} = \big(A(x-1, y) - A(x+1, y)\big)/2h \tag{1}$$

$$\frac{\partial A(x, y)}{y} = \big(A(x, y-1) - A(x, y+1)\big)/2h \tag{2}$$

where h – Step, In the case $h = 1$.

For each pixel is being calculated gradient from the first derivative to x and y direction. Local orientation for single pixel is given by:

$$D(x, y) = \tan^{-1}\left(\frac{\dfrac{\partial A(x, y)}{x}}{\dfrac{\partial A(x, y)}{y}}\right) \tag{3}$$

where D – Output image.

a) b)

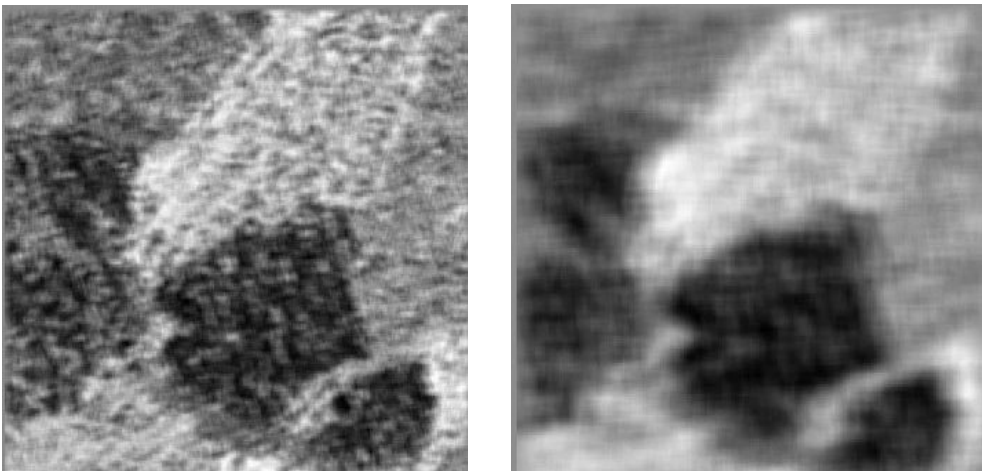


Fig. 2. Results of features extraction from image (Fig. 1b):
a) window size = 11×11; b) window size = 35×35

Local orientation can also be estimated by means of neighboring pixels (e.g. a square in R^2 or a cube in R^3). Size of the neighborhoods of the pixel ($W \times W$ *or* $W \times W \times W$) is an important parameter. Large values (above 25×25) (see Fig. 2a) are reducing the influence of

the noise, but, in the order hand, they are reducing the accuracy of boundary detection between different α-lamellae colonies. Lower values (below 25×25) (see Fig. 2b) are increasing the sensitivity of the algorithm for noise, but the accuracy of boundary is more pronounced. Mean values can be calculated by:

$$E(x, y) = \frac{1}{(r+1)^2} \sum_{i=x-r}^{x+r} \sum_{j=y-r}^{y+r} D(i, j) \tag{4}$$

where $(r+1)^2$ – Size of window (in pixel).

2.2. Local orientation of 3D image based on its gradient

The method from previous paragraph can be extend to 3D images. The components of the gradient image in point (x, y, z) of image A are estimated by:

$$\frac{\partial A(x, y, z)}{x} = \big(A(x-1, y, z) - A(x+1, y, z)\big)/2h \tag{5}$$

$$\frac{\partial A(x, y, z)}{y} = \big(A(x, y-1, z) - A(x, y+1, z)\big)/2h \tag{6}$$

$$\frac{\partial A(x, y, z)}{x} = \big(A(x, y, z-1) - A(x, y, z+1)\big)/2h \tag{7}$$

where h – Step = 1.

