Typical Segment Descriptors: A New Method for Shape Description and Identification

Nancy Aimé Alvarez-Roca [1], José Ruiz-Shulcloper [2], and Jose M. Sanchiz-Marti [3]

[1] Departamento de Ciencia de la Computación
Universidad de Oriente, Santiago de Cuba, Cuba
aime@csd.uo.edu.cu
[2] Laboratorio de Reconocimiento de Patrones,
Instituto de Cibernética, Matemáticas y Física, La Habana, Cuba.
recpat@cidet.icmf.inf.cu
[3] Departamento de Ingeniería y Ciencia de los Computadores,
Universidad Jaume I, Castellón, Spain.
sanchiz@uji.es

Abstract. In this paper we introduce a new method for recognizing and classifying images based on concepts derived from Logical Combinatorial Pattern Recognition (LCPR). The concept of Typical Segment Descriptor (TSD) is introduced, and algorithms are presented to compute TSDs sets from several chain code representations, like the Freeman chain code, the first differences chain code, and the vertex chain code. The typical segment descriptors of a shape are invariant to changes in the starting point, translations and rotations, and can be used for partial occlusion detection. We show several results of shape description problems pointing out the reduction in the length of the description achieved.

1 Introduction

Recognition of 2D objects is an important task useful in many machine vision applications and research areas such as robotics and computer vision [1, 2]. A 2D shape is a feature often used for its distinctive classification power. A shape is what remains of a region after disregarding its size, position and orientation in the plane [3]. Non-numeric shape description methods search representations (e.g. a chain code, a graph) of the original shape so that only important characteristics are preserved. Other shape description techniques generate numerical descriptors given as feature vectors.

The required properties of a shape description scheme are invariance to translation, rotation an scaling. Shape matching or recognition refers to methods for comparing shapes. Usually, given a group of known objects, the identical or most similar objects in a scene must be found. There are many imaging applications where scene analysis can be reduced to the analysis of shapes [4], though effectively representing shapes remains one of the biggest hurdles to overcome in the field of automated recognition.

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 512–520, 2003.

In this paper we introduce a new boundary-based method for recognizing and classifying images originated from concepts of Logical Combinatorial Pattern Recognition (LCPR). The concept of Typical Segment Descriptor (TSD) is introduced. Using a shape's Freeman chain code its TSDs are computed, so they inherit the data reduction property of this representation. An algorithm for computing the TSD set from closed shapes is given.

Conversion (mapping) of an analog image onto a discrete one (digitization) is based on several assumptions [5]. It is assumed that the acquisition of an image is done using a set of physical captors, which could be modeled by a set of subsets of the continuous plane. The simplest idea is to assume a discrete partition of the plane. If only partitions involving regular polygons are considered, the number of different partitions is reduced to three: triangles, squares, or hexagons. The selection of the type of partition determines differences in concepts like neighborhood, adjacency, and connectivity. In this work we assume that partition is in regular squares and the algorithms presented assume closed boundary shapes.

Recently, machine learning and symbolic processing tools have been extended to Image Processing problems. New image representation concepts have been developed [6, 7]. The new method presented here uses ideas from LCPR.

Chain codes are frequently used for image representation since they allow considerable data reduction. The first approach for representing digital curves was introduced by Freeman [8]. By means of this representation several properties of arbitrary planar curves can be determined: moments, inertial axes, etc. [9]. Curves are encoded as line segments that link points of a rectangular grid. These points are the grid points closer to the curve. This process is called *chain encoding*.

Many authors have used chain coding for shape representation [2, 10], a normalization of the code with respect to the starting point is achieved by using shape numbers [3]. In [11] contours of handwritten characters are chain coded and recognition cost and accuracy are reported.

A new chain code for shapes composed of regular cells is defined in [12]. It is called *Vertex Chain Code* (VCC), and is based on the number of cell vertices that are in touch with the bounding contour of the shape. Concepts of VCC are extended for representing 3D shapes in [13], producing a curve descriptor invariant to translation.

Concepts from LCPR are used for image identification in [14], where a method for solving supervised pattern recognition problems using binary descriptors is reported. A generalization can be achieved by transforming numerical descriptors into k-valued sets so that k-valued logic tools can be used.

If some features of an object take values that cannot be found in the descriptions of objects of the remaining classes, then such a sequence of values is called a *descriptor*. If a certain descriptor loses this property when a feature is not included, then it is called a *Non-Reducible Descriptor* (NRD) [15].

