UDC 528.715:004.932 (045)
DOI: 10.18372/1990-5548.55.12785

[1]M. P. Mukhina,
[2]I. V. Barkulova

ALGORITHM OF VARIATIVE FEATURE DETECTION AND PREDICTION IN CONTEXT-DEPENDENT RECOGNITION

[1,2]Educational & Research Institute of Information and Diagnostic Systems,
National Aviation University, Kyiv, Ukraine
E-mails: [1]marinamukhina79@gmail.com, [2]irochkaseden@gmail.com

Abstract**—***Application of context-dependent classification for recognition tasks is proposed. In the context-free classification, the starting point was the Bayesian classifier. Morphological features such as object form, area, and eccentricity were considered through context-dependent classification. As result, dependences which can be used for object recognition have been obtained, and further they can be used together with interesting point detectors. The procedure of prediction of object variative features was developed.*

Index Terms—Object recognition; context-dependent classification; (binary large object) blob analysis.

I. INTRODUCTION

Nowadays Unmanned Aerial Vehicle (UAV) can be used to solve many tasks that are not performed by aircraft because of various reasons. Unmanned Aerial Vehicles play an important role in mobile aerial monitoring operations and have been widely applied in diverse applications.

As most monitoring systems require detection and recognition of the object, then recognition problems for airborne video observation can be solved with the context-dependent classification. The idea of this classification is the following. The object is observed not once, but continuously, during some period of time, frame by frame, N frames (observations, feature vectors). If there is a standard pitch or roll evolution, in context-dependent classification it is necessary to have a priori data about object changing in time, frame by frame.

II. REVIEW OF EXISTING METHODS

Feature detection, description and matching are essential components of various computer vision applications, thus they have received a considerable attention in the last decades. Image representation, image classification and retrieval, object recognition and matching, 3D scene reconstruction, motion tracking, texture classification, robot localization, and biometrics systems, all rely on the presence of stable and representative features in the image. Thus, detecting and extracting the image features are vital steps for these applications.

Recently, several methods have been developed for extracting invariant local image descriptors. Scale Invariant Feature Transform (SIFT) [1] is a method to extract features invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. Those properties make it suitable for being used in robotics applications, where changes in robot viewpoint distort the images taken from a conventional camera. But it is known that SIFT suffers from a high computational payload. Speeded-Up Robust Features (SURF) [2] are other detector-descriptor algorithms developed with the aim of speeding up the key point localization step without losing discriminative capabilities. The scale-space is analyzed by up-scaling the filter size instead of by iteratively reducing the image size as occurs in the SIFT approach.

Several methods can be used for extracting the region of interest (ROI). A priori knowledge of objects to be identified can be used, for instance, shape or color information, but this would make the method specific for a concrete environment or class of objects. Instead, the approach can be generalized by scanning the image for continuous connected regions or blobs. A blob (binary large object) is an area of touching pixels with the same logical state. Blob extraction, also known as region detection or labelling, is an image segmentation technique that categorizes the pixels in an image as belonging to one of many discrete regions. The process consists of scanning and numbering any new regions that are encountered, but also merging old regions when they prove to be connected on a lower row. Therefore, the image is scanned and every pixel is individually labelled with an identifier which signifies the region to which it belongs [3].

Main advantages of this technique include high flexibility and excellent performance. Its limitations are clear background-foreground relation requirement and pixel-precision.

The basic scenario of the Blob Analysis solution consists of extraction, refinement, analysis.

There are two techniques that allow extracting regions from an image:

Image Thresholding – commonly used methods that compute a region as a set of pixels that meet certain condition dependent on the specific operator. The resulting data is always a single region.

Image Segmentation is more specialized set of methods that compute a set of blobs corresponding to areas in the image that meet certain condition. The resulting data is always an array of connected regions (binary large objects – blobs).

Recognition process by contour or object form is non stable. Under unstable conditions, such as airborne observations, the feature will be unstable also. As usual, researchers don't apply variative features for ground object recognitions.

In this article morphological features such as object form, area, and eccentricity were considered through context-dependent classification. As result, dependences which can be used for object recognition will be obtained, and in the following they can be apply for SURF method.

