UDC 004.932.

# USING COMPUTER VISION TO FIND OUTLINES OF OBJECTS

*S. RAHOUSKI, V. TANANA, S. VABISHCHEVICH*
**Polotsk State University, Belarus**

*The issues of using digital processing of surface images during microhardness tests to determine the geometric dimensions of prints and deformation zones are considered. An image processing algorithm has been built and a processing program has been implemented. The simulation results can be used when testing polymer films for microhardness to determine the strength characteristics: microhardness, crack resistance, specific energy of adhesion.*

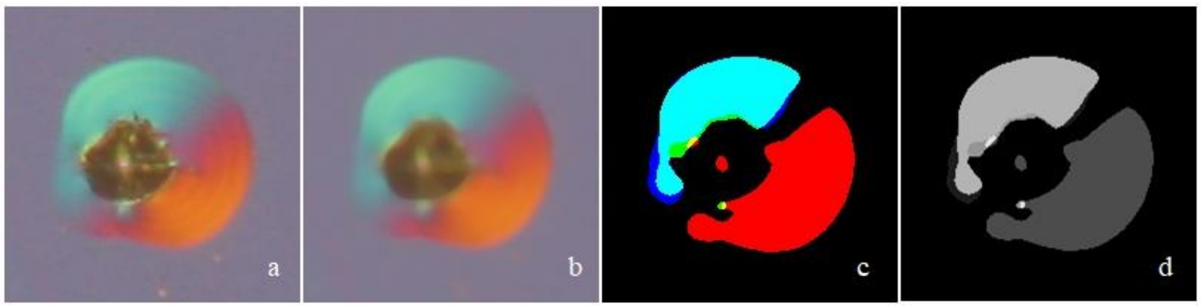
**Introduction.** Information technologies are widespread in all spheres of human activity. They enable us to quickly and most accurately carry out calculations, process information with the greatest accuracy and exclude subjectivity in the analysis of results.

The purpose of this work was to develop a program that allows one to obtain information on the strength characteristics of materials based on the analysis of indentations formed during microindentation of samples.

**Physical aspects of the project.** The modern production process of semiconductor devices involves a number of technological treatments, as a result of which, among others, the mechanical properties of materials change, which, in turn, can lead to the appearance of microcracks, scratches, chips and other surface defects. To determine the strength characteristics of silicon, various techniques are used, however, the closest to the real technological process is microindentation, since this method allows the most accurate modeling of the contact interaction of abrasive particles with the processed material [1]. There is a need to obtain a software product for recognizing digital images, which makes it possible to carry out a complete calculation of the strength characteristics of the material based on the analysis of photographs. The programming language Python was chosen as the development environment for the software product. The scheme for processing and analyzing digital image data was reduced to the following stages:

– Analysis of input image data;
– Processing of input data;
– Using the median filter;
– Image segmentation;
– Search for image contours;
– Display of contours in the image;
– Calculation of the geometric parameters of the photographing object.

**Image input data.** The input data are photographic images of prints during polymer indentation (fig. 1, a). In the center there is an imprint of an indenter pyramid in a polymer film surrounded by polymer heaps. The heap zone is spherical. The image was obtained by the method of differential interference contrast [2], which makes it possible to reveal the irregularities of the polymer surface. An image is made up of a collection of pixels. If you represent the image as a grid, then each square in the grid contains one pixel, where the square with coordinates [0,0] is the top-left pixel. The image size is 200x200 pixels for a total of 40,000 pixels. All pixels in the image are represented in RGB color space (red, green, blue), where one value for the red component, one for green and one for blue. Each of the three components is expressed as an integer in the range from 0 to 255, inclusive, which indicates how much color is contained.



**a – input image; b – image after using the median filter; c – image after segmentation; d – binary image**

**Fig. 1. – Stages of processing the input image**

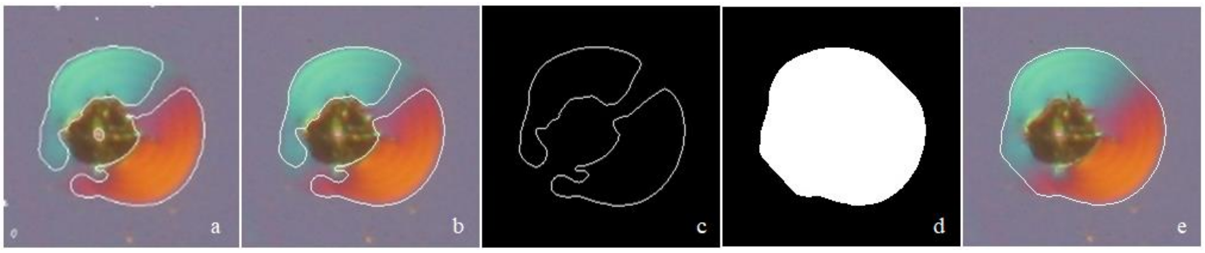
**Processing input data.** To process the image data, the OpenCV library and the Python programming language were selected [3]. OpenCV (Open Source Computer Vision Library) is one of the most popular libraries for computer vision applications. OpenCV-Python is the Python version of the interface for OpenCV. The presence of the C / C ++ code of this version of the library in the backend guarantees the speed of the library, and the Python wrapper in the frontend ensures ease of customization and deployment. Thanks to this, OpenCV-Python is an excellent solution for high-load computing programs for computer vision. The library also contains a huge number of algorithms, there are interfaces for many programming languages - Java, Ruby, Matlab, Lua, C / C ++.

