STUDIA INFORMATICA 2001
Volume 23 Number 2A (48)

Adam KAPRALSKI
Politechnika Śląska, Instytut Informatyki

DISTRIBUTED AND FOCUSED PATTERN RECOGNIZERS BASING ON RANK ANALYSIS

Summary. The CF DATABASES, the multiple coordinate system *n*-CCS and the rank analysis create mathematical foundations for development different type D-transforms used in order to represent planar shapes and curves. Then, the nesting classes of such curves and shapes are investigated basing on projection of general distributed or focused representations. Sequential or distributed or parallel patterns and recognition algorithms with low asymptotic complexities are given.

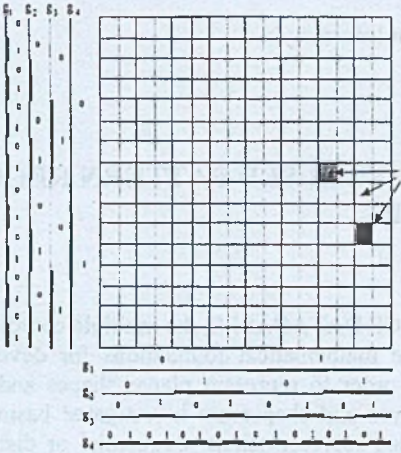
ROZPROSZONE I SKUPIONE WZORCE KSZTAŁTÓW ORAZ METODY ICH ROZPOZNAWANIA W OPARCIU O ANALIZĘ RANKINGOWĄ

Streszczenie. W pracy przedstawiono transformaty D płaszczyznowych krzywych i kształtów w oparciu o podstawy matematyczne rozwinięte na bazie CF BAZDANYCH, systemu wielokrotnych współrzędnych płaszczyznowych *n*-CCS oraz analizy rankingowej. Struktury reprezentujące transformaty D stanowią skupione oraz rozproszone wzorce kształtu, są wykorzystane dla systemów rozpoznawania oraz klasyfikacji figur lub krzywych płaszczyznowych. Przedstawione metody i algorytmu cechuje niska złożoność asymptotyczna i czasowa, zwłaszcza gdy komponentami systemu są maszyny do głębokiego wyszukiwania.

1. The scene representation

We are concerned with 2D scene that contains a set of rectangles (pixels) O. The set O is the domain of the following characteristic functions g_i^x, g_i^y ; $g_i^x : O \rightarrow \Gamma_i^x$, $g_i^y : O \rightarrow \Gamma_i^y$; $1 \le i \le k$ that are organized in two ordered strings $< g_i^x >$, $1 \le i \le k$ and $< g_i^y >$, $1 \le i \le k$. Each codomain Γ_i^x and Γ_i^y is well ordered. The linear order of the characteristic functions and the

orders of each codomain create lexical order of the objects $<\gamma_1^x, \gamma_2^x..., \gamma_k^x>$ and $<\gamma_1^y, \gamma_2^y..., \gamma_k^y>$, where $\gamma_i^x \in \Gamma_i^x$, $\gamma_i^y \in \Gamma_i^y$ [3]. Apart of that we define an attribute function $f: O \rightarrow \Phi$ that maps pixels o into a single value φ, for the considered case the values of function f represent the gray scale.



Rys. 1. The representation of 2D scene
Fig. 1. Reprezentacja sceny dwuwymiarowej

In Fig.1. we have illustrated this basic assignments and for the case we have $k = 4$. For the following considerations we assume $\Phi=\{0,1\}$. So, the general data structure representing the given scene is the following table of records that each record corresponds into one pixel.

f	X Coordinates	Y Coordinates
⋮	⋮	⋮

The data structure is organized into two CF DATABASES as follows.

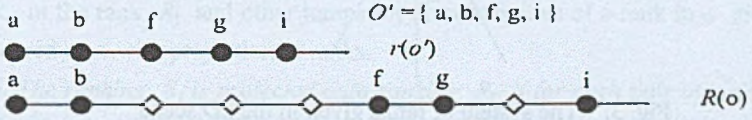
f	CF DATABASE				CF DATABASE			
	g_1^x	g_2^x	...	g_k^x	g_1^y	g_2^x	...	g_k^x
⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮

The field devoted to the function f is shared by these two CF DATABASES. Since $<\gamma_1^x, \gamma_2^x ..., \gamma_k^x>$ and $<\gamma_1^y, \gamma_2^y ..., \gamma_k^y>$ can be identified with choice functions h of indexed families $<\Gamma_i^x>$, $1 \le i \le k$ and $<\Gamma_i^y>$, $1 \le i \le k$, respectively. So, the successor can be evaluated following the general algorithm NEXT [4], for evaluation of the predecessor, we can also use NEXT assuming only the reversal lexical order. For instance, if we have <0,0,1,1>, then its predecessor is <0,0,1,0>, while successor is <0,1,0,0>. We use the following denotations:

$<\gamma'_1, \gamma'_2 ..., \gamma'_k> = pred(<\gamma_1, \gamma_2 ..., \gamma_k>)$
$<\gamma''_1, \gamma''_2 ..., \gamma''_k> = succ(<\gamma_1, \gamma_2 ..., \gamma_k>)$.

