Available online at www.sciencedirect.com

ScienceDirect

Computer Physics Communications 177 (2007) 329–338

Computer Physics Communications

www.elsevier.com/locate/cpc

TRIAC II. A MatLab code for track measurements from SSNT detectors ☆

D.L. Patiris [a], K. Blekas [b], K.G. Ioannides [a,*]

[a] *Nuclear Physics Laboratory, Department of Physics, The University of Ioannina, 451 10 Ioannina, Greece*
[b] *Department of Computer Science, The University of Ioannina, 451 10 Ioannina, Greece*

Received 22 November 2006; received in revised form 30 March 2007; accepted 9 April 2007

Available online 24 April 2007

Abstract

A computer program named TRIAC II written in MATLAB and running with a friendly GUI has been developed for recognition and parameters measurements of particles' tracks from images of Solid State Nuclear Track Detectors. The program, using image analysis tools, counts the number of tracks and depending on the current working mode classifies them according to their radii (Mode I—circular tracks) or their axis (Mode II—elliptical tracks), their mean intensity value (brightness) and their orientation. Images of the detectors' surfaces are input to the code, which generates text files as output, including the number of counted tracks with the associated track parameters. Hough transform techniques are used for the estimation of the number of tracks and their parameters, providing results even in cases of overlapping tracks. Finally, it is possible for the user to obtain informative histograms as well as output files for each image and/or group of images.

Program summary

Title of program: TRIAC II
Catalogue identifier: ADZC_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/ADZC_v1_0
Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland
Computer: Pentium III, 600 MHz
Installations: MATLAB 7.0
Operating system under which the program has been tested: Windows XP
Programming language used: MATLAB
Memory required to execute with typical data: 256 MB
No. of bits in a word: 32
No. of processors used: one
Has the code been vectorized or parallelized?: no
No. of lines in distributed program, including test data, etc.: 25 964
No. of bytes in distributed program including test data, etc.: 4 354 510
Distribution format: tar.gz
Additional comments: This program requires the MatLab Statistical toolbox and the Image Processing Toolbox to be installed.
Nature of physical problem: Following the passage of a charged particle (protons and heavier) through a Solid State Nuclear Track Detector (SSNTD), a damage region is created, usually named latent track. After the chemical etching of the detectors in aqueous NaOH or KOH solutions, latent tracks can be sufficiently enlarged (with diameters of 1 µm or more) to become visible under an optical microscope. Using the appropriate apparatus, one can record images of the SSNTD's surface. The shapes of the particle's tracks are strongly dependent on their charge, energy and the angle of incidence. Generally, they have elliptical shapes and in the special case of vertical incidence, they are circular. The manual counting of tracks is a tedious and time-consuming task. An automatic system is needed to speed up the process and to increase the accuracy of the results.

☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: kioannid@cc.uoi.gr (K.G. Ioannides).

0010-4655/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2007.04.002

Method of solution: TRIAC II is based on a segmentation method that groups image pixels according to their intensity value (brightness) in a number of grey level groups. After the segmentation of pixels, the program recognizes and separates the track from the background, subsequently performing image morphology, where oversized objects or objects smaller than a threshold value are removed. Finally, using the appropriate Hough transform technique, the program counts the tracks, even those which overlap and classifies them according to their shape parameters and brightness.
Typical running time: The analysis of an image with a PC (Intel Pentium III processor running at 600 MHz) requires 2 to 10 minutes, depending on the number of observed tracks and the digital resolution of the image.
Unusual features of the program: This program has been tested with images of CR-39 detectors exposed to alpha particles. Also, in low contrast images with few or small tracks, background pixels can be recognized as track pixels. To avoid this problem the brightness of the background pixels should be sufficiently higher than that of the track pixels.

PACS: 29.40.Wk; 29.40.-n; 29.85.+c; 92.20.Td; 93.85.Np; 29.30.Ep; 29.90.+r; 87.53.Pb; 87.53.Qc

Keywords: Solid state nuclear tracks detectors; Automatic track counting; Image analysis; Alpha-particle radioactivity; Radon dosimetry

1. Introduction

The naturally occurring radon gas and its decay products are responsible for approximately half of the total radiation dose received by the public [1]. Solid State Nuclear Tracks Detectors (SSNTDs) are polymer dielectric materials, widely used in radon research. Charged particles from protons upwards but not β-particles nor photons lead to intensive ionization, when they pass through these detectors. Along the path of the particle, a damaged region is created, usually named latent track, with diameter some tens of nm. A chemical etching procedure with a suitable etchant (NaOH or KOH) etches the detector's surface, preferentially with a faster rate on the damaged regions. As a result, latent tracks become permanent and can be sufficiently enlarged in order to be visible under an optical microscope. An ordinary track recording apparatus consists of a microscope equipped with a video camera—frame grabber—computer recording set-up. The shapes of the tracks are generally elliptical. For a standard chemical etching procedure the size, the brightness and the orientation of the track's ellipse are strongly dependent on the particles' charge, energy and incident angle [2]. Typical tracks created by alpha particles on the surface of the CR-39 SSNTDs are shown in Fig. 1.

