System of technical vision for autonomous unmanned aerial vehicles

To cite this article: A S Bondarchuk 2018 IOP Conf. Ser.: Mater. Sci. Eng. 363 012027

View the article online for updates and enhancements.
System of technical vision for autonomous unmanned aerial vehicles

A S Bondarchuk
Student, National Research Tomsk State University, Tomsk, Russia
E-mail: bondarchuk.a.c@gmail.com

Abstract. This paper is devoted to the implementation of image recognition algorithm using the LabVIEW software. The created virtual instrument is designed to detect the objects on the frames from the camera mounted on the UAV. The trained classifier is invariant to changes in rotation, as well as to small changes in the camera's viewing angle. Finding objects in the image using particle analysis, allows you to classify regions of different sizes. This method allows the system of technical vision to more accurately determine the location of the objects of interest and their movement relative to the camera.

1. Introduction
One of the important ways to use unmanned aerial vehicles (UAVs) is to collect information. UAVs are used in various areas of human activity: security monitoring, geodesy, exploration, etc. The greatest amount of information is contained in the video signal [1, 2]. To work with this information, UAVs are equipped with vision systems, and image storage and transmission system for later use for other purposes. With the help of a vision system, unmanned aerial vehicles will be able to recognize and respond to certain objects. This system will allow the UAV to follow a certain object and thereby monitor its location. This paper was devoted to creating a virtual instrument in LabVIEW for detecting objects on the frames from the camera mounted on the UAV.

2. Classifier training
The image of an object can be directly represented as a vector in a multidimensional space. Then train some classifier using a set of sample images of objects. The classifier in this case is some tool that takes an input image, represented as a vector in a multidimensional space, and outputs some information that classifies the input image relative to some attribute [3, 4].

To train the classifier, the virtual instrument "HOG-LBP Texture Classification" was used, which is located in the LabView Examples. The IMAQ Extract HOG Feature Vector and IMAQ Extract LBP Feature Vector blocks are combined to form a feature vector. The best results were obtained using the following parameters: Grid Size: 4×4 (along the x and y axes respectively); Histogram Bins: 25.

The classifier was trained in two classes, the class "Car", in which samples of vehicle images were added, and the class "negative". In the "negative" class, samples that did not contain objects of interest were added. When the first sample was added, the SVM model of one class was used, the C-SVC model was used for the subsequent training.

3. Pattern recognition on frames from drones
This virtual instrument is designed to detect vehicles on an image or video stream from a camera mounted on a UAV. To begin with, Vision Acquisition VI was installed, After closing the
configuration window, a While loop is created in the VI, which contains the Vision Acquisition and Image Out blocks. Next, the construction of the entire block diagram will occur within this cycle.

The frames received with the Vision Acquisition block will be delivered to the IMAQ ExtractSingleColorPlane and to the Vision Assistant. The frames arriving at the input of this module are converted first into images in gradations of gray (with a sensitivity to the red color). Further on the received images the filter using edge detection-prewitt is superimposed.

After using edge detection-prewitt, a threshold binarization of the image is performed [5, 6]. When setting the binarization parameters, the "Auto Threshold: Clustering" method was selected and the Lower Limit value was set to 10 and the Upper Limit was set to 245. In the Look For section, the Bright Objects option was chosen to assign the value 1 to the pixels at the object boundary. The last image conversion in Vision Assistant is the use of the Equalize function, which assigns the values 0 and 255 to the pixels 0 and 1. All these transformations are necessary for further work with the images, namely finding the coordinates of regions containing a continuous region of non-zero pixels. These areas are usually called particles.

A subVI was created to detect pixel groupings in the image and sequentially obtain the coordinates of these groupings. Particle analysis consists of a series of operations for processing and analyzing functions that provide information about particles in an image. Using particle analysis, you can detect and analyze any shapes in the image. A typical particle analysis process scans the entire image, detects all particles, and builds a detailed report on each particle. You can use several parameters such as perimeter, angle, area, and center of mass to identify and classify these particles. This analysis is necessary for the search for particles whose spatial characteristics satisfy certain criteria. To reduce the calculation time, particle filtration is used, excluding particles that are not of interest based on their spatial characteristics. Only the particles of the necessary size remain for further analysis. Thus, the output of this subVI will be the location of the objects in the image. A block diagram of the virtual instrument Classification analysis is shown in Figure 1.

