ROZPRAWY INŻYNIERSKIE • ENGINEERING TRANSACTIONS • 38, 2, 329-343, 1990
Polska Akademia Nauk • Instytut Podstawowych Problemów Techniki

HIERARCHICAL PATTERN RECOGNITION METHOD WITH THE MULTI-LEVEL DISTANCE FUNCTION

L.J. CHMIELEWSKI and W. KOSIŃSKI (WARSZAWA)

To overcome the disadvantages of the conventional distance pattern recognition method, based on the distance function of a complicated form, it is proposed to introduce a set of simpler functions which form a multi-level distance function. The resulting recognition method consists in gradual elimination of the classes to which a recognized object does not belong. The proposed, very simple algorithm operates in such a way as if in the set of classes there existed a hierarchy which would make it possible to exclude subsequently the whole groups of classes; however, such hierarchy is not introduced. It is (implicitly) comprised in the structure of the recognition algorithm. This provides for a significant reduction of the execution time. The algorithm has been applied to object recognition in computer vision, where the requirement of large calculation speed is particularly important.

1. Introduction

In the paper we shall discuss some particular aspect of the domain of pattern recognition. This domain is considered mainly within the context of computer analysis of sensor data, like speech signal or picture.

The object of our interest will be the picture. As the data for analysis we shall understand its representation in the form of a rectangular array, the elements of which are values of the brightness function for the picture in a discrete set of points of the rectangle. These points, which form a rectangular grid, are called pixels. The result will be the information about the viewed scene. The contents and suitable symbolic representation of this information depends on the needs of the particular application [1].

For a manipulating robot (the majority of industrial robots manipulate objects in such or another sense) the necessary information is what objects are present in the field of view and what are their locations.

Processing of a picture, given by the brightness function, in order to obtain its symbolic description, can be conventionally divided into three stages. Their customary names are: low level, middle level and high level vision analysis.

The general goal of the low level analysis is to extract from the array of elements, which represent the brightness function, this geometrical information on the discrete counterparts of lines, blobs and surfaces, present in the brightness array, which can be obtained without using any knowledge about the viewed objects.

On the middle level, the description (interpretation) of the obtained information in terms of the elements of physical objects, the presence of which in the field of view can be expected, is carried out. This process is called the segmentation of the picture into parts, which corresponds to the fragments of (hitherto unknown) bodies.

The processing on the high level consists in the aggregation of these parts, i.e. in their grouping and labelling in such a way that lines, blobs and surfaces are assigned to specified objects which are present in the scene. On the basis of these assignments the description and identification of objects takes place.

The presentation of the problems of picture processing can be found e.g. in the review books and papers [2,3,4,5,9,10] and monographs [6,7,8].

In the simplified world of the tasks of industrial robots which manipulate objects that have a small number of stable positions, the vision problem can be restricted to considering the features of the silhouettes of these objects. The investigated object is then a two-dimensional silhouette. It frequently happens that the objects do not touch and occlude each other. Analysis of the picture on the subsequent levels is then greatly simplified. Within the low level we find the light and dark blobs which are each other's neighbours or holes in one another. The middle level algorithm assigns blobs to objects and, in this way, labels the background, the objects and their holes. The high level analysis consists in describing objects, i.e. finding their features, and in recognizing them with the use of these features, i.e. in assigning each object to the appropriate class.

As the features one can use numerical quantities, such as area or perimeter of the silhouette, logical data as the existence or non-existence of sharp vertices in contour, or structural information, as e.g. the graph of mutual location of holes. The ordered set of numerical features is usually called the feature vector, although formally it does not have to be a vector in the sense of linear algebra. An object described by its features, or the feature vector itself, is called a pattern.

The algorithm of recognizing a pattern, i.e. an object which has already been distinguished in the picture and described, will be the subject of our considerations in the present paper.

In respect of a very large volume of information contained in the brightness table, and because of frequent ambignities in the picture interpretation problems, which occur even in very simple cases, picture processing requires computationally effective algorithms. As in many applications, especially in control problems, the response time of the vision system should not be much longer than the "tv frame time", i.e. 1/50 s, this requirement is indeed very strict. Therefore, the algorithms should be simple (suited to hardware implementation) and effective at the same time.