The computer code described in this paper has been used by the authors for radon measurements and for the differentiation of α-particles' energies [3]. The code uses as input files a number of images in .jpg format recorded from SSNTD's surface. As output the user is provided with files in which the number of the recognized tracks is presented as well as tracks' parameters, such as the mean intensity values (brightness) and the orientation.

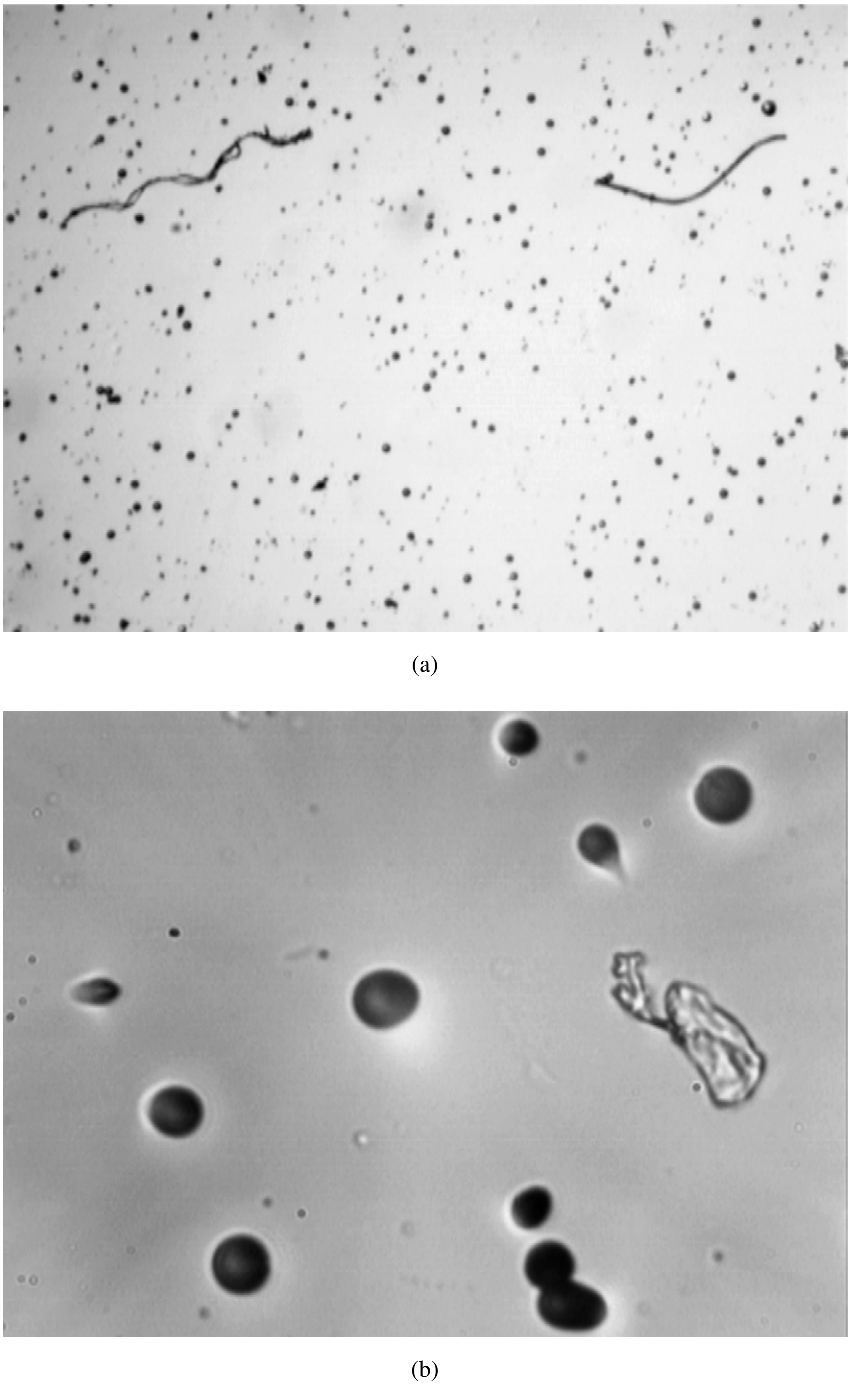
2. Computer code

2.1. General information

The program TRIAC II is written in the high level, language MatLab, which is accompanied by a variety of tools for special applications. It runs in two different modes followed by the corresponding calibration actions. The first mode is dedicated to the estimation of the number of the tracks from a SSNTD's surface image. The accurate measurement of the number of tracks is important for reliable radiation dose estimations. Usually, a number of images is required, containing as many tracks as possible for improved statistics. To include a large number of tracks in an image, lenses of low magnification may be used. As a result, most tracks are resolved as circles. For this reason the program counts the tracks as circles, when is functioning in the first mode. On the other hand, in the second mode the user is provided with the parameters of the recognized tracks, which in general have elliptical shapes. The major and minor axes (in pixels), the mean value of brightness and the orientation (in degrees) of the tracks are output in files. The results are more accurate if a lens of higher magnification is used. Also, in the calibration modes, a group of images is provided on screen. This group contains the initial and a number of analyzed images produced after certain program steps. The aim of the calibration modes is to help the user set the input program parameters, which fit better to his/her criteria. Finally, TRIAC II is supplied with a GUI, which facilitates the entry of input parameters and the choice of actions. A general flow diagram of the program is presented in Fig. 2.

