Available online at www.sciencedirect.com
SCIENCE DIRECT®
Nuclear Instruments and Methods in Physics Research B 244 (2006) 392–396
NIM B Beam Interactions with Materials & Atoms
www.elsevier.com/locate/nimb

TRIAC: A code for track measurements using image analysis tools

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

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

Received 13 September 2005
Available online 15 December 2005

Abstract

A computer program named TRIAC written in MATLAB has been developed for track recognition and track parameters measurements from images of the Solid State Nuclear Track Detectors CR39. The program using image analysis tools counts the number of tracks for dosimetry proposes and classifies the tracks according to their radii for the spectrometry of alpha-particles. Comparison of manual scanning counts with those output by the automatic system are presented for detectors exposed to a radon rich environment. The system was also tested to differentiate tracks recorded by alpha-particles of different energies.

PACS: 29.40.Wk; 29.30.Ep; 29.85.+c

Keywords: Solid state nuclear tracks detectors; CR39; Automatic track counting; Image analysis; Alpha-particle radioactivity

1. Introduction

Following the passage of a charged particle through a solid state nuclear track detector (SSNTD) a damage region is created usually named latent track. Latent tracks can be etched using a suitable etchant (i.e. NaOH or KOH), sufficiently enlarging them to become visible under an optical microscope (with diameters of 1 μm or more). Using the appropriate apparatus one can take images of the SSNTD's surface and count the number of the tracks. The manual counting of many images is a tedious and time-consuming task, so an automatic system is needed to speed up the process. A number of automatic track counting systems have been reported [1–5], which recognize tracks on the basis of the grey level of images.

The code presented here, TRIAC for track image analysis, is based on a segmentation method that groups image pixels in a number of grey level groups chosen by the user. After the segmentation of pixels, TRIAC counts the tracks that were recorded, even those tracks which overlap and finally classifies them according to their diameters. The analysis of an image with a PC (Intel Pentium III processor running at 600 MHz) takes 2–5 min, depending on the number of observed tracks and the digital analysis of the image. As an application, the system was used to measure activities of radon and its daughters for dosimetry purposes and also to classify tracks according to their radii.

2. Description of the algorithms used for the automatic detection of tracks

Code TRIAC was written in the high level language MATLAB, which is accompanied by many tools for special applications. The algorithm performs image segmentation, the process that groups image pixels together, based on attributes such as their intensity, location, texture features, etc. A variety of methods have been proposed for image segmentation, such as edge-based or region-based methods [6]. Amongst them, histogram-based clustering methods have been proved very effective, since they basically correspond to clustering approaches. A well known clustering method is the *K-means* algorithm [7], which tries to

* Corresponding author. Tel.: +30 2651098545; fax: +30 2651098692.
E-mail address: kioannid@cc.uoi.gr (K.G. Ioannides).

0168-583X/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.nimb.2005.10.026

appropriately adjust the K cluster centers in order to minimize the distance from each data point to its nearest center. In the available track data, the resulting images contain, apart from the background (light pixels) and the track regions (dark pixels), pixel regions with a middle level of brightness (grey pixels). To capture the grey level pixels into the image model used for subsequent analysis, we have used $K = 3$ or 4 as the number of clusters for the clustering procedure.

Following image segmentation, pixels are labelled according to the cluster they belong to (1 for dark, 2 and more for grey light levels). Since we are mainly interested in the pixels related to the track regions, we produce a binary image from the segmented image by setting zero (0) to all pixels associated with labels 2 or higher. Furthermore, a morphological operator is performed to remove small objects whose number of interconnected pixels is less than a threshold value P (during our experiments, we have used $P = 20$). Fig. 1 provides an example of the above image analysis steps.

In this stage, the problem is reduced to counting the number of tracks in the isolated objects in the resulting binary image. The discovered objects may represent one or more overlapping tracks. For each object, the edges are found using the Canny edge detection algorithm [8], performed in two passes, one carried out with values only from pixels on a horizontal line adjacent to the central pixel, and the other executed with values only from pixels on a vertical line.