In [16] an algorithm (KORA) is reported to select the features that form a minimal descriptor of every object in a database of descriptors. It has been extended and used on non-image-like data [17, 18] and is used in [14] for recognition of objects in raw images. In this latter work the concept of sub-description of an object is transformed in a fragment of an image. Considering each image as a one-dimensional array a learning matrix is formed. Differences of corresponding pixels are used to conform a dissimilarity matrix. In this manner, the concept of feature is lost and the set of co-

lumns selected as descriptive attributes change if transformations such as rotations, translations or scaling are applied.
NRDs are reported as a concept for representing a minimal set of characteristics that can be used for object recognition, the TSD proposed here complies with the same objective but, instead of using the whole learning matrix, only the chain code is used.
This paper is organized as follows: in Section 2 TSDs are defined and their main properties discussed. Section 3 presents results, and in Section 4 conclusions are drawn.

2 Typical Segment Descriptors

Suppose that a set of segmented binary 2D-images (shapes) $X_1, X_2,\ldots,X_n$ is known. For each shape, the chain code is computed based on δ-connectivity, δ=4,8. Each δ-chain has a *first difference* (*derivative*) D(X_1), D(X_2),…,D(X_n) associated. The derivative of a shape is a sequence of codes representing changes of direction. The length of the derivative D(X) will be denoted |D(X)|. It is assumed that objects can be rotated in angles $k * \frac{360}{\delta}$, with k integer, without change in the derivative.

Definition. A sequence $y_1y_2\ldots y_p$ is a *p-segment* of a circular sequence of codes D(X)=$x_1x_2\ldots x_m$ if p≤m and for some fixed j=1,…,m it is observed that $y_t=x_{j+t}$ with t=1,…,p. p is the *length* of the p-segment. For example 222310 is a 6-segment of 112223100223312.

Definition. Given two segments Θ and Ξ of length p and q respectively, Θ is a *sub-segment* of Ξ, if p≤q, and for a certain circular index rotation $y_t=x_t$, t=1,…,p. It is easy to observe that Φ=12112 is a sub-segment of Θ=112223100223312.

Definition. A p-segment Θ of the derivative D(X) is a *segment descriptor* of image X with respect to image Y, if there does not exist a p-segment Ξ of D(Y) such that Ξ=Θ. If, from Θ it is not possible to eliminate either the first element or the last one while keeping the property of segment-descriptor, then Θ is a *typical segment descriptor*. It will be denoted as *tsd*(X/Y) or simply *tsd*. A *tsd* of minimal length is called *minimal segment descriptor*. In general, it is not unique.

Figure 1 shows two shapes, the starting point from which their boundary was traversed, the path of their 4-chain codes which are given as shape numbers, and corresponding first differences.

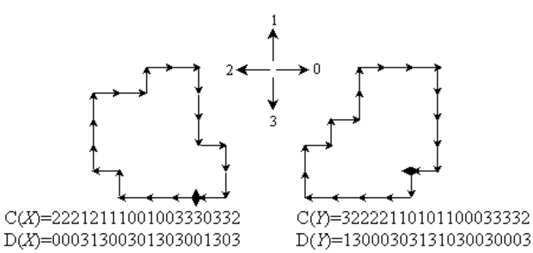

Fig. 1. Two shapes, their 4-chain codes (C) and first derivatives (D)

It can be verified that the Θ=30300 is a 5-segment descriptor of the shape *X* with respect to *Y*, while Ξ=3003 is not. Θ is not a typical segment-descriptor because 3030 (a subsegment) is a segment descriptor of *X* with respect to *Y*.

The previous definitions do not depend on the concept of the derivative. That is, they can be applied to chain codes or other representations directly.

The set of all typical segment descriptors of *X* with respect to *Y* will be denoted as TSD(*X/Y*). A *tsd* can be present only in X_i (but not in X_j (i≠j)), so it differentiates X_i with respect to the other shapes. The set of *tsd*s that fulfill this quality will be denoted TSD(X_i).

The properties of a shape's TSD set include:

Property 1. From the condition $X_i \neq X_j$, i≠j, i,j=1,…,n, it follows that each shape has at least one *typical segment descriptor*. This is obvious because in the worst case, for each X_i, a subchain of D(X_i) of length |D(X_i)|-1 is a *tsd*.

Property 2. Different *tsd*s have different discriminative power, since they can discriminate with respect to a different number of shapes. Based on this, a *weight* W(Θ) can be associated with each *tsd* Θ, being proportional to the number of discriminated shapes.