2.2. Image segmentation

An image of a SSNTD's surface is represented in MatLab as a two-dimensional matrix of size equal to the image's digital resolution. TRIAC II is applied to greyscale images, in which grey scale intensities are represented using 8-bits per pixel (jpeg images). The first step of the analysis is the image segmentation task that groups the image pixels together and separates the objects (the useful information) from the background. A variety of methods have been proposed for image segmentation, such as the edge-based or the region-based methods [4]. Amongst them, histogram-based clustering methods have been proved very efficient, since


(a)

(b)

Fig. 1. Typical images from a CR39 SSNTD's surface exposed to a rich Radon environment. Figure (a) contains an optical field of 2.13 mm^2 (such images are usual to radiation dose measurements). Figure (b) contains a much smaller optical field of 0.03 mm^2, which is more suitable for tracks' parameters estimation.

they basically correspond to clustering approaches. A well-known clustering method is the *K-means* algorithm [5], which tries to appropriately adjust the K cluster centres in order to minimize the distance from each data point to its nearest centre. In the case of TRIAC II program the data points are the pixel intensity values. The algorithm estimates K centres (pixel intensity values) and groups all pixels accordingly. The criterion for a value to be a cluster's centre is the minimization of the sum of the distances from the nearest cluster centre. However, the term "distance" used here is not a real-space distance but actually it is the absolute difference of intensities.

Due to its local search, a known drawback of the algorithm is the initialization of their parameters (centres). This is accomplished by uniformly selecting a small subset of the pixel intensities (e.g., 10%) and executing the algorithm several times. The optimum solution found is then used for initializing the cluster centres. The number of these cluster centres K is entered by the user. As it has been observed, the input images, apart from the background (light pixels) and the track regions (dark pixels), contain

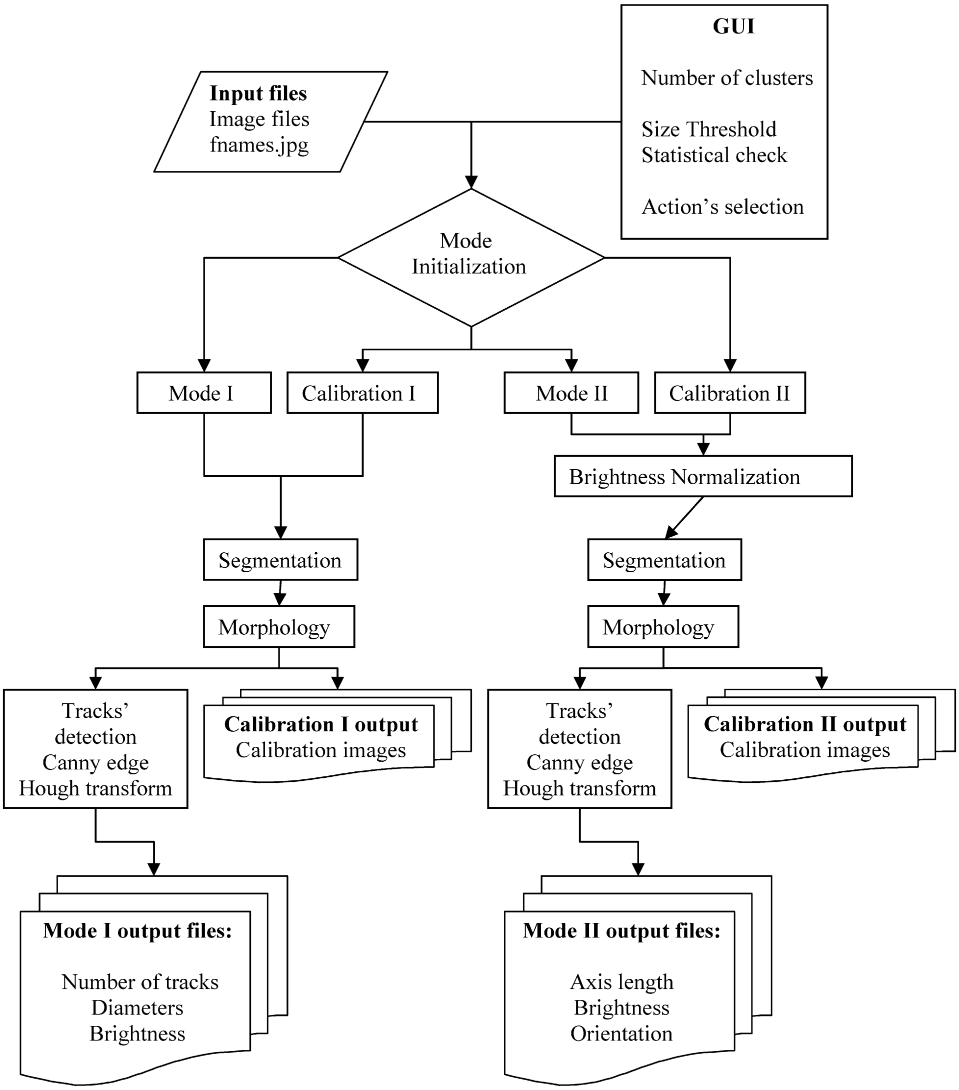


Fig. 2. A flow diagram of TRIAC II program.