Local orientation for a pixel is given by:

$$D_{xy}(x, y, z) = \tan^{-1}\left(\frac{\dfrac{\partial A(x, y, z)}{y}}{\dfrac{\partial A(x, y, z)}{x}}\right) \tag{8}$$

$$D_{yz}(x, y, z) = \tan^{-1}\left(\frac{\dfrac{\partial A(x, y, z)}{z}}{\dfrac{\partial A(x, y, z)}{y}}\right) \tag{9}$$

$$D_{xz}(x, y, z) = \tan^{-1}\left(\frac{\dfrac{\partial A(x, y, z)}{z}}{\dfrac{\partial A(x, y, z)}{x}}\right) \tag{10}$$

The output is a set of 3 volumes of the same size as the input volume and the same voxel position in the different volumes is described by three values:

$$D_{xyz}(x, y, z) = \left(D_{xy}(x, y, z), D_{yz}(x, y, z), D_{xz}(x, y, z)\right) \tag{11}$$

Results for 3 different planes intersecting a given voxel are presented in Figure 3. These three values allow for assigning voxel to the appropriate class. Properties of each class can be fund using the cluterization algorithm described in the next sub-section.

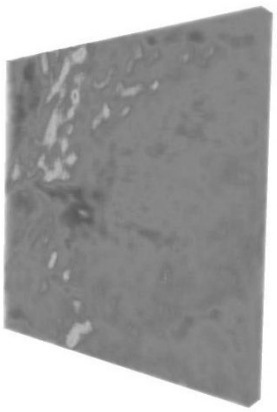


Fig. 3. Example features extraction from *XY* plane in 3D

2.3. Clustering data and classification

Clustering algorithm

Every sample of material can contain a certain number of α-lamellae colonies that are, for a geometrical point of view, defined by their interspacing and orientation. Before analysis of the sample, the number and properties are α-lamellae colonies are unknown. Important computing task is to automatically find the number of possible classes and their respective geometrical properties. It can be done using a clustering method. K-means [8] is one of the simplest and unsupervised learning algorithms that solve the clustering problem. The main idea is to define K-centroids, one for each cluster. The number of custer K is set before the algorithm execution. This algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is defined as follows :

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2 \tag{12}$$

where $\left\| x_i^j - c_j \right\|^2$ – is a chosen distance measure between a data point and the cluster center.

The algorithm is composed of the following steps:

1. *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroid (the place in the space of solutions can be chosen randomly).*
2. *Assign each object to the group that has the closest centroid (using the chosen measure of the distance).*
3. *Recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated. And also maximum number of iterations can be a stop condition.*

Classification algorithm

Image segmentation is a division of image into areas, homogeneous in terms of selected criteria. In this article, the image segmentation is based on assigning image points (pixel or voxel) to the respective classes with a predefined set. The assignment to the current class is done using k-NN (k-nearest neighbor algorithm) classifier. K-NN classifier is an instance-based learning algorithm that is based on a distance function for pairs of observations, such as the Euclidean distance. Training data are computed first. The object to training data is being chosen randomly. The number of objects is set before starting the algorithm. The object consists of three values – values which they are describing each voxel. In this solution, training data are computed by K-means algorithm.

Texture segmentation algorithm

The algorithm is composed of the following steps:

1. *Initialization: set:*
   – *Size of window (see paragraph 2.)*
   – *Number of object in training data*
   – *Number of clusters (for k-means algorithm)*
   – *Number of maximum iterations (for k-means algorithm)*
   – *Number of nearest neighbor (for k-NN algorithm)*
2. *Read image (raw data, 8 bit color depth.)*
3. *Extract image features (Local orientation, see paragraph 2.2), every voxel is being described with three values. Output are three images with calculated local orientation.*
4. *Randomly choose object for training data (from images used in step 3.)*
5. *Run k-means algorithm*
6. *Run K-NN algorithm for each voxel from image used in step 3, with training data obtained at step 5.*

Figure 4 shows the diagram of the algorithm for texture segmentation applied to the set of images corresponding to fully lamellar titanium alloys microstructure.