III. Problem Statement

The classification tasks considered so far assumed that no relation exists among the various classes. In other words, having obtained a feature vector x from a class ω_i, the next feature vector could belong to any other class. In this chapter we will remove this assumption and we will assume that the various classes are closely related. That is, successive feature vectors are not independent. Under such an assumption, classifying each feature vector separately from the others obviously has no meaning. The class to which a feature vector is assigned depends (a) on its own value, (b) on the values of the other feature vectors, and (c) on the existing relation among the various classes. Such problems appear in various applications such as communications, speech recognition, and image processing [4].

In the context-free classification, our starting point was the Bayesian classifier. In other words, a feature vector was classified to a class ω_i, if $P(\omega_i \mid x) \succ P(\omega_j \mid x). \forall j \neq i$.

The Bayesian point of view will also be adopted here.

However, the dependence among the various classes sets demands for a more general formulation of the problem. The mutual information that resides within the feature vectors requires the classification to be performed using all feature vectors simultaneously and also be arranged in the same sequence in which they occurred from the experiments.

For this reason, in this chapter we will refer to the feature vectors as observations occurring in sequence, one after the other, with x_1 being the first and x_N the last from a set of N observations.

A. The Bayes classifier

Let $X: x_1, x_2, ..., x_N$ -be a sequence of N (feature vectors) observations and $\omega_i, i = 1, 2, \dots, M$, the classes in which these vectors must be classified. Let Ω_i: $\omega_{i_1}, \omega_{i_2}, ..., \omega_{i_N}$ be one of the possible sequences of these classes corresponding to the observation sequence, with $i_k \in \{1, 2, ..., M\}$ for $k = 1, 2, ..., N$.

The total number of these class sequences Ω_i is M^N, that is, the number of possible ordered combinations of M distinct objects taken in groups of N.

Our classification task is to decide to which class sequence Ω_i a sequence of observations X corresponds.

This is equivalent to appending x_1 to class ω_1, x_2 to ω_2, and so on. A way to approach the problem is to view each specific sequence X as an (extended) feature vector and $i = 1, 2, ..., M^N$, as the available classes. Having observed a specific X, the Bayes rule assigns it to Ω_i for which

$$P(\Omega_i \mid X) \succ P(\Omega_j \mid X), \quad \forall j \neq i, \tag{1}$$

and following our already familiar arguments, this is equivalent to

$$P(\Omega_i) p(X \mid \Omega_i) \succ P(\Omega_j) p(X \mid \Omega_j), \quad \forall i = j \tag{2}$$

B. Markov chain models

One of the most widely used models describing the underlying class dependence is the Markov chain rule. If $\omega_{i_1}, \omega_{i_2}, \dots$ is a sequence of classes, then the Markov model assumes that

$$P(\omega_{i_k} \mid \omega_{i_{k-1}}, \omega_{i_{k-2}}, ..., \omega_{i_1}) = P(\omega_{i_k} \mid \omega_{i_{k-1}}). \tag{3}$$

The meaning of this is that the class dependence is limited only within two successive classes. This type of model is also called a first-order Markov model, to distinguish it from obvious generalizations (second, third, etc.). In other words, given that the observations $x_{k-1}, x_{k-2}, ..., x_1$ belong to classes $\omega_{i_{k-1}}, \omega_{i_{k-2}}, ..., \omega_{i_1}$, respectively, the probability of the observation x_k, at stage k, belonging to class ω_{i_k}

depends only on the class from which observation x_{k-1}, at stage $k-1$, has occurred. Now combining (3) with the probability chain rule

$$P(\Omega_i) \equiv P(\omega_{i_1}, \omega_{i_2}, ..., \omega_{i_N})$$
$$= P(\omega_{i_N} | \omega_{i_{N-1}}, ..., \omega_{i_1}) P(\omega_{i_{N-1}} | \omega_{i_{N-2}}, ..., \omega_{i_1}) ... P(\omega_{i_1}),$$

we obtain

$$P(\Omega_i) = P(\omega_{i_1}) \prod_{k=2}^{N} P(\omega_{i_k} | \omega_{i_{k-1}}), \quad (4)$$

where $P(\omega_{i_1})$ is the prior probability for class ω_{i_1}, $i_1 \in \{1, 2, ..., M\}$, to occur.