a middle level(s) of brightness pixel regions (grey). This happens due to system imperfections during the generation process. Therefore, a value of $K = 3$ or 4 is used for the number of clusters during the clustering procedure to optimally distinguish the useful information (track objects) from the remaining parts of the image. At the end of the segmentation process, every pixel is labeled with a discrete value in the range of $[1, K]$ based on the cluster it belongs (minimum distance from the cluster centres). Since we are interested in the track objects, a binary image is finally produced by setting only the darker pixels to the value one (1) and leaving the rest as zero (0).

It must be noted that before the image segmentation and in the second mode of TRIAC II, a brightness normalization procedure of the initial image is performed. This step aims to reduce the influence of apparatus characteristics or imperfections in the measurement of pixel intensities. In some cases the background in photos containing particle tracks is not uniform. In addition because of apparatus imperfections the detector's photo may contain objects, which are not real tracks. An example is shown in Fig. 5(b). The center of the image is brighter and if a track was located there, its mean brightness value would be overestimated. Also, there are some "track-like" objects, which are not objects from the detector's surface but results of apparatus imperfections and obviously are not real tracks. These phenomena can increase the number of observed tracks and may affect the accuracy of brightness measurements.

The Brightness normalization step obliterates such problems. A user in addition to the detector's photos has to provide a background photo. For this photo, a user has not to include the detector itself but he must be careful to keep exactly the same apparatus adjustments (i.e. magnification, focusing) with which the tracks' photos were recorded. The grey scale pixel values of this photo, which must be named "BrightNormal.jpg", include any brightness inhomogeneities or imperfections of the apparatus and are subtracted from corresponding values of the detector photos.

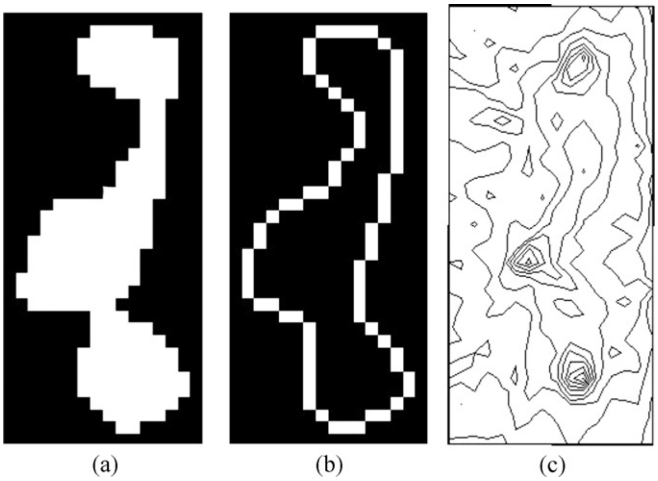


Fig. 3. (a) Object composed of three overlapped tracks. (b) Estimation of the object's edges. (c) Hough transform detects three tracks, because the accumulator function has 3 major peaks.

2.3. Image morphology

TRIAC II performs two types of morphological operations to the binary image which was produced from the segmentation task. The first uses an input parameter named *morphological threshold* provided by the user and aims to remove small objects, whose number of interconnected pixels is less than the user set threshold value. Depending on the size and characteristics of the image, this value should be adjusted, the higher the threshold value is set then larger objects will be removed. Furthermore, a statistical operation follows. After removing small objects, the program performs an initial estimation of all objects sizes and their average size together with the associated standard deviation. Then, objects with sizes greater than the mean value plus n-times the standard deviation are removed. An input value named *statistical check value* is related to the procedure described above that aims to remove imperfections on the detectors' surfaces or other oversized objects, which are not tracks. Setting this value low will allow more objects to be removed since their size exceeds the average size. The morphological threshold value and the statistical check value are determined using the calibration modes.

2.4. Tracks detection

The next step in the analysis is the determination of the number, as well as the geometric features of the tracks in the isolated objects of the resulting binary image. These objects may contain one or more overlapping tracks. In order to separate them, we followed the next strategy: In each object, we initially perform the Canny edge detection algorithm [6] to find object boundaries in the image. The known Hough transform [7,8] approach is then applied to identify the geometric structure of the tracks. In the first mode, as it was already mentioned, the tracks are considered circular so we apply the circular Hough transform. In this way, we extract the length of the diameter (in pixels) and the mean intensity value of each detected circular track. In the second mode a more complicated procedure is followed dealing with the general case of the elliptic shape of tracks. Here, we apply the Hough transform approach for ellipses [8] and the output includes four (4) features: the major axis, the minor axis, the orientation of the ellipse, and the mean intensity value of the surrounding pixels. In either case, we build an appropriate Hough space of dimensions equal to the parameters of the assumed geometric shape, where the Hough transform tries to fit any observed object with a number of objects of the appropriate shape (circle or ellipse) and its strongest peaks correspond to the number of the overlapping tracks in the object. The method iteratively detects the strongest peak and then sets the values of the accumulator function in the surrounding pixels to zero in order to avoid re-detecting the same feature during the next steps. This process continues until the number of remaining pixels in the current object is smaller than an internal and predefined threshold value. An example is presented in Fig. 3 in the case of the circular Hough transform.