The recognition algorithms can be roughly divided into symbolic processing algorithms and classic numerical algorithms. The algorithms from the first group enable modelling of complex processes of reasoning about the scene, but they necessitate for relatively sophisticated software, are time-consuming, and therefore their significance is mainly cognitive. Conventional methods, such as maximum likelihood function methods or minimum distance methods, are less flexible but, nevertheless, they are simpler and, in particular applications, more effective. This supports the need of studies on the methods which would be effective and simple from the numerical point of view, and which could restrict the way of deciding on the object assignment to classes to less extent than the classic methods do.

The method presented here is the modified version of the distance function method.

2. The conventional minimum distance method

In the classic minimum distance method [11] the notion of distance between an object and a class is introduced. This distance reflects the degree of dissimilarity of a recognized objects to the objects belonging to a class. The class is represented by a chosen object, denoted a class prototype. The distance function depends on the differences between the features of an object and a prototype, and is an increasing function of those differences. It can have the properties of a metric, but this is not necessary. The recognized object belongs to the class which prototype is the nearest, in the sense of the distance such defined (Eq.(2.1)), or to the class for which the distance between the object and the prototype is not larger then a threshold value, specified for each class (2.2)

(2.1)[(1)] $$Obj_i \in Cla_j \iff \forall_k \; Dst(Obj_i, Prt_j) \leq Dst(Obj_i, Prt_k) \;,$$

(2.2) $$Obj_i \in Cla_j \iff \quad Dst(Obj_i, Prt_j) \leq Thr(Cla_j) \;,$$

where

$$i = 1, ..., NumObj; \qquad j, k = 1, ..., \; NumCla.$$

The class Cla_i is then a pair: prototype – threshold:

$$Cla_i \equiv (Prt_i, Thr_i).$$

3. Difficulties

In the application of the conventional minimum distance method two difficulties arise. They are related to the problem of taking into account the different importance of the features and to the problem of computational complexity.

In some practical problems the objects are described by a large number of features, and each of these features can have different meaning and different significance if the assignment of the object to a class is considered. While aggregating the influence of the differences of the features in one, common distance function, one should consider their mutual importance. If the features are real numbers, frequently the

---

[(1)] Notation and symbols according to the dictionary in Sect.8.

weight coefficients are introduced. Let us take an example of an Euclidean function for $NumFea$ features, used according to Eq. (2.2) with the threshold equal to (2.1):

$$Dst(Obj_i, Prt_j) = \sum_{k=1}^{NumFea} (Wgt_k * \Delta_{ij}Fea^k)^2 \leq 1 \,, \tag{3.1}$$

where

$$\Delta_{ij}Fea^k = Fea^k(Obj_i) - Fea^k(Prt_j).$$

Each weight Wgt_k can be made dependent on some estimator of the scatter of the corresponding feature Fea^k , e.g. on its standard deviation within a class:

$$Wgt_i = (Dev(Fea^i(Cla_j)))^{-1} \,. \tag{3.2}$$

In the feature space the locus of the points corresponding to the patterns belonging to one class, i.e. the region which represents this class, will have the shape of a hyperellipsoid (Fig.1) [11].

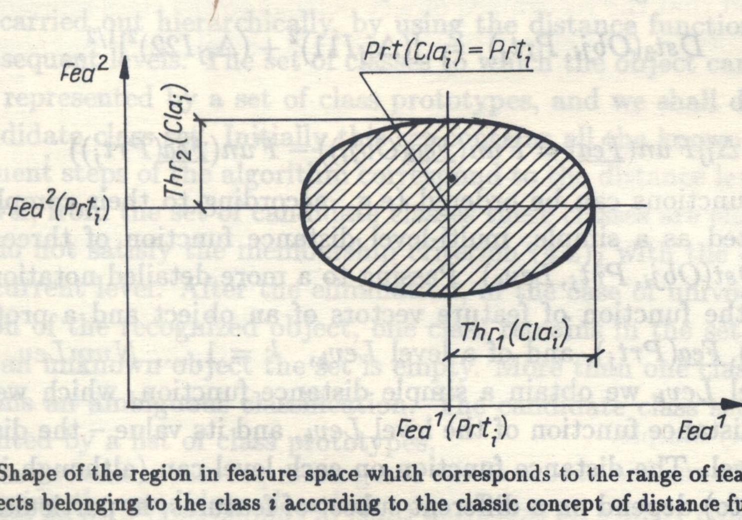

Fig. 1. Shape of the region in feature space which corresponds to the range of features of the objects belonging to the class i according to the classic concept of distance function.