Furthermore, two commonly adopted assumptions are: 1) given the sequence of classes, the observations are statistically independent; 2) the probability density function in one class does not depend on the other classes. That is, dependence exists only on the sequence in which classes occur; but within a class observations "obey" the class' own rules. This assumption implies that

$$P(X | \Omega_i) = \prod_{k=1}^{N} p(x_k | \omega_{i_k}). \quad (5)$$

Combining equations (3) and (4), the Bayes rule for Markov models becomes equivalent to the statement:

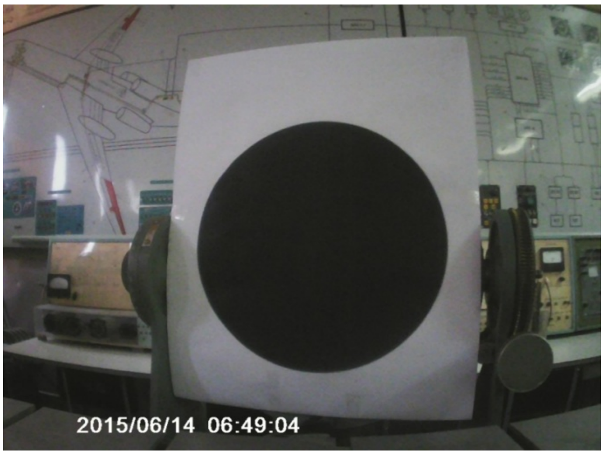
Statement. Having observed the sequence of feature vectors $X : x_1, ..., x_N$, classify them in the respective sequence of classes $\Omega_i : \omega_{i_1}, \omega_{i_2}, ..., \omega_{i_N}$, so that the quantity becomes maximum.

$$p(X | \Omega_i) P(\Omega_i) = P(\omega_{i_1})$$
$$\cdot p(x_1 | \omega_{i_1}) \prod_{k=2}^{N} P(\omega_{i_k} | \omega_{i_{k-1}}) p(x_k | \omega_{i_k}). \quad (6)$$

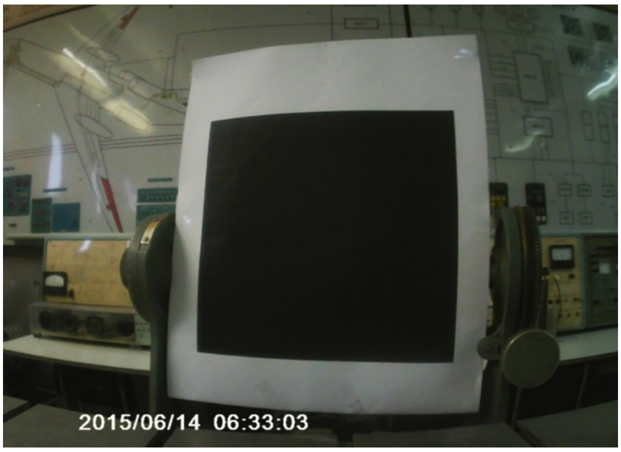
$P(X | \Omega_i)$ can be obtained from series of experiments of object observation. If state vector is predicted, a priory probability of our state vector also can be predicted and set.

IV. THE PROCEDURE OF PREDICTION OF OBJECT VARIATIVE FEATURES

Realization of research was done in Matlab 2014a program software on the set of serial images of known forms (Fig. 1). First of all the arrays for data with the help of built-in Matlab function **zeros**(), which returns the array of zeroes, were created.



(a)



(b)

Fig. 1. Two standard shapes for BLOB analysis: (a) – circle; (b) – square

Then the object of **BlobAnalysis** was created, and the properties were set, need to find in the blob-regions as true.

During the square processing it is needed to find area, perimeter and ratio of perimeters. To find area and perimeter it was processed the first image and got the above mentioned parameters which must satisfy the sated conditions and written them into the arrays.

For this purpose, a variable was created and set it equal to -30 degrees, then convert this variable to string with the help of function **num2str**(), and connect two strings with the help of built-in function **strcat**(). As the result of these actions the name of image to load was obtained.

Load the image by the function **imread**().

Then the **imcomplement**() function was used. In the complement of an intensity or RGB image, each pixel value is subtracted from the maximum pixel value supported by the class (or 1.0 for double-precision images) and the difference is used as the pixel value in the output image. In the output image, dark areas become lighter and light areas become darker.

Then function **rgb2gray**() was applied. It converts the true color image RGB to the grayscale intensity image by forming a weighted sum of the R, G, and B components:

$$0.2989 \cdot \mathrm{R} + 0.5870 \cdot \mathrm{G} + 0.1140 \cdot \mathrm{B}.$$

By the function **im2bw** it is converted the grayscale image to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). Specify level in the range [0,1]. This range is relative to the signal levels possible for the image's class.