Since the mapping $O \rightarrow \{<\gamma_1, \gamma_2 ..., \gamma_k>\}$ is an onto mapping, so the linear order on the set $\{<\gamma_1, \gamma_2 ..., \gamma_k>\}$ generates a partial order on the set O. We are concerned now with a subset $O' \subset O$ that $o' \in O'$ if $f(o') = \alpha$. The operations NEXTMAX(o) and NEXTMIN(o) create the partial order on O' that is a suborder of the mentioned lexical order on O. For the Depth Search Machine the operations NEXTMAX(o) and NEXTMIN(o) can be performed in time O(1) [3], while for the von Neumann's model of computation different implementations of these operations are possible. Briefly speaking the operations NEXTMAX(o) and NEXTMIN(o) retrieve successor and predecessor of a given $o \in O'$. Therefore, the operations $succ(<\gamma_1, \gamma_2 ..., \gamma_k>)$ and $pred(<\gamma_1, \gamma_2 ..., \gamma_k>)$ determine the position of the corresponding element o in the string of elements O, while the operations NEXTMAX(o') and NEXTMIN(o') determine the position of the element o' in the string of elements O'. The position of an element $o \in O$ (in the string of elements O), we call the rank $R(o)$. Similarly, the position of an element $o' \in O'$ (in the string of elements O'), we call the rank $r(o')$. Illustration of the investigated concepts is given in Fig. 2.

Let O be the set of characters of the Roman alphabet.



Rys. 2. Mutual relation between ranks $R()$ and $r()$
Fig. 2. Wzajemna zależność pomiędzy rankingami $R()$ oraz $r()$

For the following considerations each CF DATABASE is identified with the ranks $R_x()$ and $r_x()$ or with $R_y()$ and $r_y()$, respectively. The operations NEXTMAX, NEXTMIN are applied into the ranks of type $r()$, while the operations *succ*, *pred* are applied into ranks of type $R()$. We emphasize that the given rank systems are the distributed rank systems [5] since we do not know a 'priori the position of a given element o or o' in the corresponding strings of the elements O or O', we can evaluate only the attributes of the neighbors of each element instead.

2. The multiple rank systems

We extend our coordinate system investigating multiple-coordinate system *n-CCS* for 2D scene [3]. For each axis of the *n-CCS* we investigate the CF DATABASE organization in the similar manner as we did for the X and Y axes, see Fig.3. However, we have the ordered set of characteristic functions $<g_i>$, $1 \le i \le k$ assigned into each axis but for our consecutive considerations we do not need to recall to any CF DATABASE organization recalling into

corresponding ranks $R_i()$ and $r_i()$ instead. Then, the data structure organized for representing pixels is given as follows.

f	R_1	R_2	...	R_n
	CF DATABASE	CF DATABASE	...	CF DATABASE
φ	$<\gamma_1^1, \gamma_2^1, \ldots, \gamma_k^1>$	$<\gamma_1^2, \gamma_2^2, \ldots, \gamma_k^2>$	...	$<\gamma_1^n, \gamma_2^n, \ldots, \gamma_k^n>$
⋮	⋮	⋮	...	⋮

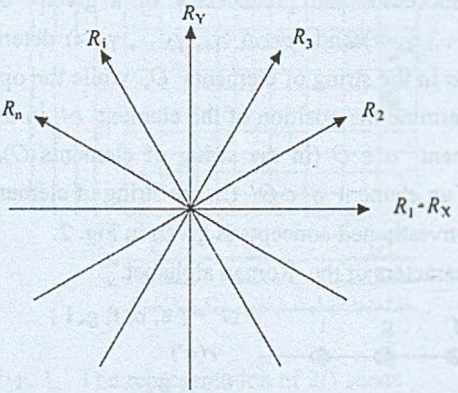


Fig. 3. The system of ranks given in the 2D scene
Rys. 3. System rankingów reprezentujących 2D scenę

Observe, we do not need to represent the rank systems $r_i()$ in the table since we get it taking the rank $R_i()$ and fixing a value $\varphi \in \Phi$ Then, the corresponding subset O' is selected from O and usage of the operators NEXTMAX and NEXTMIN for the CF DATABASE that is dedicated into $R_i()$ defines simultaneously the rank $r_i()$. So, operating on the rank r_i does not require any additional assignment.

3. Rank analysis

The given multiple rank system create only a part of the rank system given or produced in the plane for consecutive evaluations. Assume now we have a complete rank system containing individual ranks of any origin. We recall now the basic concepts concerning the general rank analysis [5]. In order to simplify the matter we are concerned only with the ranks being in monotonic accordance, however extensions of the following definitions for the case of the opposite monotony is enough obvious.

Definition 1. *The rank r_i is discontinuous in R_j at the right side of o if $NEXTMAX(r_i(o)) \neq succ(R_j(o))$, $1 \leq i \leq n$, $1 \leq j \leq n$.*

Definition 2. *The rank r_i is discontinuous in R_j at the right side of o if* $NEXTMIN(r_i(o)) \neq pred(R_j(o))$, $1 \leq i \leq n$, $1 \leq j \leq n$.