If not all the features are real numbers with non-zero standard deviations then the Eq. (3.2) have no application. In particular, if the non-numeric features are used, then, although their differences can be defined so that they would belong to the set of reals, it is difficult to find the convincing interpretation of the coefficients in (3.1).

The choice of the nearest prototype for an object necessitates for calculating the distance between this object and the prototypes of all the classes. With the large number of features, the numerical complexity of the distance function becomes an important problem. On the basis of a single distance function it is impossible to restrict the number of calculations by rejecting the most "unlikely" classes earlier.

4. The multi-level distance function

To overcome the aforementioned difficulties, we propose to replace a single distance function with a set of distance functions. Each of these functions serves to calculate the distance with respect to one or several chosen features, e.g.:

$$
\begin{aligned}
Dst_1(Obj_i, Prt_j) &= |\,\Delta_{ij} NumHol\,|, \\
Dst_2(Obj_i, Prt_j) &= |\,\Delta_{ij}(Per^2/Are)\,|, \qquad (4.1)\\
Dst_3(Obj_i, Prt_j) &= [(\Delta_{ij} I11)^2 + (\Delta_{ij} I22)^2]^{1/2},
\end{aligned}
$$

where

$$\Delta_{ij} Fun(\underline{Fea}) = Fun(\underline{Fea}(Obj_i)) - Fun(\underline{Fea}(Prt_j)) \ .$$

The functions can be ordered (e.g. according to their complexity) and treated as a simple, multi-level distance function of three arguments: $Dst(Obj_i, Prt_j, Lev_k)$. Passing to a more detailed notation, distance is the function of feature vectors of an object and a prototype $\underline{Fea}(Obj_i)$, $\underline{Fea}(Prt_j)$, and of a level Lev_k, $k = 1, ..., NumLev$. For a fixed level Lev_k we obtain a simple distance function, which we shall call the distance function of the level Lev_k, and its value – the distance of this level. The distance function on each level can (although it does not have to) depend on a different subset of features, in particular – on a single feature. The example of the function (4.1) illustrates this. It should be stressed that the distance function is not a vector function, but an ordered set of scalar functions.

With the distance function defined in this manner the condition of membership of an object Obj_i in a class Cla_j $(4.2)_1$ will be the conjunction of the conditions $(4.2)_2$ for the functions of all the levels:

$(4.2)_1 \quad Obj_i \in Cla_j \Longleftrightarrow \forall_k [Obj_i \in Cla_j]_{Lev_k}$,

where

$(4.2)_2 \quad [Obj_i \in Cla_j]_{Lev_k} \Longleftrightarrow Dst(Obj_i, Prt_j, Lev_k) \leq Thr(Cla_j, Lev_k)$,

or in one expression

$(4.2)_3 \; Obj_i \in Cla_j \Longleftrightarrow \forall_k \; Dst(Obj_i, Prt_j, Lev_k) \leq Thr(Cla_j, Lev_k)$.

The class Cla_i is now a pair: prototype – threshold vector: $Cla_i \equiv (Prt_i, \underline{Thr}_i)$, and in more detail: vector of the prototype – threshold vector: $Cla_i \equiv (\underline{Fea}(Prt_i), \underline{Thr}_i)$.

5. The hierarchical recognition algorithm

In the recognition algorithm based on the multi–level distance function, the class to which the recognized object belongs is found by eliminating those classes to which this object does not belong. Such elimination is carried out hierarchically, by using the distance functions of all the subsequent levels. The set of classes to which the object can belong will be represented by a set of class prototypes, and we shall denote it the candidate class set. Initially this set contains all the known classes. Subsequent steps of the algorithm correspond to the distance levels. On each level, from the set of candidate classes these classes are eliminated which do not satisfy the membership criterion $(4.2)_2$ with the distance of the current level. After the elimination, in the case of univocal classification of the recognized object, one class remains in the set. In the case of an unknown object the set is empty. More than one class in the set means an ambiguous classification. The candidate class set can be represented by a list of class prototypes.

Forming the candidate class list

Step 1. Form the candidate class list of the prototypes of all the known classes and set the lowest level Lev_k, $k = 1$.