2.5. User interface

In order to use the TRIAC II code, a MatLab v7.0 (or a newer version), the statistical toolbox and the image processing toolbox must be available. Before running the code a user has to create a working folder in which all the images are placed. Also this folder should include the special image file named BrightNormal.jpg, which represents the brightness distribution of the images' background, all the 12 TRIAC II's .m files (containing the TRIAC II code) and a text file where the names of images must be listed as imagename.jpg. If it is desired the images can be grouped by naming them using the same n-characters (letters or numbers) as the first characters in their filenames. This number of the same characters is another input value, which the user has to set on the GUI. After initializing MatLab, the user has access to the specified working folder using the MatLab's task bar.

When this step is completed in the command window, the user has to type TRIAC_II or double click the TRIAC_II.fig icon in the current directory window and the TRIAC II GUI will appear. In this state the user has to enter data to the program, completing the edit boxes with the name of .txt file, where the names of images are written, the length of the string which will name a group of images, the values for the number of clusters, the morphological threshold value, the morphological statistical check value and to specify if he/she wishes to produce histograms for any separate image or (and) for any image group. After completing the GUI's edit boxes, the user has only to click on the action button to initialize the desired mode of TRIAC II.

The most important parameters, which must be provided, are the number of clusters and the two morphological operator parameters. These will determine the final binary image which the Canny edge detection algorithm and the Hough functions will process. For an accurate measurement these parameters must be chosen carefully. It is strongly recommended to the user before any extensive counting to select a small number of representative images, and using the test modes of the program to determine the values that best fit certain criteria. A usual way is to keep one value constant (e.g., the number of clusters), then to adjust the other parameters running the test modes and from the resulting images to select the most appropriate combinations.

3. Conclusion

The program TRIAC II written in MatLab is useful for experimental measurements involving nuclear track detectors. It enables measurements of tracks' number and parameters like the axis (diameter) of an elliptical (circle) track, the mean value of their brightness and the orientation. The program is user friendly and demands a MatLab 7 or newer installation plus the Statistical and Image Processing toolboxes. When used in a systematic way, it provides reliable counting results of a big number of images even when overlapping tracks do exist.

Acknowledgements

This work was partly funded by the Greek Secretariat of Research and Technology (Contract GSRT-174γ).

Appendix A. Test runs

Typical images which can be analyzed by TRIAC II are presented in Fig. 1. They are images of a CR-39 SSNTD detector's surface exposed to a radon rich environment. The first one (Fig. 1(a)) represents an actual optical field of 2.18 mm^2, while the second (Fig. 1(b)) a field of 0.03 mm^2. Both have the same digital dimensions 1600 × 1200.

Table 1
TRIAC II input parameters for calibration actions

Calibration action I	Input parameters		Calibration action II	Input parameters	
	File name	TestI.txt		File name	TestII.txt
	String length	7		String length	7
	Clusters	3		Clusters	3
	Morphological	20		Morphological	1500
	Size check	3		Size check	2
	Histogram per image	n		Histogram per image	n
	Histogram per group	n		Histogram per group	n

Table 2
TRIAC II input parameters for the two running modes of TRIAC II

Running Mode I	Input parameters		Running Mode II	Input parameters	
	File name	ModeI.txt		File name	ModeII.txt
	String length	7		String length	7
	Clusters	3		Clusters	3
	Morphological	20		Morphological	1500
	Size check	3		Size check	2
	Histogram per image	n		Histogram per image	n
	Histogram per group	n		Histogram per group	n

