UDC 004.93

D. Kvashuk
(National Aviation University, Ukraine)

Application of image recognition techniques in aviation to detect the icing of outer parts of an aircraft

Considered algorithms for pattern recognition, which are advisable to use in aviation. An example of a comparison of graphic images that can be used to detect the icing of outer parts of an aircraft is proposed.

Since icing is characterized by bulges on the investigated surface, aiming at a certain angle the directional light will cause a shadow and in turn will create a distinct contrast in the colors. It is precisely this contrast that can be identified by a number of machine algorithms for efficient contour analysis of graphical changes.

The corresponding procedure can be implemented using the Kenny algorithm to detect the change of the boundaries of the image. Therefore, the problem is reduced by the ratio of the image, which was before icing, and changes in this image in the presence of ice.

An important aspect in setting the task is the optimal direction of the DVR. It should cover the entire study area in such a way that it does not allow third-party graphic objects to fall into this area.

Since icing is characterized by bulges on the investigated surface, aiming at a certain angle the directional light will cause a shadow and in turn will create a distinct contrast in the colors. It is precisely this contrast that can be identified by a number of machine algorithms for efficient contour analysis of graphical changes.

The corresponding procedure can be implemented using the Kenny algorithm to detect the change of the boundaries of the image. Therefore, the problem is reduced by the ratio of the image, which was before icing, and changes in this image in the presence of ice.

An important aspect in setting the task is the optimal direction of the DVR. It should cover the entire study area in such a way that it does not allow third-party graphic objects to fall into this area.

The solution to this problem can be accomplished by comparing arrays of data that describe a graphic object using almost any programming language. For example, using the Python programming language, it is possible to work with the OpenCv library. The imread function of this library makes it possible to convert the raster image into an array of data, where each pixel has its numerical characteristic. By passing the created arrays into the calcHist function, you can obtain the total value of all the digital color values that are represented in the image. Then compare their histograms using the comparHist function.

Can apply the matchTemplate function to match the image areas that match. The results returning both functions can be correlated in the form of a sum, which characterizes the quantitative difference in the process of comparing two images, which can be applied to determine the changes of a homogeneous surface of the body

of the aircraft if it begins to appear ice. So, the base image (1.jpg) can be compared to image (2.jpg). The code that can be used for such a comparison is given below:

```
import cv2
class Image(object):
    def __init__(self, im_1_path, im_2_path):
        self.min_com_im_diff = 1
        self.im_1_path = im_1_path
        self.im_2_path = im_2_path
    def compare_im(self):
        im_1 = cv2.imread(self.im_1_path, 0)
        im_2 = cv2.imread(self.im_2_path, 0)
        com_im_diff = self.get_im_difference(im_1, im_2)
        if com_im_diff < self.min_com_im_diff:
            print ("Disagreement")
            return str(com_im_diff*100) + ' %'
        return 'Disagreement 100 %'
    @staticmethod
    def get_im_difference(im_1, im_2):
        first_im_hist = cv2.calcHist([im_1], [0], None, [256], [0, 256])
        second_im_hist = cv2.calcHist([im_2], [0], None, [256], [0, 256])
        img_hist_diff = cv2.compareHist(first_im_hist, second_im_hist, cv2.HISTCMP_BHATTACHARYYA)
        img_templ_probability_match = cv2.matchTemplate(first_im_hist, second_im_hist, cv2.TM_CCOEFF_NORMED)[0][0]
        img_templ_diff = 1 - img_templ_probability_match
        commutative_im_diff = img_hist_diff + img_templ_diff
        return commutative_im_diff
if __name__ == '__main__':
    compare_im = Image('1.jpg', '2.jpg')
    im_difference = compare_im.compare_im()
    print (im_difference)
```

Thus, the problem of diagnostics of icing of an aircraft fuselage can be solved by machine vision. Experiments in this example were not carried out, therefore the issue is for further development.

References

1. Gary Bradski, Adrian Kaehler. Learning OpenCV. Computer Vision with the OpenCV Library. // O'Reilly Media, Inc. – 2008.
2. OpenCV documentation. - Режим доступу: http://docs.opencv.org/index.html.
3. Shervin Emami. Face Recognition using Eigenfaces or Fisherfaces // Mastering OpenCV with Practical Computer Vision Projects. Pact Publishing, 2012.