Property 3. Each *tsd* Θ∈TSD(*X/Y*) is linked to a unique subsequence in its original chain code which corresponds to a differentiating characteristic of *X*. Therefore, two or more occurrences of Θ inside TSD(*X/Y*) can be associated to the appearance of this characteristic with different or equal starting directions.

To highlight the differences between two occurrences of the same Θ in TSD(*X/Y*), we adopt the following conventions:

- Θ^0 : the first code of the subsequence of *X* that originates Θ is 0.
- Θ^1 : the first code of the subsequence of *X* that originates Θ is 1.
- …
- $\Theta^{\delta-1}$: the first code of the subsequence of *X* that originates Θ is δ-1.

Observe that, though a *tsd* Θ can appear as Θ^d (d∈{0,1,…, δ-1}) in TSD(*X/Y*), the same *tsd* can appear as Θ^f (f≠d) in other object having the same shape. In Figure 1, 01 is a *tsd* of *X* with respect to *Y* and the subsequence 330 originates it. That means that the four subsequences: 001, 112, 223, and 330, never can be present in *Y*. Figure 2 shows a graphical representation of these subsequences.

Note that not necessarily all of these sequences are simultaneously present in the shape, but they can eventually appear depending on the shape pose. In Figure 1, 001 and 330 are present, but not 112 nor 223.

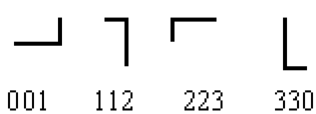

Fig. 2. Subsequences that can originate *tsd* 01 from TSD(*X/Y*)

The following holds for the shapes in Figure 1:

TSD(*X/Y*)={01^3, 01^0, 3030^1, 1303^3, 3030^0, 13003^1}

TSD(*Y/X*)={10^0, 131^0, 0303^2, 3031^2, 00300^0, 13000^2, 00030^2}

2.1 TSD Rotation Invariance

Property 4. Compatibility of TSD(*X/Y*): Let TSD(*X/Y*)= $\{\Theta_1^{d_1}, \Theta_2^{d_2}, ..., \Theta_q^{d_q}\}$, $d_i<\delta$. For any rotation of *X* with magnitude $\delta_j \in \{0, 1, \ldots, \delta\}$, all $\Theta_i^{d_i}$ in D(*X*) are in the form $\Theta_i^{d_i \oplus_\delta \delta_j}$ for i=1,…,q, where $\oplus_\delta$ represents the sum mod δ. This property states that each TSD(*X/Y*) has associated other δ-1 sets that contain the results of rotating the original set, and that it is possible to predict in what form every *tsd* will appear in each set. So, if shape *X* is rotated by δ_j, there will be a corresponding set TSD(*X/Y*) with a known form for every *tsd*.

Definition. *Compatibility* of *X* with respect to *Y*, Comp(*X/Y*), is the set of all possible n-tuples of d_j in all possible sets $\{\Theta_1^{d_1}, \Theta_2^{d_2}, ..., \Theta_q^{d_q}\}$, these sets can be obtained by rotating the original. A similar definition could be formulated for TSD(*X*). In Figure 1 it holds that the set of *tsd*s $\{1303^3, 3030^0, 13003^2\}$is not a compatible set. However, $\{1303^0, 3030^1, 13003^2\}$ is compatible.

2.2 Partial Occlusion Detection Using TSD

Definition. The number of occurrences of a *tsd* $\Theta^d \in$ TSD(*X/Y*) is denoted the *frequency* of Θ^d, α_d. The frequency of Θ is a δ-tuple, Freq(Θ)=$(\alpha_0,\alpha_1,\ldots,\alpha_{\delta-1})$. $\sum_{i=0}^{\delta-1} \alpha_i$ is the *absolute frequency of* Θ in TSD(*X/Y*) and will be denoted as $\|\Theta\|$.

Another important attribute is the relative order in which the *tsd*s appears. Θ is the *predecessor of* Ξ (Θ=Pre(Ξ)) if there is no other *tsd* in between. Then Ξ is the *successor* of Θ (Ξ=Suc(Θ)). In Figure 1, 3030^1 is the predecessor of 01^3.

Definition. A sequence of *tsd* in TSD(*X/Y*) is *connected* if they, all in sequence, form a subchain of the original chain code. The *connectivity of X*, denoted as Con(*X*), is the set of all connected sequences in TSD(*X/Y*). Each single *tsd* is a member of Con(*X*). In Figure 1, 3030^1, 01^3, 1303^3 and 3030^0 are connected.