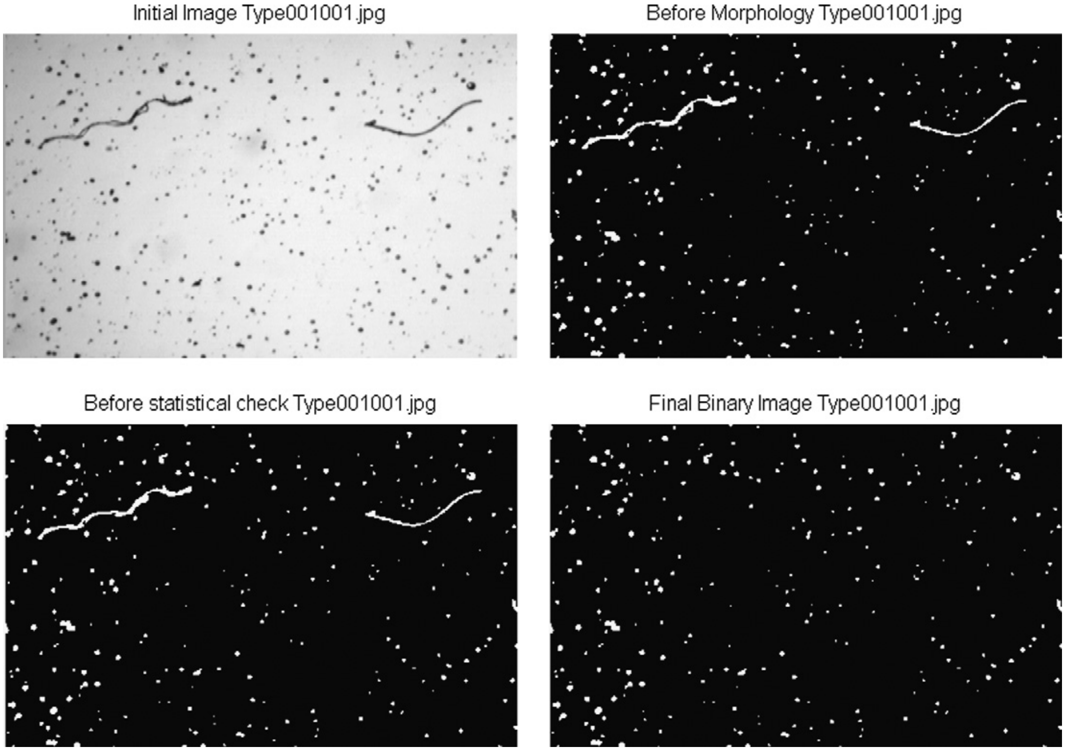

Initial Image Type001001.jpg
Before Morphology Type001001.jpg
Before statistical check Type001001.jpg
Final Binary Image Type001001.jpg

Fig. 4. Using the calibration run of the first TRIAC II mode, the transformation of an image to a binary image and the following morphological actions are displayed so to determine the input parameters.

A.1. Test run #1: Mode I and corresponding calibration action

The analysis of the image presented in Fig. 1(a) is performed by the first test run.

Input files: Type001001.jpg, Type001002.jpg, Type002001.jpg, Type002002.jpg (image files identical to the image shown in Fig. 1(a)), ModeI.txt (contains the names of the images), TestI.txt (contains the name of one image file for the Calibration I action).

GUI input parameters: The images have been analyzed using the input parameters, which are presented in Tables 1 and 2.

We chose to analyze the same image four times (with different names) in order to show the reliability of the program. The four image files compose two groups named Type001 and Type002 (the first seven characters). All the input files must me copied to the working folder together with the program's .m files.

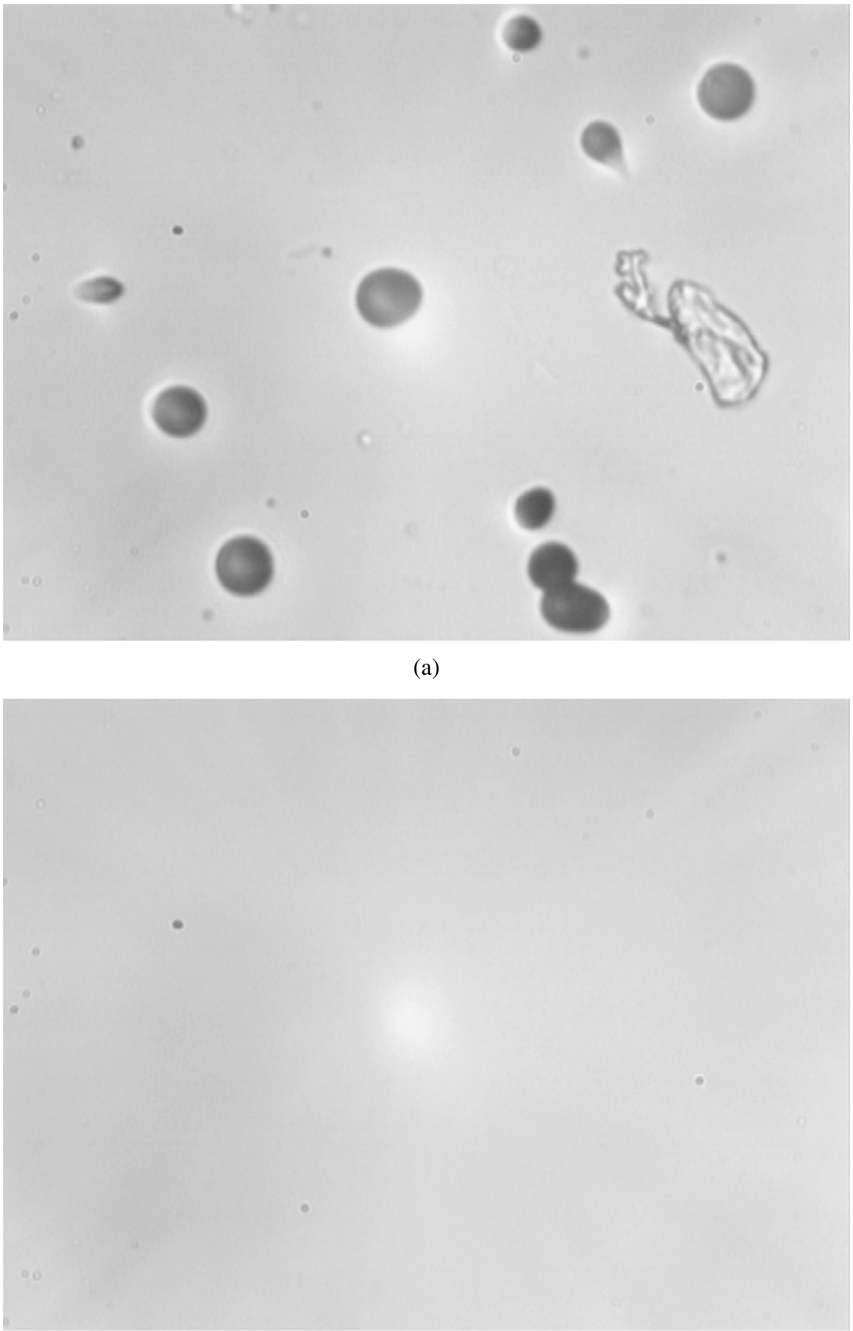
Output file #1: Output file of the Calibration I action. In Fig. 4 the results of the first calibration-mode are presented. The four images are the original, the first binary image produced by the clustering action and the two binary images resulting after the two morphology actions.