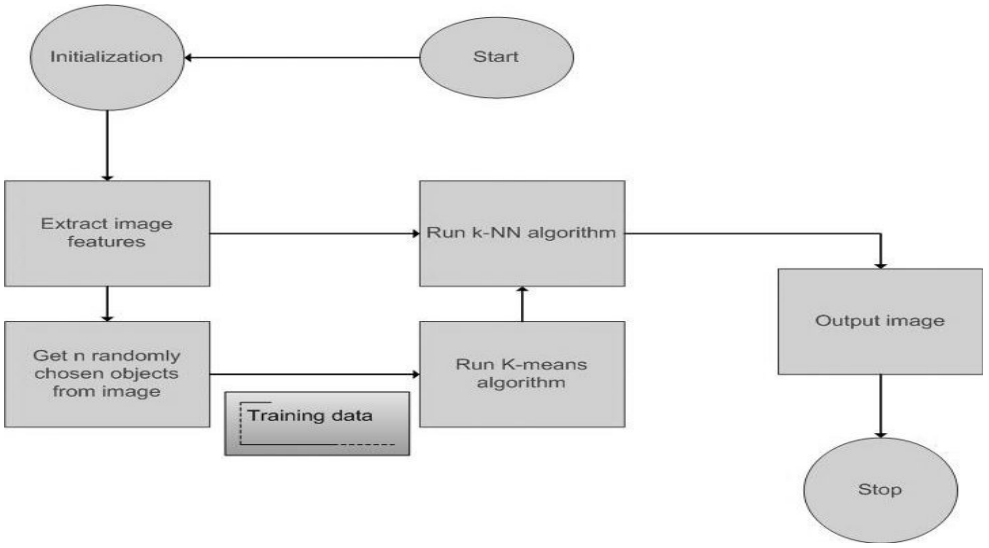


Fig. 4. Algorithm diagram for the texture segmentation of lamellar colonies in titanium-based alloys 2D/3D images

3. Results and discussion

Figure 5b displays the result of the above mentioned method to the image presented in Figure 5a. It is 2D image. Algorithm found 5 different classes. One can see that some large areas (from *A* to *D*) of classes were correctly detected. Areas from *E* to *G* are not correctly detected, but, in this case, a major problem is correct detecting class bounduares. The remaining parameters are summarized in Table 1.

Despite this known current drawback that will be addressed in the near future, the window size of 25×25 produces the best compromise between reducing the influence of the noise and the accuracy of manual boundary detection between different α-lamellae colonies.

Figure 6b displays the result of the above mentioned method to the image presented in Figure 6a. The size of the original image is 501×500×50 with a window size of 9×9×9. The remaining parameters are summarized in Table 2 for other window sizes. In the case [2, 7] to detect it was only possible 3 directional classes – class: horizontal, vertical and without direction. Real amount different α-lamellae colony in a input image isn't affecting the detected number of classes. The presented method allows for detecting the real number of classes and their characteristics.

Figure 7b, d and f displays the result of the above method, for three plains: *XY*, *XZ* and *YZ* to the image presented in Figure 6a.