We use denotation $NEXTMAX(r_i(o))$ instead previously used $NEXTMAX(o)$ or NEXTMAX in order to emphasize that we perform the operation with respect to the given rank r_i. Similarly, the given denotations specify exactly what rank is taken into account for evaluation the corresponding ''neighbors in the rank''. Observe, for the general definition of the discontinuity the index i can equal j. In the similar way we can also consider the discontinuity of a rank R_j in R_i, $i \neq j$, however more cases we have to consider.

Definition 3. *The rank R_i is discontinuous in R_j at the right side of o if* $succ(R_i(o)) \neq succ(R_j(o))$, $1 \leq i \leq n$, $1 \leq j \leq n$.

In order to measure the discontinuity we investigate the jump concept.

Definition 4. *The right side jump J of r_i in R_j at o is defined to be the amount of ranks $R_j()$ between $NEXTMAX(R_j(o))$ and $succ(r_i(o))$.*

For instance, consider the situation given in Fig. 2., the jump J at ''b'' equals to 3, while the jump at ''g'' equals to 1. In the similar way we define the jump for a discontinuity of a given rank R_i in the rank R_j and other jumps for discontinuities of a rank in a given rank. We are concerned now with projection of ranks.

Definition 5. *The ranking R_i is projected onto ranking R_j if for each pair of elements o_w, o_z the following inclusion holds* $(R_i(o_w) = R_i(o_z)) \Rightarrow (R_i(o_w) = R_j(o_z))$. *Moreover,* $(R_i(o_w) \neq R_i(o_z))$ *does not imply* $(R_j(o_w) \neq R_j(o_z))$ *neither* $(R_j(o_w) = R_j(o_z))$ *implies* $(R_j(o_w) = R_j(o_z))$.

If R_i is projected onto R_j and if the R_i and R_j are in accordance, then compact ranges of ranks R_i correspond into one rank R_j. Then, the projection can be specified in a simple table that each row represents an elementary assignment of a range of ranks in R_i into a single rank in R_j. The concepts of the discontontinuity, of the jump and of the projection of R_i onto R_j represent structural properties of an arbitrary set O' that $O' \subseteq O$.

4. Neighborhood

The goal of this chapter is to explain usage of the concepts of the rank analysis for defining the shape and its edge or border. Given are the set of elements O and partial orders $<^i$, $<^j$ on O that correspond into the ranking systems R_i and R_j. Suppose, we have the set of pixels O and the rank system $< R_x , R_y >$. Let $R_i()$ be a fixed value of R_i for $i=x$ or $i=y$. Since the system of partial orders is orthogonal, so to each pair of ranks $(R_x(), R_y())$ corresponds uniquely an element $o \in O$.

Definition 6. *We say that* $(R_x(),R_y())$ *and* $(R'_x(), R'_y())$ *specify two neighboring elements* o *and* o' *if for every* i *we have* $(R_x() = R'_x()$ or $R_x() = succ(R'_x()))$ or $R_x() = pred(R'_x()))$ *and* $(R_y() = R'_y()$ *or* $R_y() = succ(R'_y()))$ *or* $R_y() = pred(R'_y()))$, *where* $(R_x(), R_y()) \neq (R'_x(), R'_y())$.

With $o = N(o')$ we denote that o and o' are the neighbors. The set O' create the pixels that $f(o') = \alpha$.

Definition 7. *We say that the subset* O' *is strongly compact if for each pair of elements* (o_p, o_p) *that* $o_p \in O'$, $o_q \in O'$ *there is a connecting string of neighbors i.e.,* $o_p = N(o_1)$, $o_1 = N(o_2)$, ..., $o_s = N(o_{s-1})$, $o_q = N(o_s)$ *that for each* o_i, $1 \leq i \leq s$, *we have* $o_i \in O'$.

The rank system $<R_x(), R_y()>$ induces the corresponding rank system $< r_x(), r_y()>$ on O'. Given is the maximal acceptable jump system $<J_x, J_y>$ corresponding to the ranks $<r_x(), r_y()>$ in $<R_x(), R_y()>$, respectively i.e.. the jump $r_i()$ in $R_i()$ equals J_i.

Definition 8. *We say that an element* o' *specified by* $(R'_x(), R'_y())$ *belongs to the surrounding of a given* $o \in O'$ *specified by* $(R_x(), R_y())$ *if* $| R_i() - R'_i()| \leq J_i$ *for every* $i \in \{x,y\}$.

Example 1. *Consider the* Fig. 4. *Suppose, we have given* $J_i = 2$, $1 \leq i \leq k$, *then the elements in* Q *create the surrounding, while the elements* a, b *and* c *do not belong to* Q.