Seeing that the tracks are almost circular, the Hough transform is then applied [9], a technique to automatically detect the number of circles contained in each object. In particular, a three-dimensional Hough space is constructed. The generalized circular Hough transform requires a three-dimensional space, since three parameters are required to define a circle (the x, y coordinates of the center and the radius). Each point in the real-space image produces a peak into the Hough space, corresponding to all of the circles of various radii and center positions that could be drawn through the point. Once a cone is detected and a circle is 'recognized' at a particular point, nearby points are excluded as possible circle centers to avoid repeated detection of the same circular feature. Finally, the number of circles found corresponds to the number of tracks inside the object. An example of this process is presented in Fig. 2. It must be noticed that in the above

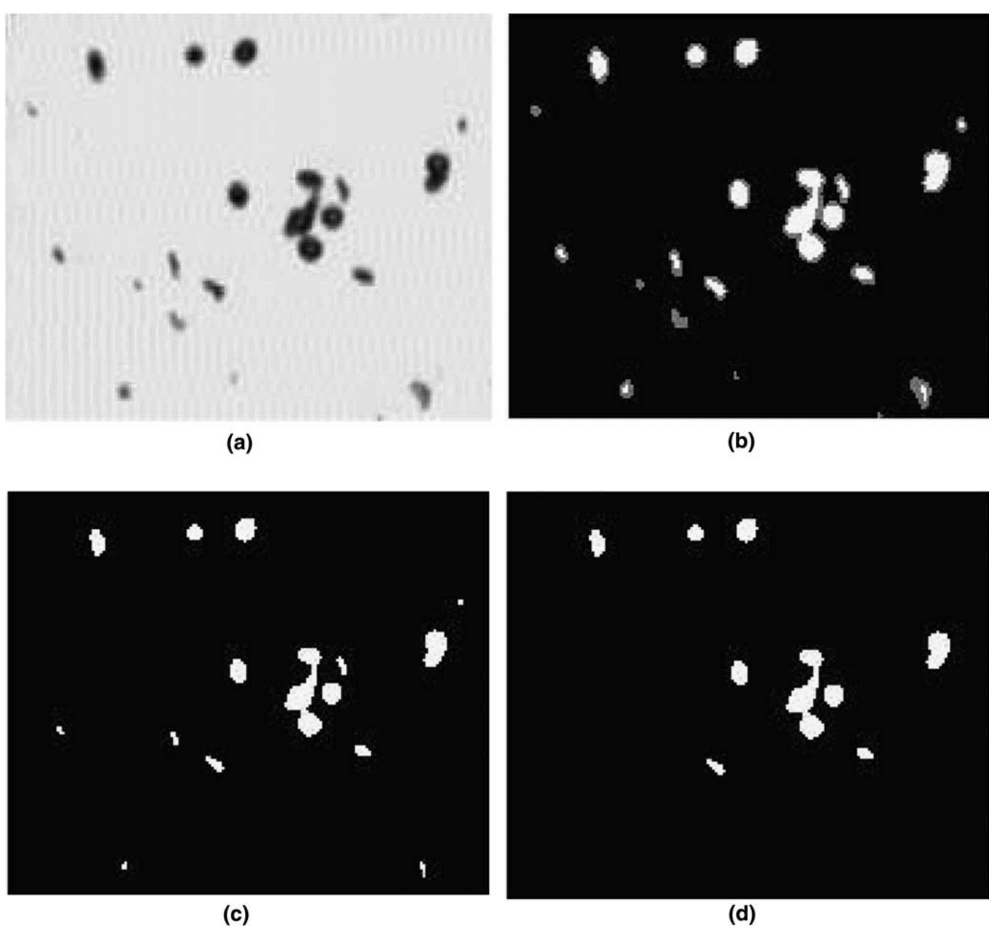

Fig. 1. The three main steps of track image analysis. The resulting corrected binary image (d) is finally used for counting the tracks: (a) original track image, (b) segmentation result: clustering the pixels into three segments (background, middle, track), (c) morphological correction: small track regions are removed and (d) binary image: only the track regions are visible.