Function **bwlabel**() finds a matrix, of the same size as binary image, containing labels for the connected objects in the binary image. This function will numerate the objects in the same way as blob analysis will do it if their connectivity coincides.

Then the step method is used. The step method computes and returns statistics of the input binary image depending on the property values specified.

After all the rest images have to be processed in the cycle automatically, by comparing the values of objects in the image with the area, perimeter found in the previous image and write it into the array.

Also the ratio of perimeter of bounding box to perimeter of the square itself has to be found. For this purpose another approach is applied, it would be used in circle processing. The size of labeled matrix of our image was found and dividing by 2 the coordinates of the center are obtained.

By taking the number which is written in the center cell the number of square among the other objects will be found.

As the property of **BlobAnalysis**, **BoundingBox** returns the coordinates of upper left corner of bounding box and its width and height. Knowing the number of object, it could easily find the bounding box perimeter and find the ratio, as the perimeter of the square is already known.

Also it was created an array, named Angle which contains the pitch and roll angles of the images. It is used in finding the linearized coefficients and building graphs. The same is done for circle processing. And then we have obtained such graphs.

In the standard formulation of Least Square Method, a set of N pairs of observations $\{Y_i, X_i\}$ is used to find a function relating the value of the dependent variable (Y) to the values of an independent variable (X). With one variable and a linear function, the prediction is given by the following equation:

$$\hat{Y} = a + bX.$$

This equation involves two free parameters which specify the intercept a and the slope b of the regression line.

Solving the normal equations gives the following least square estimates of a and b as:

$$a = M_Y - bM_X,$$

(with M_Y and M_X denoting the means of X and Y) and

$$b = \frac{\sum (Y_i - M_Y)(\mathrm{X}_i - M_X)}{\sum (X_i - M_X)^2}.$$

The standard deviation σ is the square root of the variance:

$$\sigma = \sqrt{p_1, p_2, \ldots, p_{n+1} \sum_{i=1}^{\min N} p^{(n)}(x_i) - y_i)^2}.$$

Morphological features such as object form, area, and eccentricity were considered through context-dependent classification. As result, dependences which can be used for object recognition have been obtained, and further they can be used together with interesting point detectors (Tables I and II). Using stressed variances the prediction can be done. Approximation of object parameters on rotation angle was done and results are presented in Table III.

TABLE I. CHANGES OF CIRCLE OBJECT PARAMETERS WITH ROTATION

Pitch (at turn table), deg	**Roll (at turn table), deg**	**Circle parameters**		
		Eccentricity ε	***Minor semi axis R_1***	***Major semi axis R_2***
-30	0	0.994924958992809	1253.13142238587	12454.1347470843
-25	0	0.994377739067458	1289.75205311403	12179.9924777981
-20	0	0.994098908925513	1311.12007473435	12086.5669033497
-15	0	0.994021186095166	1322.70304542757	12114.0664442731
-10	0	0.994764320113121	1296.20586698273	12683.5739193823
-5	0	0.995375869301636	1261.47753488595	13132.6422140036
0	0	0.993504799355868	1352.07976380840	11882.2143983671
5	0	0.614973880128102	1069.36388894846	1356.11858172303

10	0	0.994185986827588	1317.46529503643	12235.4049614340
15	0	0.994869516035837	1279.00700262332	12642.6032196238
20	0	0.995638862121070	1230.31906737052	13187.9371814185
25	0	0.498474931678574	1170.41436679682	1350.10821634461
30	0	0.571961532839223	1102.59609937165	1344.16967874569
0	-30	0.994975820824839	1362.60047277338	13610.2830606043
0	-25	0.994390882992566	1367.27410425397	12927.1622164135
0	-20	0.993920198354908	1370.16519572963	12444.4108240683
0	-15	0.993497454651913	1371.76884653455	12048.4560579856
0	-10	0.993202515561859	1373.32413415806	11798.4030421206
0	-5	0.992953648000559	1376.42672858090	11615.0833661476
0	0	0.992838216317717	1378.16635703947	11535.9940360336
0	5	0.992800244658300	1380.95537087561	11528.9270238133
0	10	0.992895489324830	1382.51149415130	11618.7498905739
0	15	0.993099978133292	1384.02942533479	11801.9983035186
0	20	0.993411888383932	1384.67169972952	12082.8064817491
0	25	0.993778372074907	1387.65423932115	12459.2186913966
0	30	0.994293956970706	1387.95247544451	13011.0542592614