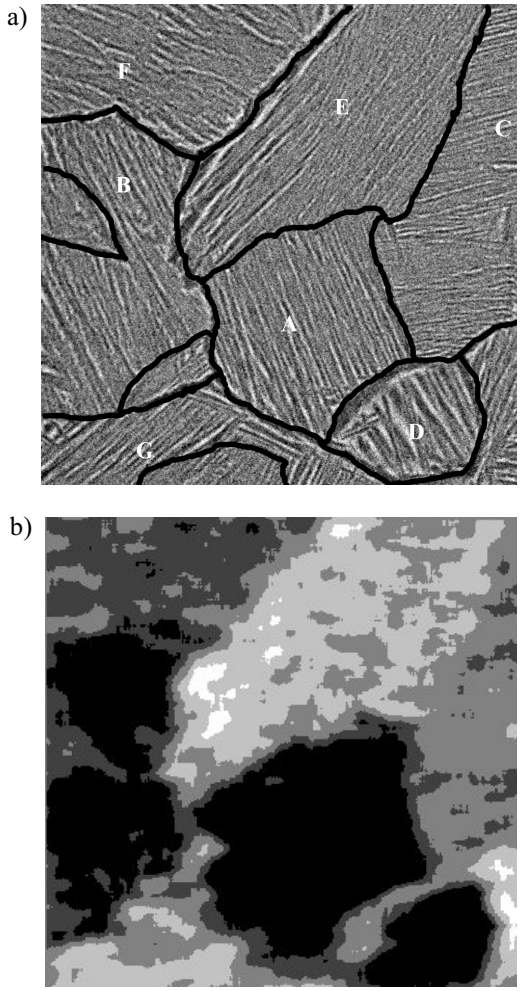


Fig. 5. Results in 2D (a) input volume (b) segmented volume

Table 1
Parameters value in use to segmented image from Figure 5a

Window size	Number of nearest neighbor (for k-NN algorithm)	Number of clusters (for k-means algorithm)	Number object in training data
15×15	5	7	300
25×25	5	10	300
35×35	5	10	300

a) b)

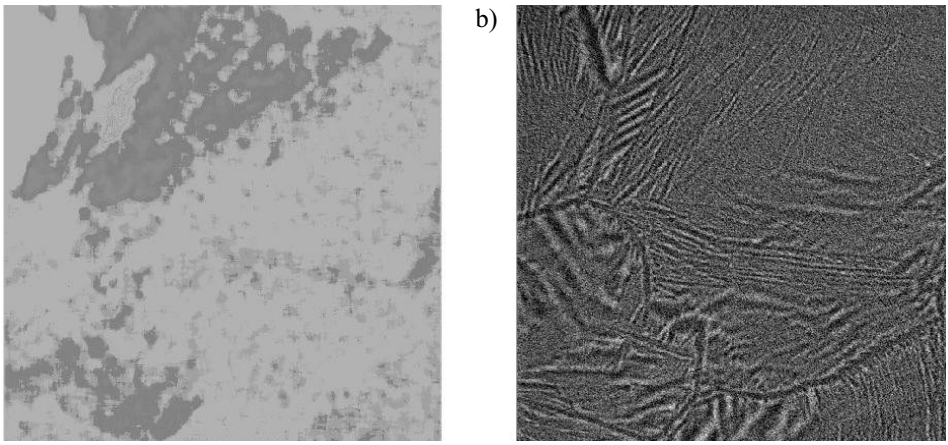


Fig. 6. Results in 3D (a) input volume (b) segmented volume

Table 2
Parameters value in use to segmented image from Figure 6a

Window size	Number of nearest neighbor (for k-NN algorithm)	Number of clusters (for k-means algorithm)	Number object in training data
5×5×5	5	7	300
7×7×7	5	10	300
9×9×9	5	10	300

The black lines in images are manually set boundaries between different α-lamellae. Indeed, the current method is currently not able to automatically segment colonies based on the classification in such a ways that boundary between adjacent colonies obtained after segmentation are in agreement with the one that can be guessed from the original input image. Despite this known current drawback that will be addressed in the near future, the window size of 9×9×9 produces the best compromise between reducing the influence of the noise and the accuracy of manual boundary detection between different á-lamellae colonies. One can see that some large areas of classes were correctly detected like areas *A*, *B*, *C*, *D* from Figure 7b. In the other hand some areas of classes were no correctly detected, god example of this is *F*, *G* or *H* areas in Figure 6b. Some small regions are also detected. In Figure 7d we can see that areas *A*, *B*, *C* are aslo correctly detected. In the present test training data which containing 300 objects extracted randomly from the input 3D image. It is a good compromise between quality classification and time needed to process the image.