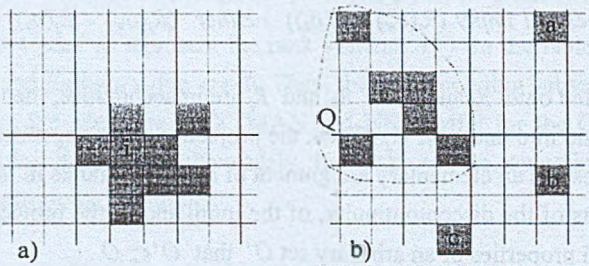


Fig. 4. The planar shapes given as: a) compact set, b) surrounding
Rys. 4. Typy kształtów płaszczyznowych: a) zwarty zbiór, b) otoczenie

We define two types of shapes. The first shape corresponds to the compact subset O' of O, while the second type of shapes corresponds to a given surrounding Q for a given pair of acceptable jumps (J_x, J_y). Correspondingly, we define two types of borders of a given shape.

Definition 9. *The border of a compact set of elements is a subset* O'' *of* O' *such that* o *represented by* $(R_x(), R_y())$ *belongs to* O'' *if exists* $o' \notin O'$ *represented by* $(R'_x(), R'_y())$ *and if* $R'_i() = succ(R_i()) or R'_i() = pred(R_i())$, *while* $R'_j() = R_j()$, *for* $j \neq i$, *where* $j,i \in \{x,y\}$.

Similarly, for a given surrounding Q we define the border as follows.

Definition 10. *The border of a given surrounding* Q *is a subset* $Q' \subset Q$ *that o represented by* $(R_x(), R_y())$ *belongs to* Q' *if there exists* $o' \notin Q$ *represented by* $(R'_x(), R'_y())$ *and if* $|R'_i() = R_i()| = J_i$, *while* $R'_j() = R_j()$, *for* $j \neq i$, *where* $j,i \in \{x,y\}$.

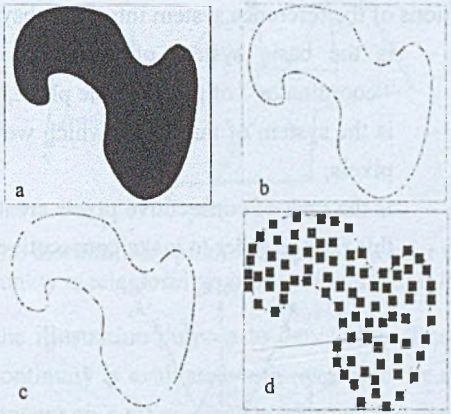


Fig. 5. The cases of the shape specification in the plane
Rys. 5. Przypadki przedstawiania kształtów na płaszczyźnie

In the Fig. 4. all the pixels that belong to Q belong also to the border Q', while the pixel denoted by i at the compact set of pixels does not belong to the border. The given formality enables us to unify shape analysis for any possible data structure that represents it. We have demonstrated the cases of a given shape in the plane in Fig. 5. Processing does not depend on type of the scene used. We can be concerned with 2D scene or 2D image only. The model used can be the solid model or the edge model, see Fig. 5 a). b). c). Moreover, we do not need to care whether we have a real edge or the border is specified. The similar situation we have even if the border is not well defined and we have separated pixels places inside a given range. Therefore, we will use the common term border in order to emphasize that the approach is strongly unified for any type of shapes, see Fig. 5 d). Since the structure of a border is a necklace, so we can well order all the elements of Q'' or O''. Accepting an arbitrary starting element $o_s \in O''$ or $o_s \in Q'$, respectively and moving clockwise along the border we make a ranking R_c.

5. The rank system in the plane

If we have a multiple – coordinate system in the plane, then each axis corresponds to ordering of the elements O and to investigation of the rank R_i, $1 \leq i \leq n$. We distinguish two ranks $R_x()$ and $R_y()$ as the reference ranks. Apart of that we have marked the rank border

R_e. All the basic rank systems are given in Fig. 6. We define formally the rank system in the plane as follows

<<reference>, <multiple-coordinates>, <border >> = <<R_x(),R_y()>, <R_1 - R_n >,<Re>>.

We assign the basic functions of the reference system into the subsystems as follows:

<reference> – is the basic system of reference, we use it as the basic "coordinates" of pixels in the plane,

<multiple-coordinates> – is the system of ranks into which we refer properties of border pixels,

< border > – is the rank of consecutive pixels creating the border, we follow this rank in order to make consecutive evaluations.

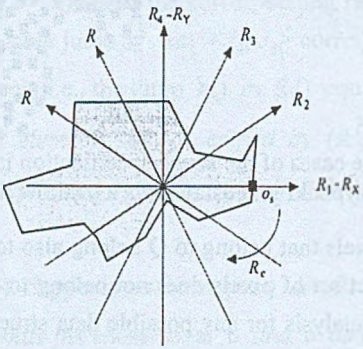


Fig. 6. The rank system used for a shape representation
Rys. 6. System rankingów używanych dla reprezentacji kształtów

6. The D - transform

We are concerned now with shapes represented by a compact set of elements O. Then, the consecutive border elements (pixels) are the neighbors, so the rank R_e is the proper reference for the border specification. Suppose, we follow the border elements in accordance with the rank R_e. Simultaneously, we follow the rank R_i, decreasingly or increasingly i.e., accordingly or in opposite direction into R_i, $1 \leq i \leq n$. Our concern are the pixels o that the rank R_e becomes discontinuous in R_i. For a rank system R_i a given shape can have a number of discontinuities. It is convenient to distinguish the discontinuities, so we denote by **d** the right side discontinuities of R_e in R_i and with **c** we denote the left side discontinuities, accordingly. Similar evaluations we can make for each rank R_i, $1 \leq i \leq k$. Then, we are in position to define the D-transform.