**Input filtering.** The input image contains noise that interferes with image analysis. There are many different types of noise such as Gaussian noise, salt and pepper noise, and so on. We can remove this noise from the image by applying a filter that removes it, or at least minimizes its impact. There are many different filters, such as box filter, median filter, mode filter, Gaussian filter and many others. To remove noise in our processing algorithm, we use a median filter. Median filter is a type of digital filter widely used in digital signal and image processing to reduce noise. Median anti-aliasing is widely used in edge detection algorithms because, under certain conditions, it preserves edges while removing noise.

**Image segmentation.** The background is characterized by a constant color value, and the trace is characterized by a sharp change in color value. To select the contour, you need to separate the trace (object) from the background. Segmentation is used to separate the trace from the background. As discussed above, an image consists of RGB components and each component has a value from 0 to 255. If you set the segmentation value to 140 in the program, then all component values above 140 will be 255, and components with values below 140 will be 0. After segmentation by In the image, the trace has a more distinct outline (fig. 1, d). Those. the background turned black and took on the values of 0, and the area inside the trace became brighter and took on the corresponding value.

**Search for contours.** The outline of an object is its visible edge, which separates the trail from the background. In fact, most image analysis methods work with contours, not pixels per se. The OpenCV library implements convenient methods for detecting and manipulating image contours. To find contours, use the cv2.FindContours () function. This function takes three values: image, grouping mode and packing method. According to its format, this edge finder function accepts a binary image. To convert an image to binary, use the cv2.cvtColor (image, cv2.COLOR_BGR2GRAY) function. After applying the edge search function, the image (fig. 1, d) is ready for processing. The resulting contours of the trace must be displayed on the original image using the cv2.drawContours () function. The function takes an image, outline, outline color (specified in the RGB palette), outline thickness in pixels.

The result of mapping the contour in the image is shown in figure 2, a. There are several small unnecessary outlines in the image that need to be removed. When displaying contours on the image, you need to sort them by area. Using the cv2.contourArea () function, you can get the area value. The function takes all paths and outputs the area value in pixels. After that, you need to remove all unnecessary contours by sorting them by area. The area function is given by the expression: S = cv2.contourArea (contours [i]).



**a – the image with the applied contour; b – the image with the applied contour after sorting the contours by size; c – trace of the resulting contour on a black background; d – trail of the object on a black background after joining the contours; e – the original image with the started contour**

**Fig. 2. – Stages of search and display of contours in the original image**

Due to uneven lighting, we got several contours (fig. 2, c), which need to be combined into one. The contour consists of points, and when connecting the points of the contour, an object was obtained that coincided with the trace (fig. 2, c). The resulting trace coincides with the original processed image (fig. 1, a), the contour to which is the purpose of processing. To obtain a complete contour, it remains to apply the contour search function and display the resulting contour on the original image. The result is shown in figure 2, e.

The implementation of this algorithm for searching for a contour can be used to determine the geometric parameters of indentations during indentation of polymers, which makes it possible to calculate such strength characteristics as microhardness, fracture toughness К1с (stress intensity factor) and specific peel-off energy of the film, which is a characteristic of the adhesion of the polymer film to the base [4-6].

**Conclusion.** Thus, an algorithm for processing color images obtained by photographing the polymer surface after microhardness tests has been implemented, which aims to determine the geometric dimensions of prints, contours of heap areas, lengths of cracks and other objects. This is essential for the objective determination of the strength characteristics of materials and the automation of the measurement process in materials science.

REFERENCES

1. Litvinov, Yu.M. Methodology for determining the mechanical properties of semiconductor materials using the continuous indentation method / Yu.M. Litvinov, M.Yu. Litvinov // Izvestiya vuzov. Electronic engineering materials. - 2004. - No. 4. - P.11-16.
2. Anisovich, A.G. Optical effects in microscopy of nonmetallic materials / A.G. Anisovich // Casting and metallurgy. - 2017. - No. 1. - P.110-114.
3. Bradski, G. Learning OpenCV. Computer vision with the OpenCV library / G.Bradski, A.Kaehler //O'Reilly Media, Inc., – 2008. – 580 p.
4. Vabishchevich, S.A. Physical and mechanical properties of irradiated films of diazoquinone-novolac photoresist on silicon / SA Vabishchevich [et al.] // Herald of Polotsk State University. Series C. Fundamental Sciences. Physics – 2020. – No. 12. – P.60-64.
5. Strength properties of photoresist-silicon structures, γ-irradiated and implanted with B + and P + ions / S.A. Vabishchevich [and others] // Herald of Polotsk State University. Series C. Fundamental Sciences. Physics - 2016. - No. 12. - P.30-36.
6. Vabishchevich S.A., Brinkevich S.D., Brinkevich D.I., Prosolovich V.S. Adhesion of diazoquinon-novolac photoresist films with implanted boron and phosphorus ions to single-crystal silicon// High energy chemistry. – 2020. – V.54, № 1. – P.46-50.