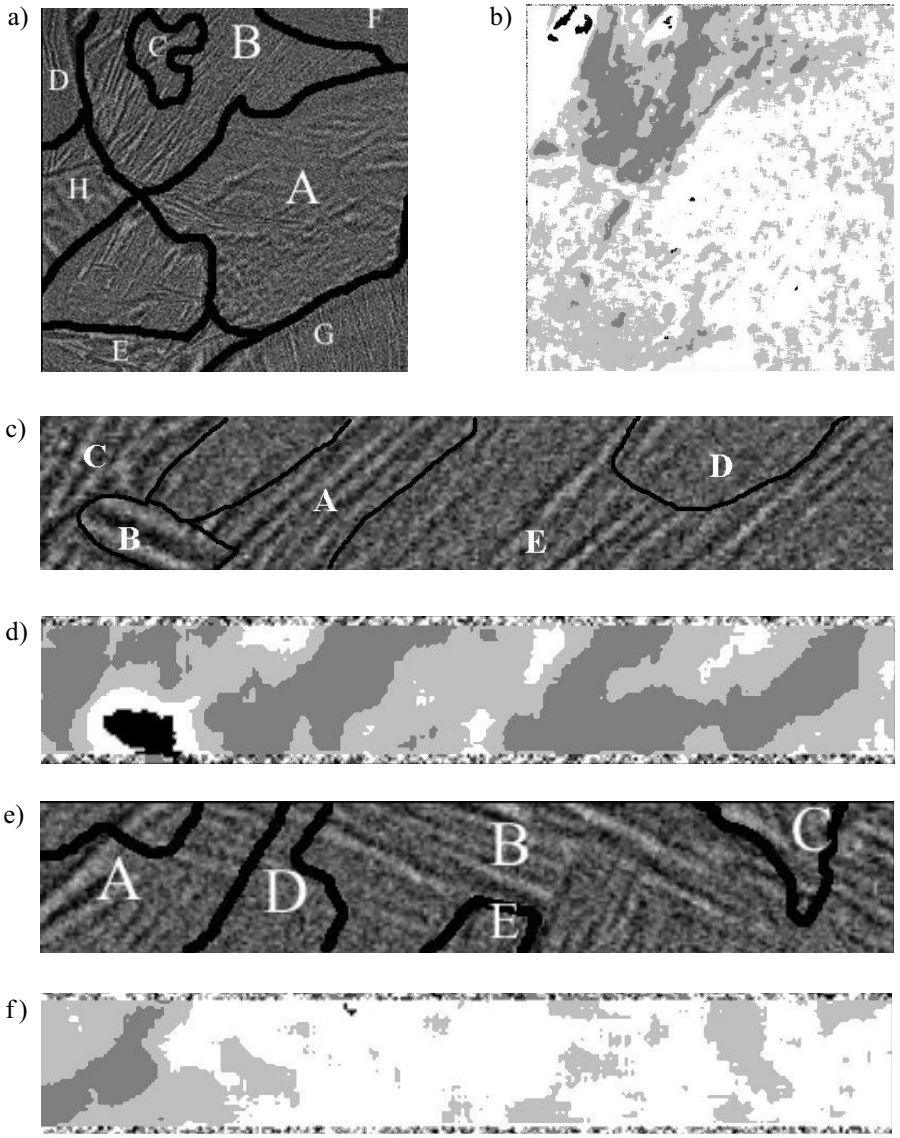


Fig. 7. The result 3D for three plains: (a–b) *XY*, (c–d) *XZ* and (e–f) *YZ*

Typical times is show in Table 3. Computer parameter: CPU 2Ghz, 2GB Ram, MS Windows 2003, a programming language is c++. This number of objects does not guarantee the right choice of the object from all classes to classify the local window zone with the proper colony orientation. On the other hand, large training data set (which contain over a flew thousand objects) extends drastically the computation time. There are two solutions

to this problem. The first solution is to select objects manually. Then every class can have the same number of representatives. However, this solution is rather impractical since a large number of objects needs to be selected for the training data and the supposed improvement of the results is biased by the time required to select the objects. The second solution consists in using reduction of reference set algorithm [10, 11, 12]. Such an algorithm reduces the training data by 10% of the original data, without changing the quality of the classification. In this case, the original training data can be large (for example 5000 objects). Then the plausibility of choosing objects from all classes is much greater.

Table 3
Number of objects in training data with time need to classify all voxel to class

Number of objects in training data	Processing time [in minutes]
100	40
200	80
300	120
500	400

Another problem of the current method resides in the choice of the algorithm for classification, i.e. k-NN classifier (k-nearest neighbor algorithm). The main disadvantage of this classifier is that the nearest neighbor number needs to be determined *a priori*. In the present case, it has been empirically set to 5. This value was selected from range 2–10. Every value, from 2, to 10, was checked, a value giving the best quality to classification was chosen. The long computation time is also a non negligible drawback of the current method. However, it is a problem shared by any proosed method for texture classification and segmentation. Nonetheless, it is possible, using current computing improvement, to process calculation using parallel computing. In that case, every voxel can be processed by separate processors (threads) such as GPU (graphics processor) using CUDA technology [15]. Another possibility is use to clustering and classify self organized artificial neural network. Good example can be Kohonen [14] self organized maps.