Definition 11. *For a given multiple-coordinates* $<R_i>$, $1 \leq i \leq k$ *the* D *-transform is the collection of all the discontinuities* **d** *and* **c** *of* R_e *evaluated for each rank* R_i.

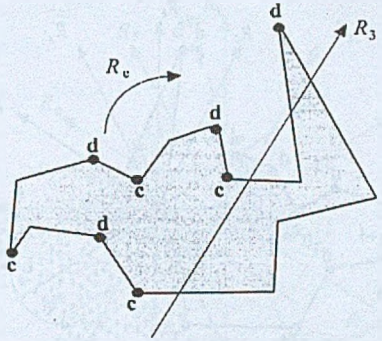


Fig. 7. The discontinuity points of type **d** and **c** for the rank R_3
Rys. 7. Punkty nieciągłości typu **d** i **c** dla rankingu R_3

It is convenient for the illustration purpose to denote the discontinuities **d** and **c** with superscript i if the discontinuity is evaluated with respect to the rank R_i. The D-transform has specific features dependent on a given shape properties. We distinguish three cases of the shapes models that posses interesting D-transform properties, we specify them as follows:

(i) the border is smooth,

(ii) the shape is a polygon, so the border segments are the segments of straight lines,

(iii) the shape is defined by a surrounding Q, then the border is not a collection of neighboring pixels.

At first we consider the case (i), see Fig. 6.

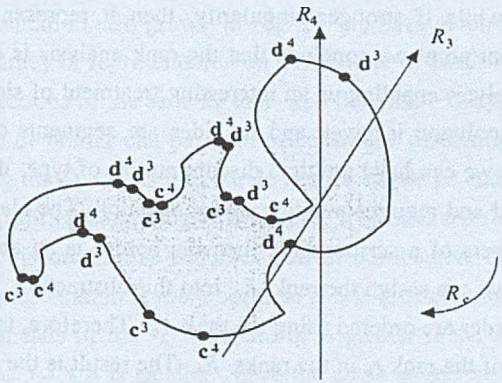


Fig. 8. A part of D-transform evaluated for R_3 and R_4
Rys. 8. Część transformaty D obliczonej dla R_3 i R_4

In order to make easier an analysis we have placed only the discontinuities evaluated for the ranks R_3 and R_4.

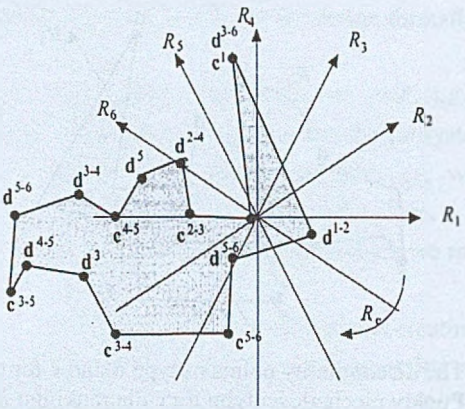


Fig. 9. The D-transform for a polygon in the plane
Rys. 9. Transformata D dla wielokątów na płaszczyźnie

The first observation is that for a smooth segment of the border the discontinuities are single i.e., a given pixel o can be a discontinuity for at most one rank R_i. If the border were represented by a real function $y = f(x)$ of the real argument, then the discontinuity $\mathbf{d}^i$ ($\mathbf{c}^i$) at y would correspond into $\frac{dy}{dx} = m_i$, where m_j is the directional coefficient for the stright line perpendicular into R_i. At certain singular elements of the border we can have multi discontinuities. The second observation is that singularity is a relative notion dependent on the number of ranks (axes) in the plane. So a "weak" singularity can behave like an element of a smooth edge, while if stronger singularity, then it represents a greater number of discontinuities. At that point we conclude that the rank analysis is certain generalization of classical function analysis enabling us an interesting treatment of singularities apart of other novel features. If a polygon is given and its sides are segments of straight lines, then at certain border points we can have multiple discontinuities of type **d** or **c**. The case (iii) is the most complicated and requires additional formalization. The elements that belong to the border are the members of a surrounding, then the border is an abstraction connecting the selected pixels. So, we can assign the rank R_e into this abstraction, while the physical pixels that belong to the border are ordered using the rank r_e. Therefore, the D- transform contains only discontinuities of the rank r_e in the ranks R_i. The result is the same as for the case (ii) i.e., placement of the discontinuities is the same, nevertheless we need to use the jump concept in order to produce the next border pixels as it was specified earlier. In Fig. 10. we have shown the case of the D-transform evaluated for a shape given as a surrounding Q of pixels.