Output file #2: CircularTracks_PerImage.xls. Output file of the Mode I. A sample follows:

```
Num Objects  Num Tracks                       20                     161.6
        265         296                       14                     205.6
                                               6                     190.6
Diameter(pxl) Brightness                      18                     166.4
      18           192.3                      12                     180.1
       6           193.2                       4                     192.9
      12           169.3                       6                     186.2
       4           177.7                       6                     182.5
       8           171.6                      10                     181.7
                                              12                     168.8
```

At the first two rows, the number of the recognized objects and the estimated tracks are presented. The other rows contain the estimated diameter (in pixels) and mean brightness value for each one of the tracks. The names of the Excel's sheets are changed so that the name of the analyzed image is used instead.

Output file #3: CircularTracks_PerGroup.xls. Output file of the Mode I. This output file instead of data from each image, it contains data from each group of images. In the first two rows, the number of images which compose the corresponding group is



(a)

(b)

Fig. 5. An image of a track detector (a) and the image used for brightness normalization (b).

presented followed by the total number of tracks of the group. The other rows contain again the estimated parameters for each one of the tracks. The excel sheets now are labeled by the group name. Sample lines from this output .xls file are presented below:

```
Num of images Num of tracks                      14                      205.6
            2           592                       6                      190.6
                                                 18                      166.4
Diameter(pxl) Brightness                         12                      180.1
          18       192.3                          4                      192.9
           6       193.2                          6                      186.2
          12       169.3                          6                      182.5
           4       177.7                         10                      181.7
           8       171.6                         12                      168.8
          20       161.6
```

A.2. Test run #2: Mode II and corresponding Calibration action

The analysis of the image presented in Fig. 1(b) is the subject of the second test run.

Input files: Type003001.jpg, Type003002.jpg, Type004001.jpg, Type004002.jpg (image files identical to the image shown in Fig. 1(b)), BrightNormal.jpg (it must be included in this mode, Fig. 5(b)), ModeII.txt (contains the name of the images), TestII.txt (contains the name of one image for the Calibration II action).

GUI input parameters: The images have been analyzed using the input parameters which are shown in Tables 1 and 2.

Output file #1: Output file of the Calibration II action. In Fig. 6 the results of the Calibration II action are presented. The four images are the original image, the brightness normalized image, the first binary image produced at the end of the clustering phase and the final binary image which resulted after the two consequent morphology actions.

