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

1,2Educational & Research Institute of Information and Diagnostic Systems, National Aviation University, Kyiv, Ukraine
E-mails: 1marinamuakhina79@gmail.com, 2irochkaseden@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 speed-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 \( \omega_i \), 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 \( \omega_i \), if

\[
P(\omega_i | x) > P(\omega_j | x), \forall j \neq i.
\]

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 \( x_j \) 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 \( \omega_i, i = 1, 2, ..., M \), the classes in which these vectors must be classified. Let \( \Omega_i : \omega_1, \omega_2, ..., \omega_{N_i} \) be one of the possible sequences of these classes corresponding to the observation sequence, with \( i \in \{1, 2, ..., M\} \) for \( k = 1, 2, ..., N \).

The total number of these class sequences \( \Omega_i \) 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 \( \Omega_i \) a sequence of observations \( X \) corresponds.

This is equivalent to appending \( x_i \) to class \( \omega_i \), \( x_j \) to \( \omega_j \), 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 \( \Omega_i \) for which

\[
P(\Omega_i | X) > P(\Omega_j | X), \quad \forall j \neq i,
\]

and following our already familiar arguments, this is equivalent to

\[
P(\Omega_i) p(X | \Omega_i) > P(\Omega_j) p(X | \Omega_j), \quad \forall i = j
\]

B. Markov chain models

One of the most widely used models describing the underlying class dependence is the Markov chain rule. If \( \omega_1, \omega_2, ..., \omega_N \) is a sequence of classes, then the Markov model assumes that

\[
P(\omega_1, \omega_2, ..., \omega_N) = P(\omega_1 | \omega_2, ..., \omega_N).
\]

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 \( x_{k-1}, x_{k-2}, ..., x_k \) belong to classes \( \omega_{k-1}, \omega_{k-2}, ..., \omega_k \), respectively, the probability of the observation \( x_k \), at stage \( k \), belonging to class \( \omega_k \),
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

\[
P(\Omega_j) = P(\omega_{i_k}, \omega_{i_{k-1}}, ..., \omega_{i_1})
\]

we obtain

\[
P(\Omega_j) = P(\omega_{i_k}) \prod_{k=2}^{N} P(\omega_{i_k} | \omega_{i_{k-1}}),
\]

where \( P(\omega_{i_k}) \) is the prior probability for class \( \omega_{i_k} \), \( i_k \in \{1, 2, ..., M\} \), 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

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

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 \( \Omega_i : \omega_i, \omega_i, ..., \omega_i \), so that the quantity becomes maximum.

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

\( P(X | \Omega_i) \) 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 \texttt{zeros}(), which returns the array of zeroes, were created.

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

Then the object of \texttt{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 \texttt{num2str}(), and connect two strings with the help of built-in function \texttt{strcat}(). As the result of these actions the name of image to load was obtained.

Load the image by the function \texttt{imread}().

Then the \texttt{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 \texttt{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:

\[
0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B.
\]

By the function \texttt{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 \texttt{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 been 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 was done and results are presented in Table I.

```plaintext
<table>
<thead>
<tr>
<th>Pitch (at turn table), deg</th>
<th>Roll (at turn table), deg</th>
<th>Eccentricity ε</th>
<th>Minor semi axis R₁</th>
<th>Major semi axis R₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>-30</td>
<td>0</td>
<td>0.994924958992809</td>
<td>1253.131422385847</td>
<td>12454.1347470843</td>
</tr>
<tr>
<td>-25</td>
<td>0</td>
<td>0.994377739067458</td>
<td>1289.752053114033</td>
<td>12179.9924777981</td>
</tr>
<tr>
<td>-20</td>
<td>0</td>
<td>0.994098908925513</td>
<td>1311.12007473435</td>
<td>12086.5669033497</td>
</tr>
<tr>
<td>-15</td>
<td>0</td>
<td>0.994021186095166</td>
<td>1322.70304542757</td>
<td>12114.0664442731</td>
</tr>
<tr>
<td>-10</td>
<td>0</td>
<td>0.994764320113121</td>
<td>1296.70586698273</td>
<td>12683.5739193823</td>
</tr>
<tr>
<td>-5</td>
<td>0</td>
<td>0.995375869301636</td>
<td>1261.47753488595</td>
<td>13132.6422140036</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.993504799355868</td>
<td>1322.70304542757</td>
<td>13132.6422140036</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.614973880128102</td>
<td>1069.36388894846</td>
<td>1356.11858172303</td>
</tr>
</tbody>
</table>
```