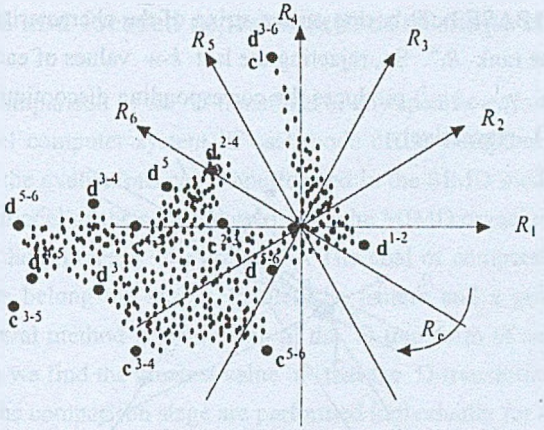


Fig. 10. The D - transform for a given surrounding of pixels
Rys. 10. Transformata D, w przypadku gdy kształt jest otoczeniem pikseli

7. Classes of shapes

The D - transform is a representation of a given shape. In fact, it represents a narrow class of shapes. The accuracy of a given shape representation by means of the D-transform depends on the number of rank axes given in the scene. If the number of axes n grows, then accuracy of the representation also grows. In Fig. 11. we have given the diagram explaining dependence of the accuracy of representation on the number of axes. The discontinuity pixels d^i and d^{i-1} and the directions of the axes R_i and R_{i-1} determine the parallelogram $d^i A d^{i-1} B$ through which a curve or border can pass not effecting the discontinuities creating the D - transform. Hence fixing the D – transform we do not fix the shape enabling the edge (border) to be passing by the specified parallelograms instead. If greater number of the axes R_i, then greater number of the discontinuities and each pair of consecutive discontinuities is placed more tightly, so the representation is more accurate if n is greater. Moreover, the accuracy of the representation by means of the D - transform is greater if the angle ψ between two consecutive axes of the n-CCS is smaller. Since the angle ψ decreases if n grows, so again we have accuracy of representation is proportional into n. In order to increase the size of classes of shapes represented by a common model, we investigate projections of the D - transform onto given ranks.

Projections onto ranks R_i^w

Let $<g_1^i, g_2^i, ..., g_k^i>$ be the ordered string of the characteristic functions that is used for creating the rank R_i. Suppose, we have assigned the characteristic functions in such a way that for any $w<k$ a fixed string of values $<\gamma_1^i, \gamma_2^i, ..., \gamma_w^i>$ creates a compact set of pixels.

Then, the CF DATABASE built basing on the string of the characteristic functions $<g_1^i, g_2^i, ..., g_w^i>$ induces the rank R_i^w. So, rejecting the last k-w values of each discontinuity $\mathbf{d}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ and $\mathbf{c}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ produces the corresponding discontinuities $\mathbf{d}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_w^i)$ and $\mathbf{c}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_w^i)$, respectively.

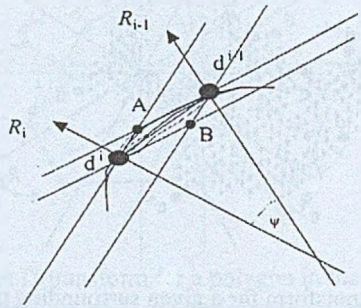


Fig. 11. Accuracy of a shape representation by means of D-transform
Rys. 11. Dokładność reprezentacji kształtu poprzez transformatę D

Formally, the discontinuities $\mathbf{d}^i$ and $\mathbf{c}^i$ are projected onto the rank R_i^w. Rejecting last k-w values $\gamma_{w+1}^i, \gamma_{w+2}^i, ..., \gamma_k^i$ of the characteristic functions $g_{w+1}^i, g_{w+2}^i, ..., g_k^i$ for every $i \leq n$ effects projection of the given D-transform onto the system of ranks $<R_1^w, R_2^w, ..., R_n^w>$, with D^w–transform we denote this projection. Each D^w -transform creates the corresponding class of shapes or curves in the scene. We observe that the class of shapes corresponding to a given D^w -transform is included in the class of shapes corresponding into D^{w-1}-transform.

Projection onto ranks $R_x^w()$, $R_y^w()$.

Each discontinuity $\mathbf{d}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ and $\mathbf{c}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ can be transformed into the ranks $R_x(), R_y()$ getting $\mathbf{d}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$ and $\mathbf{c}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$, respectively. Making such transformation for every discontinuity pixel of the given D-transform we get the $D^{x,y}$-transform. Since each discontinuity $\mathbf{d}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ and $\mathbf{c}^i(\gamma_1^i, \gamma_2^i, ..., \gamma_k^i)$ are transformed uniquely into $\mathbf{d}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$ and $\mathbf{c}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$, so the classes of shapes corresponding into the D-transform and into the $D^{x,y}$-transform are the same. We reject k-w values $\gamma_{w+1}^x, \gamma_{w+2}^x, ..., \gamma_k^x$ and $\gamma_{w+1}^y, \gamma_{w+2}^y, ..., \gamma_k^y$ of the characteristic functions $g_{w+1}^x, g_{w+2}^x, ..., g_k^x$ and $g_{w+1}^y, g_{w+2}^y, ..., g_k^y$ in the similar manner as we did for the D^w-transform. Then, we get the corresponding projection $D_w^{x,y}$-transform of the $D^{x,y}$-transform. The classes of shapes corresponding into D^w -transform and into $D_w^{x,y}$-transform are not identical, in general, however they are very close.