Despite the drawbacks liste above, we consider that the current method that uses the gradient of the brightness of the image for the segmentation of textures combined with classical methods of pattern recognition can achieve satisfying results in terms of colony segmentation in noisy-3D images of fully lamellar titanium alloys. Such a method can also provide with the easiness of adapting the algorithm for processing other tomographic images, where the local directional orientation of the texture is an important feature. The algorithm can be easily implement as the parallel algorithm, so the processing time can be short.

Acknowledgments

The authors would like to thank Dr J. Bennett of the School of Materials, University of Manchester, who scanned the titanium sample using the X-ray tomography setup of the ID19 beam line at the ESRF.

The authors Ł. Jopek and M. Postolski are scholarship holders of project entitled "Innovative education…" supported by European Social Fund.

References

[1] Babout, L., Marrow, T.J., Preuss M., *Sequential x-ray tomography studies of damage assessment in materials science.* 4th Symposium on Process Tomography in Poland, Warsaw, 2006.

[2] Babout L., Jopek Ł., Janaszewski M., Preuss M., Buffiere J-Y., *Towards the texture segmentation of X-ray tomography images of lamellar microstructure in titanium based alloys*. 5th International Symposium on Process Tomography in Poland, 2008.

[3] Birosca S., Buffiere J-Y., Garcia-Pastor F.A., Karadge M., Babout L., Preuss M., *Three-dimensional characterization of fatigue cracks in ti-6246 using x-ray tomography and electron backscatter diffraction.* Acta Materialia, 57, 2009, 5834–5847.

[4] Haralick R.M., S.L.G., *Computer and robot vision.* Addison-Wesley Publishing Co., Boston, 1992, 630.

[5] Jeulin D., Moreaud M., *Segmentation of 2D and 3D textures from estimates of the local orientation*. Image Anal Stereol. 2008.

[6] Jopek Ł., Nowotniak R., Postolski M., Babout L., Janaszewski M., *Zastosowanie kwantowych algorytmów genetycznych do selekcji cech.* Automatyka (półrocznik AGH), ISSN: 1429-3447, 2009, 1219–1232.

[7] Jopek Ł., Postolski M., Janaszewski M., Babout L., *Segmentacja obrazów stopów tytanu o strukturze płytowej.* Sympozjum Modelowanie i Symulacja Komputerowa w Technice, ISBN 978-83-60282-13-7, 2010, 59–64.

[8] Macqueen J.B., *Some Methods for Classification and Analysis of Multivariate Observations.* Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press., 1967, 281–297.

[9] Meyer F., *Un algorithme optimal pour la ligne de partage des eaux.* Dans 8me congrès de reconnaissance des formes et intelligence artificielle, vol. 2, Lyon, France, 1991, 847–857.

[10] Sierszeń A., *Kaskadowy algorytm redukcji zbioru odniesienia.* Automatyka (półrocznik AGH), ISSN: 1429–3447, 2009, 995–1008.

[11] Sierszeń A., *Methods of reference set condensation for decision rules based on distance functions.* Zeszyty Naukowe Politechniki Łódzkiej, seria: Elektryka (rocznik), nr 117, ISSN: 0374-4817, 2009, 45–53.

[12] Sierszeń A., *Reduction of reference set with the method of cutting hyperplanes*. Journal of Medical Informatics & Technologies, vol. 13, ISSN: 1642-6037, 2009, 215–220.

[13] Strzelecki M., Materka A., Szczypiński P., MaZda, *Texture Analysis for Magnetic Resonance Imaging*. Med4publishing, Prague, 2006, 105–111.

[14] Tadeusiewicz R., *Sieci neuronowe*. WNT, 1993.

[15] Webpage "CUDA ZONE", http://www.nvidia.pl/object/cuda_home_new_pl.html.