![Figure 1. Block diagram of the virtual instrument Classification analysis.](image)

Images are delivered to the IMAQ Particle Analysis Report block, which returns the number of particles found in the binary image and an array of reports containing the most commonly used particle measurements.
Subsequently, in the sub-VI Classification analysis, only the Particle Reports cluster (Pixels) will be used. To process data for all image particles, the For Loop is used.

The Particle Reports cluster (Pixels) uses the Unbundle By Name function, which returns cluster elements such as Bounding Rect and Center of Mass. To the coordinates of the sides of the rectangle, 5 pixels are added in order to have additional space around the particle, and it does not touch the boundary of the ROI. These coordinates are then fed to Case Structure. The Case Selector input receives the rectangle width values for each particle. If the width of the rectangle surrounding the particles is less than 50 pixels, the values of the coordinates of its sides become equal (this value varies depending on the iteration cycle). If the width of the rectangle is more than 50 pixels, the coordinate values are not changed. This is done in order to reduce the time spent processing the image, since objects of such small size are not of interest. Equal values of the coordinates of the sides of the rectangle can cause an error in the work of the VI. This is because the IMAQ Extract HOG Feature Vector and IMAQ Extract LBP Feature Vector blocks can not form a feature vector from this region. To prevent such an error, use Case Structure, whose input receives an error. In the absence of errors, the structure sums the unit and the current value of the iteration cycle, which is fed to the input of sub-VI Classification analysis. The resulting value goes to the previous Case Structure, where it replaces the coordinate values of rectangles that are less than 50 pixels wide. When an error occurs, this value becomes equal to one.

From coordinates passed through Case Structure, a data array is built using the Build Array block. This array is fed to the Bundle function, which creates a cluster of three elements (the array is fed to the input, for the third element of the cluster). To the input, for the first element, is fed by Enum Constant. By default, the numeric representation of this constant is a 16-bit unsigned integer. To configure the constant, after clicking on it with the right mouse button, select the Properties section. In the appeared window, in the Appearance tab, in the Label section, the word ID is written. In the next Data Type tab, the numeric representation of this constant is changed to a 32-bit unsigned integer. In the Edit Items tab, External marks are added with a value of 0 and Internal with a value of 1. After clicking OK, the Internal label is selected in the constant. On the input, for the second element of the cluster created with the Bundle function, another Enum Constant is supplied. When you configure it in the Appearance tab, in the Label section, the word Type is written. The Data Type is set to 32-bit, and the Edit Items are added with the labels specified in Table 1. After clicking OK, the Rectangle label is selected in the constant.

| Tag          | Value |
|--------------|-------|
| No Selection | 0     |
| Point        | 1     |
| Line         | 2     |
| Rectangle    | 3     |
| Oval         | 4     |
| Polygon      | 5     |
| Free         | 6     |
| Unused1      | 7     |
| Unused2      | 8     |
| Unused3      | 9     |
| Broken Line  | 10    |
| Free Hand Line | 11     |

The resulting cluster of three elements is converted into a data array using the Build Array block. This array is fed to the Bundle function, as the second element, of the new cluster being created. To the input for the first element is fed an array of coordinates, which was used in the creation of the
previous cluster. The new cluster created in this way represents the regions of interest (ROI). In these regions, there are objects that will be classified to search for vehicles in the image.

So, sub VI Classification analysis was created. Its input receives the image and the value of the iteration cycle (i). At the output, the regions of interest and the centers of mass of the image particles are returned.

For further classification, the For Loop cycle was created. Within this cycle, an ROI cluster and a pre-converted image in grayscale (block IMAQ ExtractSingleColorPlane, in the parameter Color Plane is selected Red) are delivered to the inputs of the IMAQ Extract HOG Feature Vector and IMAQ Extract LBP Feature Vector blocks. In the parameters extraction features from the image, the Histogram Bins value is set to 25 and the Grid Size is $4 \times 4$. Next, the generated feature vector is fed to the IMAQ Classify Custom block, which classifies this vector using the classifier (which is also fed into the input).