Elimination loop

STEP 2. Calculate the distances of level Lev_k between the recognized object and the prototypes from the list.

STEP 3. Eliminate from the list these prototypes which do not satisfy the inequality $(4.2)_2$ for Lev_k .

STEP 4. If Lev_k is the last level Lev_{NumLev} or if the list is empty, go to Step 5; otherwise, set the next level Lev_k, $k = k + 1$, and go to Step 3.

Sorting

STEP 5. If the obtained list contains more than one class, sort it, e.g. according to the increasing distance of a chosen level, or some other criterion. The nearer position of a prototype on the list, the larger the likelihood that the recognized object belongs to the class represented by this prototype. The empty list means that the object is unknown.

6. REMARKS

The recognition algorithm proposed here operates in a way as if in the set of classes there existed a hierarchy which would provide for the possibility of excluding the whole groups of classes at a time. Actually, such hierarchy is not introduced, and the functioning of the recognition process results merely from the structure of the algorithm, which is noticeably simple.

The hierarchic nature of the recognition process makes it possible to significantly reduce its execution time, in comparision with an algorithm with a single distance function. This time depends on the effectiveness of the functions on subsequent levels. The identification process is quicker when the functions used on lower levels are computationally simpler and when they eliminate classes from the candidate class set more effectively.

The requirements of small computational complexity and high effectivity usually contradict each other. Therefore, minimization of the recognition time necessitates for the right choice of functions and their sequence. The promising results presented in Sect.7 indicate that the detailed study of this problem is necessary [12,13].

As in the set of classes no ordering is introduced, including a new class into this set does not lead to the necessity of analyzing the differences and similarities of various class groups.

The set of candidate classes containing a single class, obtained as a result of recognition, means that the object has been univocally classified. The empty set means that the object is unknown, i.e. that it does not satisfy all the conditions $(4.2)_2$ for any class. If the set contains more than one class, the assignment of the object is ambiguous. This takes place if the regions in the feature space which represent classes are not disjoint, i.e. the classes are not separate. In this case there arises the necessity of introducing in the set of candidate classes obtained from the process of recognition an ordering which would reflect the degree of conformity of the recognized object with the prototypes of subsequent classes. This degree of conformity can not be identified with the probability of the object membership in the classes, as within the frames of the distance-type classification such probability is not defined.

To introduce an order in the candidate class set we can use the distance of a chosen level, which we consider the most decisive. Also a general function, which takes into account the distances of all the levels, except those with zero threshold, can be applied. For example, the formulae (3.1) and (3.2) can be used, with the features replaced by the distance functions at the subsequent levels and with the standard deviations of features replaced by the thresholds for these levels.

Small computational complexity of the proposed algorithm makes its implementation easy, not only in any higher level language, but also in hardware. The changes of the number, sequence and kind of the distance functions, which are the main factors which influence the recognition results, do not necessitate for any modifications of the algorithm.

The shape of the regions of the feature space which correspond to classes depend only on the kind of the distance functions. If the functions are affine with respect to the features of an object and a pattern, these regions are cubicoidal, as it is in the case of the distance functions according to Eqs.$(4.1)_1$ and $(4.1)_2$ (Fig.2). The cubicoides can be

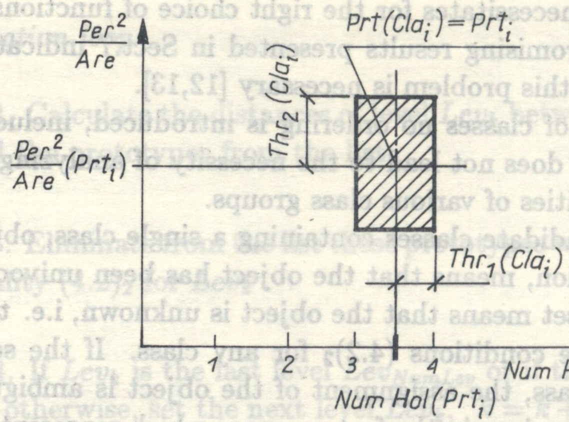

Fig. 2. Shape of the region in feature space which corresponds to the range of features of the objects belonging to the class i, according to the two-level distance function - example of Eqs.$(4.1)_1$ and $(4.1)_2$.