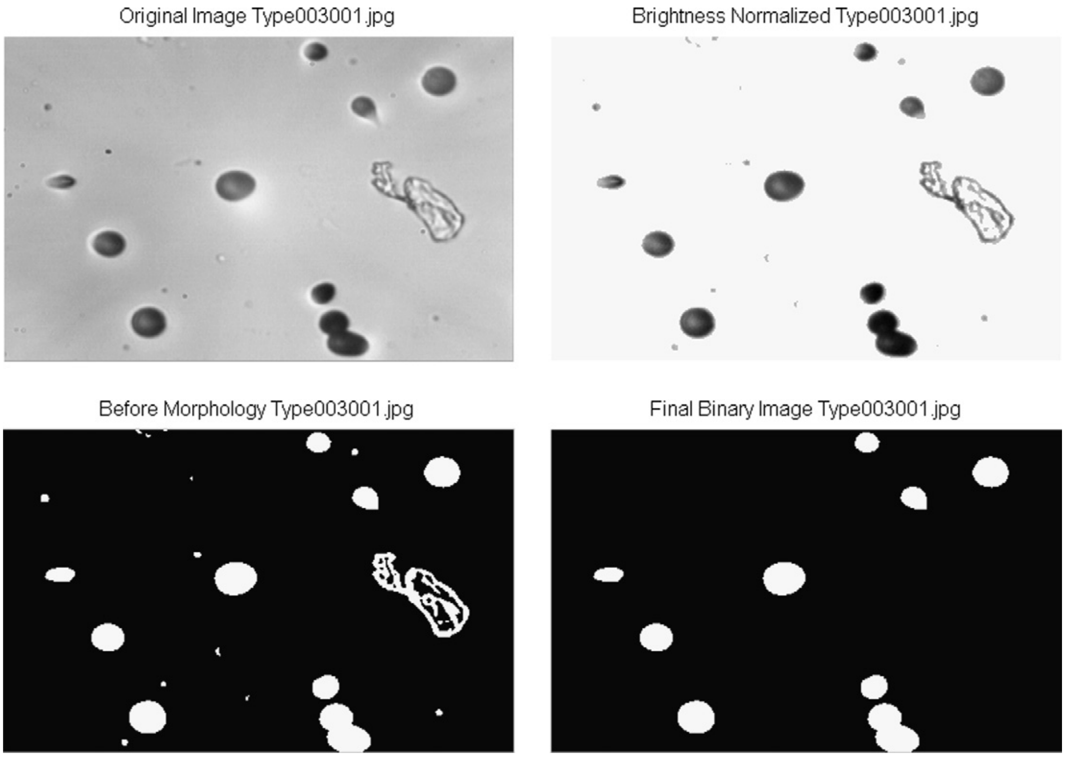

Original Image Type003001.jpg
Brightness Normalized Type003001.jpg
Before Morphology Type003001.jpg
Final Binary Image Type003001.jpg

Fig. 6. The results from the calibration run of the second TRIAC II mode.

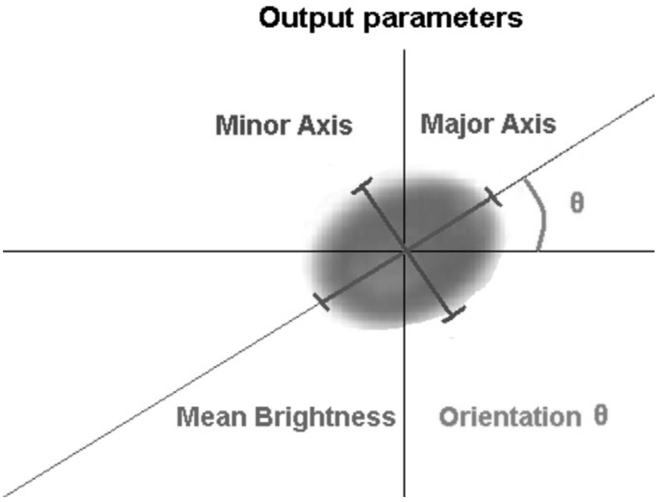


Fig. 7. A graphical presentation of the output parameters calculated by the second TRIAC II mode.

Output file #2: EllipticalTracks_PerImage.xls Output file of the Mode II. A graphical presentation of the output parameters is shown in Fig. 7. Ten lines of data are presented:

```
Major Axis(pxl)  Minor Axis(pxl)  Orientation(deg)  Brightness
      95.8             53.1              -2.6            217.6
     103.3            100.6              13.8            207.7
     129.7            112.2               8.4            201.5
     123.5            120.2             -21.8            203.0
      73.6             72.9              12.2            209.7
      85.0             82.2              34.0            193.4
     134.0            104.6             -19.0            184.9
     146.4            118.8              43.9            189.6
      81.8             75.9             -39.4            213.3
     110.9            110.2             -43.2            210.7
```

Output file #3: EllipticalTracks_PerGroup.xls Output file of the Mode II. This output file instead of data from each image, it contains data from each group of images.

```
Major Axis(pxl)  Minor Axis(pxl)  Orientation(deg)  Brightness
      95.8             53.1              -2.6           217.6
     103.3            100.6              13.8           207.7
     129.7            112.2               8.4           201.5
     123.5            120.2             -21.8           203.0
      73.6             72.9              12.2           209.7
      85.0             82.2              34.0           193.4
     134.0            104.6             -19.0           184.9
     146.4            118.8              43.9           189.6
      81.8             75.9             -39.4           213.3
     110.9            110.2             -43.2           210.7
      95.8             53.1              -2.6           217.6
     103.3            100.6              13.8           207.7
     129.7            112.2               8.4           201.5
     123.5            120.2             -21.8           203.0
      73.6             72.9              12.2           209.7
      85.0             82.2              34.0           193.4
     134.0            104.6             -19.0           184.9
     146.4            118.8              43.9           189.6
      81.8             75.9             -39.4           213.3
     110.9            110.2             -43.2           210.7
```

References

[1] United Nation Scientific Committee on the Effects of Atomic Radiation, The Report to the General Assembly with scientific Annexes, United Nations, New York, 2000.

[2] D. Nikezic, K.N. Yu, Formation and growth of tracks in nuclear track materials, Materials Science and Engineering R46 (2004) 51–123.

[3] D.L. Patiris, K. Blekas, K.G. Ioannides, TRIAC: A code for track measurements using image analysis tools, Nuclear Instruments and Methods in Physics Research B 244 (2006) 392–396.

[4] N.K. Pal, S.K. Pal, Pattern Recognition 26 (1993) 1277.

[5] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, New York, 2001.

[6] J. Canny, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986) 679.

[7] J.R. Parker, Algorithms for Image Processing and Computer Vision, John Wiley & Sons, Inc., New York, 1997.

[8] R.K.K. Yip, P.K.S. Tam, D.N.K. Leung, Modification of Hough transform for circles and ellipses detection using a 2-dimensional array, Pattern Recognition 25 (1992) 1007–1022.