TABLE II. CHANGES OF SQUARE OBJECT PARAMETERS WITH ROTATION

Pitch (at turn table), deg	**Roll (at turn table), deg**	**Square parameters**		
		Area A	***Perimeter P***	***Ratio of perimeters***
-30	0	1816865	5679.34942187369	1.00715761174506
-25	0	1855747	5657.12402512927	1.00828618475793
-20	0	1870790	5952.20937829103	0.950571399695031
-15	0	1904233	5625.47727214752	1.00471475157918
-10	0	1895386	5624.06305858516	0.994654565876663
-5	0	1881747	5614.16356364854	0.989283610467266
0	0	1866213	5596.83261120685	0.991989645872725
5	0	1829267	5559.98694013939	0.993527513548712
10	0	1767572	5488.28845532955	0.998489792328335
15	0	1709559	5425.78593001261	1.00298833573542
20	0	1629696	5324.69761825804	1.00587871537242
25	0	1545285	5275.24891681025	1.00090063677841
30	0	1459416	5102.17784899838	1.02074058453733
0	-30	1625827	5767.89061915813	0.952861342714260
0	-25	1698723	5941.76572747710	0.931036370959195
0	-20	1757343	5927.39819510896	0.939029202491041
0	-15	1802064	5741.02352006587	0.970558643511022
0	-10	1836049	5677.18289963229	0.980415832711769
0	-5	1859469	5578.79098064650	0.996631710936709
0	0	1868489	5598.74935008616	0.993436150149213
0	5	1865549	5569.63665171395	0.998987967785616
0	10	1849373	5591.19300090005	0.993705636544046
0	15	1822826	5583.09249583666	0.994072730147075

0	20	1780868	5534.74935008616	0.998780550001591
0	25	1725131	5482.93311627023	1.00274796051862
0	30	1657217	5454.08744520278	1.00145075686386

TABLE III. APPRROXIMATION OF OBJECT PARAMETERS DEPENDENCIES ON ROTATION ANGLES

	Coefficients of approximation		**Variance σ**
	b	***a***	
Eccentricity **ε** = ***f*(γ)**	-1.224929996613e-05	0.993542974326948	0.002235456054552
Minor semi axis R_1 = ***f*(γ)**	0.408487698949604	1377.50004184053	4.062667821121734
Major semi axis R_2 = ***f*(γ)**	-9.41356451785406	12190.9651733605	2.196759033895257e+03
Area **A** = ***f*(γ)**	559.883516483513	1780686.76923077	2.815714091106756e+05
Perimeter **P** = ***f*(γ)**	-7.03571942740912	5649.88410401422	2.478595649759944e+02
The ratio of perimeter of bounding box to perimeter of the square itself *R*= ***f*(γ)**	0.00108634939248195	0.981054988871848	0.045294310764797
Eccentricity **ε** = ***f*(θ)**	-0.005923240051655	0.894705576267844	0.528698286956307
Minor semi axis R_1 = ***f*(θ)**	-2.31185419491153	1250.43349857585	2.558304717163356e+02
Major semi axis R_2 = **f(θ)**	-140.099980029834	9896.11791873445	1.404678838066026e+04
Area **A** = ***f*(θ)**	-6102.75824175825	1771675.07692308	2.819803389546672e+05
Perimeter **P** = ***f*(θ)**	-9.67829876418024	5532.72346464848	3.514633929872431e+02
The ratio of perimeter of bounding box to perimeter of the square itself *R* = ***f*(θ)**	0.000299488426394598	0.997629488330344	0.053201302026730

V. CONCLUSIONS

The relation of geometric parameters at changing pitch and roll angles are similar in general. So, in general, the area changing both during pitch or roll angles are non-linear, more quadratic it can be said and not very sensitive. Taking into account possible errors, it should be stressed that perimeter change is almost linear, reverse and not sensitive at pitch and roll changing. But the ratio of bounding box perimeter to square perimeter has almost linear direct non-sensitive dependence.

As for circle investigated parameters. It is complicated to say the dependence of eccentricity at pitch angle change, because at first it has linear direct dependence and then it has no dependence at all, and as for roll angle change, the eccentricity has quadratic dependence. The Major axes at pitch angle change has similar view to eccentricity parameter at pitch change and at roll angle change-similar to eccentricity parameter at roll change. Minor axes at pitch angle change has non-linear dependence, may be it had to be quadratic. Minor axes at roll angle has almost linear direct non-sensitive dependence.