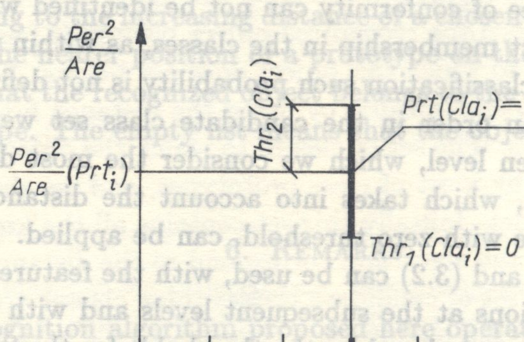

Fig. 3. The case of zero admissible deviation of one of the features.

degenerate (Fig.3). The admissible deviations of various features are independent.

On the basis of the multi-level distance function, more elaborate recognition algorithms can be built, as e.g. an algorithm in which only these features of the recognized object are calculated which are indispensable for its correct recognition.

7. Computational Example

The above presented concept of the multi-level distance function and the recognition algorithm based on this concept has been implemented in the program LOOK [15] which recognizes two-dimensional, separated objects in the two–brightness–level picture.

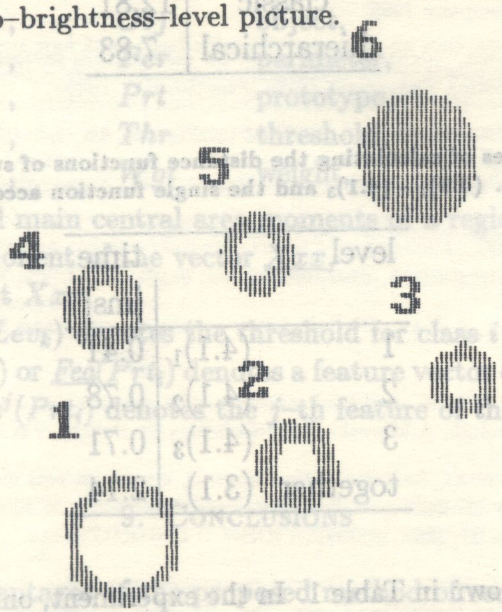

Fig. 4. Silhouettes of objects belonging to the classes used in the example: *1* – M16 nut (wrench 9); *2* – M8 nut (wrench 14); *3* – M6 nut (wrench 11); *4* – washer 5 (ϕ_o 14 mm, ϕ_i 6.25 mm); *5*– tubular rivet 6 (ϕ_o 13 mm, ϕ_i 6 mm); *6* – 2 zł coin (ϕ 21 mm).

For six classes of objects, for which the example silhouettes are shown in Fig.4, the mean times of recognition carried out in two ways have been compared. First, the objects were recognized with the algorithm described in Sect.5, in which a three-level function according to Eqs. $(4.1)_1$ - $(4.1)_3$ was used. Then, the same objects were recognized with a single distance function, according to the class membership criterion (2.2). For the sake of comparability of the results, this single function was built according to Eq. (3.1), in which instead of three features, the three functions $(4.1)_1$ - $(4.1)_3$ were used. All the weights were set to 1.25, as the specific value had no influence on the execution time, and it was necessary to model the multiplication by a real number according to Eq. (3.1). In both cases, all the classes occurred to be mutually disjoint.

The mean recognition times, averaged for five objects belonging to

Table 1. Times of recognition in the classic and hierarchical algorithms.

algorithm	time [ms]
classic	12.81
hierarchical	7.83

Table 2. Times of calculating the distance functions of subsequent levels according to Eqs. $(4.1)_1$ - $(4.1)_3$ and the single function according to Eq. (3.1).

level		time [ms]
1	$(4.1)_1$	0.41
2	$(4.1)_2$	0.78
3	$(4.1)_3$	0.71
together	(3.1)	2.14

each class,are shown in Table 1. In the experiment, only the recognition times were taken into account; no times of calculating object features were included.

The times of calculating the distance functions for subsequent levels are compared in Table 2.

Application of the hierarchical recognition algorithm with the multi-level distance function in this example, made it possible to reduce the average time of recognition by more than 40 per cent.