8. Distributed and focused representations of shape classes

Evaluation and comparison of the D-transform of any specific type can be performed in a distributed or parallel computer system. If each node of the computer system is virtual or physical DSM, then the evaluations can be performed in the SIMD model at high efficiency. If a von Neumann 's model of the nodes is used, then the MIMD organization is more suitable in order to increase the efficiency of evaluations. The goal of computations is to judge into which common class belong the shape F called the pattern and a given shape F' to be recognized. The general method is production of the D-transform of any specific type for F and for F' and then we find the greatest value w that the D-transforms mach. So, the main computations up to the comparison stage are performed individually for F and F'.

Distributed patterns

The scene (set of pixels O) at the controller is represented by the following table,

f		$R_y()$	
⋮	⋮	⋮	$\}O$

while the scene at each i -th processor is represented by the following table.

f	$R_x()$	$R_y()$	$R_i()$	
⋮	⋮	⋮	⋮	$\}O$

The controller sets the starting pixel o_s and all the i- th processor follows the border tracking the rank R_e. However, the next step can be performed using the input data structure it is convenient for illustration purpose to demonstrate the intermediate data structure.

f	$R_x()$	$R_y()$	$R_i()$	$R_e()$	
⋮	⋮	⋮	⋮	⋮	$\}O''$ or Q'

So, each processor is concerned with the border now. Then, the discontinuities $\mathbf{d}^i$ and $\mathbf{c}^i$ are produced. The result of computations is the string $< \mathbf{d}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$, $\mathbf{c}^i(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)>$ obtained at each i-th processor. The strings of the discontinuity pixels are produced for the shape F and for the shape $F' < \mathbf{d}^i_F(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)>$ and $< \mathbf{d}^i_{F'}(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$, $\mathbf{c}^i_{F'}(\gamma_1^x, \gamma_2^x, ..., \gamma_k^x, ..., \gamma_1^y, \gamma_2^y, ..., \gamma_k^y)$, respectively. Observe, the ranks $R_i()$ and $R_e()$are used for producing consecutive discontinuity pixels while the ranks $R_x()$ and $R_y()$ are used as a common reference. We assume that the shapes F and F' are close. Frankly speaking the shapes represent topological identity, while small geometrical differences are possible The goal of detailed classification is to specify the class characterizing how "geometrically close" are the shapes F and F'. This class is characterized by the value of the parameter w determining the proper projections of the corresponding D-transforms in order to

get their matching. The following algorithm DDC is used for performing the distributed classification of shapes.

Algorithm DDC (Distributed Discontinuity Classification)

Input: the tables representing the 2D scene as it is specified above.

Output: the value of w determining the $D^{x,y}{}_w$ –transform representing the smallest common class for the shapes F and F'.

1. **for** $i=1$ **to** n **do in parallel**

1.1. Evaluate $\mathbf{d}^i{}_F(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_k{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_k{}^y)$, $\mathbf{c}^i{}_F(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_k{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_k{}^y)$.

1.2. Evaluate $\mathbf{d}^i{}_{F'}(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_k{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_k{}^y)$, $\mathbf{c}^i{}_{F'}(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_k{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_k{}^y)$.

1.3. $w_i \leftarrow k$

1.4. **while** $\mathbf{d}^i{}_F(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_{w'}{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_{w'}{}^y)$, $\mathbf{c}^i{}_F(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_{w'}{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_{w'}{}^y) \neq$

$\mathbf{d}^i{}_{F'}(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_{w'}{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_{w'}{}^y)$, $\mathbf{c}^i{}_{F'}(\gamma_1{}^x, \gamma_2{}^x, ..., \gamma_{w'}{}^x, ..., \gamma_1{}^y, \gamma_2{}^y, ..., \gamma_{w'}{}^y)$

1.4.1. dec (w_i)

1.5. **return** w_i **end of parallel**

2. $w \leftarrow \min_i \{w_i\}$

The value w evaluated in the step. 2 determines the $D^{x,y}{}_w$-transforms of the shapes F and F' that equal. The step 2. can be extended in order to enable us defining threshold equalities of the $D^{x,y}{}_w$ -transforms for the shapes F and F'. Since the steps 1.1. and 1.2. can be processed in time $O(|O'|)$ and the step 2. requires time $O(n)$ and since $|O'| >> n$, so the complexity of the given method is $O(|O'|)$. Evaluation of the set of discontinuities $\mathbf{d}^i$ and $\mathbf{c}^i$ can be shared by a number of processors, the maximal number of processors effectively used for evaluation of the discontinuities $\mathbf{d}^i$ and $\mathbf{c}^i$ can be not greater than $|O|$. Moreover, using the DSM for each node the step 2. can be performed in time O (1). So, the complexity of the method of classification is $O(k)$. Since k is a constant, so the complexity of the method is O(1).