Morphological features such as object form, area, and eccentricity were considered through context-dependent classification and approximated by linear dependences with acceptable error variance from 0.002235456054552 to 0.528698286956307. Since this models of variative feature evolution can be used together with other features like SURF for increasing the reliability and accuracy of object recognition.

REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 60(2): pp. 91–100, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Proceedings of the 9th European Conference on Computer Vision*, 2006.

[3] Berthold K. P Horn, *Robot Vision*. MIT Press, 1986.

[4] Theodoridis S. Konstantinos Koutroumbas, *Pattern recognition*, Elsevier, 2003.

Received November 03, 2017.

Maryna Mukhina. Doctor of Engineering Science. Professor.

Aviation Computer-Integrated Complexes Department, Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine.

Education: National Aviation University, Kyiv, Ukraine, (2002).

Research area: Navigation and motion control.
Publications: more than 60 papers.
E-mail: marinamukhina79@gmail.com

Iryna Barkulova. Post-graduate student.
Aviation Computer-Integrated Complexes Department, Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine, (2014).
Research area: Navigation and motion control.
Publications: more than 12 papers.
E-mail: irochkaseden@gmail.com

М. П. Мухіна, І. В. Баркулова. Алгоритм виявлення варіативних ознак та прогнозування в контекстно-залежному розпізнаванні

Запропоновано використання контекстно-залежної класифікації для задач розпізнавання. Початковою точкою в довільній класифікації було обрано класифікатор Байєса. Морфологічні ознаки такі як форма об'єкта, площа, ексцентриситет, розглянуті з точки зору контекстно-залежної класифікації. Як результат, отримано залежності, які можна використовувати для розпізнавання об'єкта, і далі апроксимоване аналітичне представлення може використовуватися разом з детекторами точок інтересу. Розроблено процедуру прогнозування варіативних ознак об'єкта.

Ключові слова: розпізнавання об'єкта; контекстно-залежна класифікація; blob аналіз.

Мухіна Марина Петрівна. Доктор технічних наук. Професор.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційно-діагностичних систем, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна, (2002).
Напрям наукової діяльності: навігація та управління рухом.
Кількість публікацій: понад 60.
E-mail: marinamukhina79@gmail.com

Баркулова Ірина Володимирівна. Аспірант.
Кафедра авіаційних комп'ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційних діагностичних систем, Національний авіаційний університет, Київ, Україна.
Освіта: Національний Авіаційний Університет, Київ, Україна, (2014).
Напрям наукової діяльності: навігація та управління рухом.
Кількість публікацій: понад 12.
E-mail: irochkaseden@gmail.com

М. П. Мухина, И. В. Баркулова. Алгоритм выявления вариативных признаков и прогнозирования в контекстно-зависимо классификации распознавания

Предложено использование контекстно-зависимой классификации для задач распознавания. Начальной точкой в контекстно-зависимой классификации был выбран классификатор Байеса. Морфологические признаки, такие как форма объекта, площадь, эксцентриситет, рассмотрены с точки зрения контекстно-зависимой классификации. Как результат, получены зависимости, которые можно использовать для распознавания объекта, и дальше аппроксимированные аналитическое представление может использоваться вместе с детекторами точек интереса. Разработана процедура прогнозирования вариативных признаков объекта.

Ключевые слова: распознавание объекта; контекстно-зависимая классификация; blob анализ.

Мухина Марина Петровна. Доктор технических наук. Профессор.
Кафедра авиационных компьютерно-интегрированных комплексов, Учебно-научный институт информационно-диагностических систем, Национальный авиационный университет, Киев, Украина.
Образование: Национальный авиационный университет, Киев, Украина, (2002).
Направление навигация и управление движением.
Количество публикаций: более 60.
E-mail: marinamukhina79@gmail.com

Баркулова Ирина Владимировна. Аспирант.
Кафедра авиационных компьютерно-интегрированных комплексов, Учебно-научный институт информационно-диагностических систем, Национальный авиационный университет, Киев, Украина.
Образование: Национальный авиационный университет, Киев, Украина, (2014).
Направление навигация и управление движением.
Количество публикаций: более 12.
E-mail: irochkaseden@gmail.com