Also, another subVI was created, which eliminates unnecessary information about objects that are not of interest for the developed system of technical vision. A block diagram of the virtual Instrument Condition is shown in Figure 2.

![Figure 2. Block diagram of the virtual instrument Condition.](image)

The classifier of the technical vision system was trained in two classes, the class "Car", which includes attributes belonging to vehicles, and the class "negative", representing samples of other objects. In order to distinguish objects of only one class (namely "Car"), Case Structure was built. Its input receives a variable of type Boolean, which takes the value "False" if the sample is classified as "negative". Thus, if the input is set to "True", the data for the classification object passes through Case Structure. If the input is set to "False", the data is assigned zero values by means of creating constants. Through this structure, we direct the Class variable obtained at the output of the IMAQ Classify Custom block and the ROI that were obtained at the output of the sub-VI Classification analysis.

Further, these variables must pass through yet another Case Structure, which sets a valid value for the score of the matching of input samples to the assigned class. This value ranges from 0 to 1000, where 1000 represents 100% certainty of belonging to the class. The input of this structure receives the Identification Score variable, obtained by using the Unbundle By Name function on the Scores cluster (obtained from IMAQ Classify Custom). If the value of this variable is greater than 800, the data about the classified objects pass through the Case Structure without changes. For smaller values, the data is assigned zero values.

As a result, Class, Scores and ROI variables are supplied to the input of the Sub VI Condition, and the output returns the values of the variables (Class and ROI) that satisfy the specified requirements. The IMAQ Overlay ROI and IMAQ Overlay Text blocks are used to display the areas of interest and class names. The input of these blocks receives the variables ROI, Class and frames from the drone’s
camera. These frames can be both original images from the camera, and pre-filtered, it depends on what image needs to be obtained at the output of the VI.

It should be noted that for the IMAQ Overlay Text block, coordinates are also needed, according to which the text will be displayed. This can be the coordinates of the center of mass obtained in the sub-VI Classification analysis. Figure 3 is a block diagram of an implemented virtual instrument intended for recognizing vehicles on frames received from cameras installed on a UAV.

![Block diagram of VI intended for pattern recognition on frames from UAV.](image)

**Figure 3.** Block diagram of VI intended for pattern recognition on frames from UAV.

4. Results
Finding objects in the image using particle analysis, allows you to classify regions of different sizes. This method allows the system of technical vision to more accurately determine the location of the objects of interest and their movement relative to the camera. The classifier, trained in vehicle recognition, is invariant to changes in rotation, as well as to small changes in the camera's viewing angle.

The disadvantage of the method using particle analysis is that when changing lighting or background environments, there is a chance that object in the image will not be found. This is due to the change in the contrast of the image, so that the outlines of objects may not be highlighted during filtering. For example, a car, painted in black, drove into a section of the road that is in the shade of a large tree. In such a situation, the difference in the pixel values of the object and its background is small, and consequently, obtaining of the boundaries of the object is problematic. When the contrast of the image is increased with the help of filters, the number of particles in the image increases, which
hinders the determination of the size of objects and their location. Also, increasing the contrast adds extra noise to the image.

Consequently, this virtual instrument most effectively distinguishes objects of interest on areas whose color palette is monotonous, and differs from the color palette of objects.

The work was carried out in Tomsk State University with the financial support of the Ministry of Education and Science of Russia, the unique identifier of the project RFMEFI57817X0241.

References
[1] Shashev D V, Shidlovskiy S V, Syriamkin V I, Yurchenko A V 2015 IOP Conference Series: Materials Science and Engineering 81 012101
[2] Panin S V, Syryamkin V I, Glukhikh A I 2003 Avтометрия (4) 79–92
[3] Yane B 2007 Digital image processing (Moscow, Technosphere)
[4] Okhotnikov V V et al 2016 IOP Conference Series: Materials Science and Engineering 116 012001
[5] Shidlovskiy S 2016 MATEC Web of Conferences 79 01014
[6] Shashev D, Shidlovsky S 2017 Journal of Physics: Conference Series 881(1) 012029