Note that attributes of each Θ (tsd of X) such $\|\Theta\|$, Pre(Θ), Suc(Θ), as well as features of *X* such Comp(*X*) and Con(*X*) can be used for detecting the presence of the shape in a scene, even in case of partial occlusion. If a tsd of the shape if not detected due to an occlusion, but some of its attributes are checked, as well as other properties of the shape, then some certainty about the presence of the shape could be calculated. This is subject of our current research.

2.3 An Algorithm for the Computation of TSD(X/Y)

In order to determine the set of all possible *tsd*s of a shape with respect to other shapes, TSD(*X/Y*), two situations can be considered: each shape constitutes a class; each class is formed by more than one shape. In this latter case, the procedure is the same but only segments from different classes have to be compared [17].

Let D(X) and D(Y) be two derivatives. Each p-segment of X is tested as being a *tsd*. The set of tested segments are denoted Rev(X). The algorithm is:

Step 1. Let p = 1, TSD(X/Y)=$\emptyset$, Rev(X)= $\emptyset$.
Step 2. Let S_p be the next p-segment that can be extracted from D(X), i.e, a p-segment formed from an incremental starting position in D(X).
Step 3. If $S_p \in$ Rev(X) go to step 9.
Step 4. If $S_p \in$ TSD(X/Y) go to step 9.
Step 5. If some $\Theta \in$ TSD(X/Y) is a subsegment of S_p, go to step 8.
Step 6. If there is in D(Y) a p-segment E_p such that E_p=S_p go to step 8.
Step 7. Add S_p to TSD(X/Y), go to step 9.
Step 8. Add S_p to Rev(X).
Step 9. If it is not possible to extract another p-segment from D(X), make p=p+1.
Step 10. If p=|D(X)| end, else go to step 2.

The design of an algorithm for computation of the set TSD(X) is not complex. It is only necessary to find the intersection of all sets of *tsd*s of X with respect to the other shapes. The identification of X in an image will be easy if the shapes are isolated in the scene, finding at least one *tsd* from this intersection is sufficient to verify that shape X appears in the scene.
Additional steps should be added if the intersection is empty. Several alternatives can be used in these circumstances. A simple choice is to build a set taking one *tsd* from each set of *tsd*s of X with respect to the other shapes. In order to verify the presence of X in an image it will be necessary to find all the elements of the set created in this manner. In our experiments we build this set selecting *tsd*s with bigger weights.

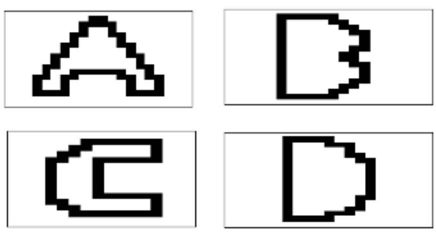

Fig. 3. Shape contours of letters A, B, C, D

3 Results

Figure 3 shows contours of some letters used in our experiments. *Tsd*s that differentiate B from the remaining shapes where computed using the proposed algorithm. Results are shown in Table 1. Note that the sum of the lengths of *tsd*s in TSD(B/X), with X={A,C,D}, is less than the length of C(B), so they describe shape B with respect to the others in a more compact form. Even in cases where the sum of *tsd*s is bigger than the length of C(B) it will not be necessary, in general, to compare the chain code or the derivatives with respect to all the *tsd*s.

Table 1. Sets of typical segment descriptors of shape B with respect to the other shapes.

TSD(B/A):	{33^2,	1000^0,	0001^0,	1313^0,	3131^3, 00131^0, 13100^0, 00000^1, 310013^3}	
TSD(B/C):	{33^2,	31001^3,	10013^0,	100001^1,	0001000^2,	0000000^1, 00000010^1}
TSD(B/D):	{33^2,	1313^0,	1001^0,	00131^0,	100001^1,	000010000^2}

Table 2 shows the sets of TSD(*X*), *X*={A,B,D}. In case of C the intersection of its sets of tsds with respect to the others letters was empty, so we use the alternative described in 2.3. Figure 4 illustrates the set of *tsd*s obtained for C and their compatibility with rotations in direction r.

Table 2. TSD sets calculated for A, B and D.

TSD(A/{B,C,D})	TSD(B/{A,C,D})	TSD(D/{A,B,C})
{11^1}	{33^2}	{ 01310^3}

Note that the presence of the shape can be verified finding only one *tsd*. This property is useful when the shapes are isolated in the scene. Other properties of the *tsd*s can be used if noise or occlusions affect the boundary: their *absolute frequency*, order of appearance, *connectivity*, etc.