8. Dictionary of the applied notations and abbreviations [14]

Are	area	,	*Lev*	level,
Cla	class	,	*Num...*	number of...,
Dev	deviation	,	*Obj*	object,
Dst	distance	,	*Per*	perimeter,
Fea	feature	,	*Prt*	prototype,
Fun	function	,	*Thr*	threshold,
Hol	hole	,	*Wgt*	weight,

$I11$, $I22$: second main central area moments of a region,
Xxx^i : i–th component of the vector $\underline{Xxx}$,
Xxx_i : i–th object Xxx,
e.g. : $Thr(Cla_i,\ Lev_k)$ denotes the threshold for class i on level k;

$\underline{Fea}(Prt(Cla_i))$ or $\underline{Fea}(Prt_i)$ denotes a feature vector of the prototype of class i and $Fea^j(Prt_i)$ denotes the j–th feature of this prototype.

9. Conclusions

The basic advantages of the proposed method of recognition are its very simple form and its hierarchical character, although in the set of classes no order or hierarchy is introduced. Therefore, in the learning process there is no need for any analysis of similarities and differences between the features of prototypes of various groups of classes, considered in the problem.

The proposed concept of the multi–level distance function and the related pattern recognition method is particularly effective in simple classification systems with large numbers of classes.

References

1. H.C.Barrow, J.Tennenbaum, *Computational vision*, Proc. IEEE, 69, 5, 572-593, 1981.
2. E.Charniak, D.McDermott, *Introduction to artificial intelligence*, Addison-Wesley Publishing Company 1985.
3. W.B.Gevarter, *Intelligent machines: an introductory perspective of artificial intelligence and robotics*, Englewood Cliffs, Prentice-Hall 1985.
4. M.Brady, *Computer vision*, Artificial Intelligence, 19, 1, 7-16, 1982.
5. N.R.Corby Jr, *Machine vision for robotics*, IEEE Trans., IE-30, 3, 282-291, 1983.
6. D.Marr, *Vision*, Freeman, San Francisco 1982.
7. D.Ballard, C.Brown, *Computer vision*, Englewood Cliffs, Prentice-Hall 1982.
8. A.Rosenfeld, A.Kak, *Digital picture processing*, vol.1 and 2, Academic Press, New York 1982.
9. J.Malec, M.Wnuk, *Artificial intelligence in robotics*, in: Fundamental Problems of Technology, vol.25, A.Morecki [Ed.]: Robotics [in Polish], PWN.
10. L.J.Chmielewski, *Robotics, vision systems, expert systems and artificial intelligence*, in: Computer Modelling of Perception and Reasoning, W.Kosiński [Ed.], [in Polish], IFTR Reports 31/1987, pp.6-72.
11. J.T.Tou, R.C.Gonzalez, *Pattern recognition principles*, Advanced Books Program, Addison-Wesley Publishing Comp., Inc., Reading, Massachusetts 1974.
12. L.J.Chmielewski, *Time-optimal recognition method with the multi-level distance function*, Proc. 3rd Int. Conf. on Machine Vision and its Applications, Warwick, UK, July 21-23, 1989; IEE Publication No. 307.
13. L.J.Chmielewski, *Feature selection and time minimization in the multi-level minimum – distance recognition algorithm*, Proc. 9th Conf. on Computer Methods in Mechanics, Rytro 16-20 May 1989 (Late Papers), Wyd. Politechniki Krakowskiej, 1989.
14. L.J.Chmielewski, *Proposition of the standard way of naming variables in the image analyzing software*, in: Computer Modelling of Perception and Reasoning - II, L.J.Chmielewski, W.Kosiński [Eds.], [in Polish], IFTR Report 42/1988, pp. 80-92.
15. L.J.Chmielewski, W.Kosiński, *Software for recognizing two-dimensional, fully visible objects in binary pictures* [in Polish], Scientific Report SCSS, TDC IFTR PAS, 1987.