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>
<thead>
<tr>
<th>Pitch (at turn table), deg</th>
<th>Roll (at turn table), deg</th>
<th>Area A</th>
<th>Perimeter P</th>
<th>Ratio of perimeters</th>
</tr>
</thead>
<tbody>
<tr>
<td>-30</td>
<td>0</td>
<td>1816865</td>
<td>5679.34942187369</td>
<td>1.00715761714506</td>
</tr>
<tr>
<td>-25</td>
<td>0</td>
<td>1855747</td>
<td>5657.12402512927</td>
<td>1.00828618475793</td>
</tr>
<tr>
<td>-20</td>
<td>0</td>
<td>1870790</td>
<td>5952.20937829103</td>
<td>0.950571399695031</td>
</tr>
<tr>
<td>-15</td>
<td>0</td>
<td>1904233</td>
<td>5625.47727214752</td>
<td>1.0047175157918</td>
</tr>
<tr>
<td>-10</td>
<td>0</td>
<td>1895386</td>
<td>5624.06305858516</td>
<td>0.99465465876663</td>
</tr>
<tr>
<td>-5</td>
<td>0</td>
<td>1881747</td>
<td>5614.16356364854</td>
<td>0.989283610467266</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1866213</td>
<td>5596.83261120685</td>
<td>0.991998645872725</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1829267</td>
<td>5559.98694013939</td>
<td>0.993527513548712</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1767572</td>
<td>5488.28845532955</td>
<td>0.998489792328335</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1709559</td>
<td>5425.78593001261</td>
<td>1.0029833573542</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>1629696</td>
<td>5324.69761825804</td>
<td>1.00587871537242</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>1545285</td>
<td>5275.24891681025</td>
<td>1.00090063677841</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>1459416</td>
<td>5102.17784899838</td>
<td>1.0207408543733</td>
</tr>
<tr>
<td>0</td>
<td>-30</td>
<td>1625827</td>
<td>5767.89061915813</td>
<td>0.952861342714260</td>
</tr>
<tr>
<td>0</td>
<td>-25</td>
<td>1698723</td>
<td>5941.76572747710</td>
<td>0.931036370959195</td>
</tr>
<tr>
<td>0</td>
<td>-20</td>
<td>1757343</td>
<td>5927.39819510896</td>
<td>0.939029202491041</td>
</tr>
<tr>
<td>0</td>
<td>-15</td>
<td>1802064</td>
<td>5741.02352006587</td>
<td>0.970558643511022</td>
</tr>
<tr>
<td>0</td>
<td>-10</td>
<td>1836049</td>
<td>5677.18289963229</td>
<td>0.98041583211769</td>
</tr>
<tr>
<td>0</td>
<td>-5</td>
<td>1859469</td>
<td>5578.79098064650</td>
<td>0.99631710936709</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1868489</td>
<td>5598.74935008616</td>
<td>0.993436150149213</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>1865549</td>
<td>5569.63665171395</td>
<td>0.9989796785616</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>1849373</td>
<td>5591.9300090005</td>
<td>0.993705636544046</td>
</tr>
<tr>
<td>0</td>
<td>15</td>
<td>1822826</td>
<td>5583.09249583666</td>
<td>0.994072730147075</td>
</tr>
</tbody>
</table>

**Table II. Changes of square object parameters with rotation**
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 change is almost linear, reverse and not sensitive. Taking into account possible errors, it should be stressed that perimeter change is almost linear, reverse and not sensitive.

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.

**V. CONCLUSIONS**

<table>
<thead>
<tr>
<th>Eccentricity $e = f(\theta)$</th>
<th>Coefficients of approximation</th>
<th>Variance $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b$</td>
<td>$a$</td>
</tr>
<tr>
<td>Eccentricity $e = f(\theta)$</td>
<td>-1.224929996613e-05</td>
<td>0.993542974326948</td>
</tr>
<tr>
<td>Minor semi axis $R_1 = f(\theta)$</td>
<td>0.408487698949604</td>
<td>1377.50004184053</td>
</tr>
<tr>
<td>Major semi axis $R_2 = f(\theta)$</td>
<td>-9.41356451785406</td>
<td>12190.9651733605</td>
</tr>
<tr>
<td>Area $A = f(\theta)$</td>
<td>559.883516483513</td>
<td>1780686.76923077</td>
</tr>
<tr>
<td>Perimeter $P = f(\theta)$</td>
<td>-7.03571942740912</td>
<td>5649.88410401422</td>
</tr>
<tr>
<td>The ratio of perimeter of bounding box to perimeter of the square itself $R = f(\theta)$</td>
<td>0.00108634939248195</td>
<td>0.981054988871848</td>
</tr>
</tbody>
</table>

**REFERENCES**


М. П. Мухина, 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 Байеса. 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 апроксимовано 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.

Мухина Марина Петровна — Доктор технічних наук. Професор. Кафедра авіаційних комп’ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційно-діагностичних систем, Національний авіаційний університет, Київ, Україна.

Баркулова Ірина Володимирівна — Аспірант. Кафедра авіаційних комп’ютерно-інтегрованих комплексів, Навчально-науковий інститут інформаційних діагностичних систем, Національний авіаційний університет, Київ, Україна.

Контактна інформація:
- Мухина М. П.: marinamukhina79@gmail.com
- Баркулова І. В.: irochkaseden@gmail.com