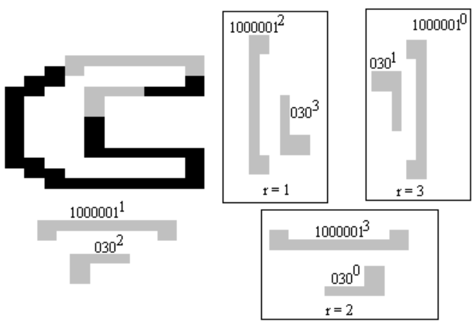

Fig. 4. TSD(C) elements and compatibility with rotations with magnitude r

4 Conclusions

A new method for description and identification of objects has been introduced. The concept of Typical Segment Descriptor is defined and its properties are enumerated.

The invariance of TSD to changes in starting point, translation, rotation and scaling, and its usefulness for partial occlusion detection are explained. The rotation applied to a known shape in the scene can be detected using the compatibility of the TSD.
Advantages of using TSD instead of chain codes are verified through examples with sensible reduction in the length of the description that will be used during identification.
Algorithms for the computation of the typical segment descriptors of one shape with respect to another, and to all shapes of a different class, are proposed. They can be used when the boundary is encoded using the Freeman's chain code or any other chain such as the Vertex chain code.
The efficiency of using the TSD approach for shape identification has been shown in synthetic scenes, obtaining encouraging results.
Suggestions for further work include to extend the use of typical segment descriptors to segmentation techniques, to study other properties that could be useful for detection of changes in scale.

References

1. Sanchiz, J.M.,Pla, F.: Feature Correspondence and Motion Recovery in Vehicle Planar Navigation. Pattern Recognition 32 (1999) 1961–1977
2. Agam, G., Dinstein, I.: Geometric Separation of Partially Overlapping Nonrigid Objects Applied to Automatic Chromosome Classification. IEEE Trans. Pattern Analysis and Machine Intelligence 19 (1997) 1212–1222
3. Bribiesca, E., Guzman, A.: How to describe Pure Form and How to Measure Diffrences in Shapes Using Shape Numbers. Pattern Recognition 12 (1980). 101–112
4. Pajares, G., de la Cruz, J.M.: Computer Vision, Digital Images and Applications. RA-MA, Madrid (2001)
5. Marchand-Maillet, S., Sharaiha, Y.M. : Binary Digital Image Processing: A Discrete Approach. Academic Press (2000)
6. Hatef, M., Kittler, J.: Combining Symbolic with Numeric Attributes in Multiclass Object Recognition Problems. Proceedings of the Second International Conference on Image Processing Vol. 3. Washington D.C (1995) 364–367
7. Hsu, J.C., Hwang, S.H.: A Machine Learning Approach for Acquiring Descriptive Classification Rules for Shape Contours. Pattern Recognition 30 (1997) 245–252
8. Freeman., H.: Techniques for the Digital Computer Analysis of Chain-encoded Arbitrary Plane Curves. Proceedings of Nat. Electr. Conf. 17 (1961) 421–432
9. Freeman., H.: Computer Processing of Line-drawing Images. Computing Surveys 6 (1974). 57–97
10. Loncaric, S.: A Survey of Shape Analysis Techniques. Pattern Recognition 31 (1998) 983–1001
11. Madhvanath, S., Kim, G., Govindaraju, V.: Chaincode Contour Processing for Handwritten Word Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 21 (1999) 928–932
12. Bribiesca, E.: A New Chain Code. Pattern Recognition 32 (1999) 235–251
13. Bribiesca, E.: A Chain Code for Representing 3D Curves. Pattern Recognition 33 (2000) 755–765
14. Valev, V.: A Method of Solving Pattern or Image Recognition Problems by Learning Boolean Formulas. Proceedings of 11th International Conference on Pattern Recognition. Vol 2. (1992) 359–362

15. Valev, V., Radeva, P.: Structural Pattern recognition by Non-Reducible Descriptors. Shape, Structure and Pattern Recognition. World Sci. Publ. (1995) 221–230
16. Djukova, E.V.: Pattern Recognition Algorithms of the Kora Type. Pattern Recognition, Classification, Forecasting – Mathematical Techniques and their Applications. 2 (1989) 99–125
17. Lazo, M., Douglas, M., Quintana, T.: Tests for classes: An application on character recognition, Proceedings of III Iberoamerican Workshop on Pattern Recognition (1998) 229–236
18. Pico, R., Almagro, G., Mena, J.: Logical Combinatorial Pattern Recognition application for classification of pathological mushrooms of sugar cane, Proceedings of III Iberoamerican Workshop on Pattern Recognition (1998) 279–292