Focused patterns

The system and input data structures are the same except the controller produces the tabular pattern of the shapes F and F'. The square tables $\mathbf{F}[2^k \times 2^k]$ and $\mathbf{F'}[2^k \times 2^k]$ are the representations of the $D^{x,y}$-transforms of the shapes F and F', respectively. The entry $\mathbf{F}[i, j]$ corresponds to the number of all the discontinuities of the $D^{x,y}$-transform projected onto the i-

th value of $R_x()$and onto the j-th value of $R_y()$. The following algorithm FDC can be used for finding the smallest class represented by the common $D^{x,y}$-transform for the shapes F and F'.

Algorithm FDC (Focused Discontinuity Classification)

Input: the tables $F[2^k \times 2^k]$ and $F'[2^k \times 2^k]$ corresponding to the projection of the $D^{x,y}$-transforms onto the ranks $R_x()$ and $R_y()$ concerning the shapes F and F', respectively.

Output: value of w.

1. $p \leftarrow 0$
2. $i \leftarrow 1$
3. **while** $i \leq 2^{k-1}$ **do**

3.1. **for** j = 1 **to** 2^{k-1} **step** 2^p **do**

3.1.1. $\text{sumF} \leftarrow \sum_{q=0}^{2^p-1} \sum_{t=0}^{2^p-1} F[i+q, j+t]$

3.1.2. $\text{sumF}' \leftarrow \sum_{q=0}^{2^p-1} \sum_{t=0}^{2^p-1} F'[i+q, j+t]$

3.1.3. **if** sumF ≠ sumF' **then**

3.1.3.1. $p \leftarrow p+1$

3.1.3.2. **if** $(i \bmod 2^p \neq 0)$ **then** $i \leftarrow i - 2^{p-1}$ **go to step 3.1.**

3.2. $i \leftarrow i + 2^p$

4. $w \leftarrow k-p$
5. **return** w

The given algorithm FDC takes time $O(2^{2k})$ for completing the classification. Observe, 2^k grows linear on the size of the scene.

REFERENCES

1. Cheikh F., Quddus A., Gabbouj M.: Shape recognition based on welvet-transform modulus maxima. in ICECS 2000 IEEE Part Vol.1. 2000 pp.461-464, Piscataway, NJ, USA.
2. Dougherty E.R., Giardina C.R.: Matrix Structured Image Processing. Prentice-Hall Inc. Englewood Cliffs, New Jersey 1987.
3. Kapralski, A.: Sequential and Parallel Processing in Depth Search Machines. World Scientific, Singapore 1994.

4. Kapralski, A.: Modelling Arbitrary Sets of Combinatorial Objects and their Sequential and Parallel Generation. Studia Informatica Vol. 21, No 2(40) Silesian University of Technology, Gliwice 2000.
5. Kapralski A: Fundamental Models for Processing in Depth Search Machine. (to appear).
6. Leavers V. F.: Which Hough Transform. CVGIP Image Understanding Vol.55, No 2 1993, pp.250-254.

Recenzent: Prof. dr hab. inż. Konrad Wojciechowski

Wpłynęło do Redakcji 8 kwietnia 2002 r.

Streszczenie

W pracy przedstawiono transformacje płaszczyznowych krzywych i kształtów w oparciu o podstawy matematyczne rozwinięte na podstawie CF BAZDANYCH, systemu wielokrotnych współrzędnych płaszczyznowych *n*-CCS oraz analizy rankingowej. Podstawą pojęciem jest pojęcie punktu nieciągłości rankingu związanego z krawędzią figury płaszczyznowej w rankingach związanych z porządkami indukowanymi poprzez osie *n*-CCS. Punkty nieciągłości są organizowane w różnego typu transformaty D, których zagnieżdżająca się struktura tworzy system klasyfikacji zadanych kształtów. Zagnieżdżające się klasy krzywych i kształtów planarnych tworzone są następnie w oparciu o rzutowanie charakterystyk otrzymanych z analizy rankingowej. Ponieważ systemy analizy rankingowej mogą być tworzone przy założeniu dowolnej dokładności reprezentacji oraz ponieważ parametry rzutowania mogą być arbitralne, wobec tego definicja klas kształtów i krzywych może być przeprowadzana z dowolną dokładnością. Wprowadzone są dwa wzorce struktur danych dla skupionego oraz dla rozproszonego reprezentowania i rozpoznawania najmniejszej wspólnej klasy. Ten abstrakcyjny system reprezentacji i klasyfikacji może być implementowany w pojedynczej maszynie lub w sieciach komputerowych typu SIMD lub MIMD. Każdy węzeł takiego systemu jest maszyną do głębokiego wyszukiwania. W przypadku użycia fizycznych maszyn do głębokiego wyszukiwania złożoność algorytmu rozpoznawania wspólnej klasy jest rzędu O(1). Podstawową własnością takich systemów jest nieograniczona skalowalność. Dla ustalonej liczby węzłów możliwe jest uzyskanie dowolnej dokładności reprezentacji lub minimalizacja czasu przetwarzania.