Streszczenie

HIERARCHICZNA METODA ROZPOZNAWANIA OBIEKTÓW NA PODSTAWIE WIELOPOZIOMOWEJ FUNKCJI ODLEGŁOŚCI

Dla przezwyciężenia wad klasycznej odległościowej metody rozpoznawania wzorców, opartej na funkcji odległości o złożonej postaci, proponuje się wprowadzenie zbioru prostszych funkcji, tworzących wielopoziomową funkcję odległości. Metoda rozpoznawania zbudowana za pomocą takiej funkcji polega na stopniowej eliminacji klas, do których nie należy rozpoznawany wzorzec. Proponowany, bardzo prosty algorytm działa w taki sposób, jakby w zbiorze klas istniała hierarchia pozwalająca wykluczać kolejno całe grupy klas, podczas gdy hierarchii takiej nie wprowadza się; jest ona natomiast (niejawnie) zawarta w strukturze algorytmu rozpoznawania. Pozwala to na znaczne skrócenie czasu jego działania. Algorytm zastosowano do rozpoznawania obiektów w dziedzinie wizji komputerowej, gdzie wymaganie dużej szybkości obliczeń jest szczególnie istotne.

Резюме

ГЕРАРХИЧЕСКИЙ МЕТОД РАСПОЗНАВАНИЯ ОБЪЕКТОВ НА ОСНОВЕ ФУНКЦИИ РАССТОЯНИЯ С МНОГИМИ УРОВНЯМИ

Для устранения недостатков классического метода распознавания образцов в расстояниях, опирающегося на функции расстояния сложного вида, предлагается введение множества более простых функций, образующих функцию расстояния с многими уровнями. Метод распознавания, построенный при помощи такой функции, заключается в постепенном исключении классов, к которым не принадлежит распознаваемый образец. Предлагаемый, очень простый, алгоритм действует таким образом, чтобы в множестве классов существовала герархия, позволяющая последовательно исключать целые группы классов, тогда как такой герархии не вводится ; она же содержится (неявно) в структуре алгоритма распознавания. Это позволяет значительно сократить время его действия. Алгоритм применен к распознаванию объектов в облости компьютерного видения, где требование большой скорости расчетов особенно существенно.

POLISH ACADEMY OF SCIENCES
INSTITUTE OF FUNDAMENTAL TECHNOLOGICAL RESEARCH

Received August 23, 1989.

S t r e s z c z e n i e

HIERARCHICZNA METODA ROZPOZNAWANIA OBIEKTÓW NA PODSTAWIE WIELOPOZIOMOWEJ FUNKCJI ODLEGŁOŚCI

Dla przezwyciężenia wad klasycznej odległościowej metody rozpoznawania wzorców, opartej na funkcji odległości o złożonej postaci, proponuje się wprowadzenie zbioru prostszych funkcji, tworzących wielopoziomową funkcję odległości. Metoda rozpoznawania zbudowana za pomocą takiej funkcji polega na stopniowej eliminacji klas, do których nie należy rozpoznawany wzorzec. Proponowany, bardzo prosty algorytm działa w taki sposób, jakby w zbiorze klas istniała hierarchia pozwalająca wykluczać kolejno całe grupy klas, podczas gdy hierarchii takiej nie wprowadza się; jest ona natomiast (niejawnie) zawarta w strukturze algorytmu rozpoznawania. Pozwala to na znaczne skrócenie czasu jego działania. Algorytm zastosowano do rozpoznawania obiektów w dziedzinie wizji komputerowej, gdzie wymaganie dużej szybkości obliczeń jest szczególnie istotne.

Р е з ю м е

ИЕРАРХИЧЕСКИЙ МЕТОД РАСПОЗНАВАНИЯ ОБЪЕКТОВ НА ОСНОВЕ ФУНКЦИИ РАССТОЯНИЯ С МНОГИМИ УРОВНЯМИ

Для устранения недостатков классического метода распознавания образцов в расстояниях, опирающегося на функции расстояния сложного вида, предлагается введение множества более простых функций, образующих функцию расстояния с многими уровнями. Метод распознавания, построенный при помощи такой функции, заключается в постепенном исключении классов, к которым не принадлежит распознаваемый образец. Предлагаемый, очень простый, алгоритм действует таким образом, чтобы в множестве классов существовала иерархия, позволяющая последовательно исключать целые группы классов, тогда как такой иерархии не вводится; она же содержится (неявно) в структуре алгоритма распознавания. Это позволяет значительно сократить время его действия. Алгоритм применен к распознаванию объектов в области компьютерного видения, где требование большой скорости расчетов особенно существенно.

POLISH ACADEMY OF SCIENCES
INSTITUTE OF FUNDAMENTAL TECHNOLOGICAL RESEARCH

Received August 22, 1989.