Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 1002–1011
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Polygonal shape description for recognition of partially occluded objects

Filip Krolupper *, Jan Flusser

Institute of Information Theory and Automation Academy of Sciences of the Czech Republic, Pod vodarenskou vezi 4, 182 08 Praha 8, Czech Republic

Received 17 October 2004; received in revised form 1 December 2006
Available online 4 February 2007

Communicated by S. Dickinson

Abstract

We introduce a new method for the recognition of partially occluded objects represented only by their contours. Object description, which stems from the inflection point detection, approximates the object by polygon and is affine invariant. The matching algorithm is simple and easy to implement.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Occluded object recognition; Polygonal approximation; Affine invariant

1. Introduction

1.1. Motivation

The recognition of a partially occluded object is a major problem in computer vision. This problem has not been sufficiently resolved yet, although many people have been working on it for about last twenty years.

Partially occluded object recognition is needed for example in forensic applications, medicine, astronomy, industry applications etc. We present an example to illustrate the use of the partially occluded object recognition algorithms.

Optical character recognition (OCR): Fig. 1 shows three images of a Machine Readable Zone (MRZ). The MRZ is a special part of any valid travel document made according to the ICAO Document 9303. It stores all important information about the holder in a special code. The MRZ is developed in such a way that a machine can read it automatically and check the holder's identity thereby improving security.

Characters in the MRZ are printed by an infra red (IR) absorbing material but sometimes it becomes necessary to capture the MRZ by a simple visible light sensitive camera. This raises a problem with the hologram labels which are printed on every travel document to avoid forgery. These labels make some characters discontinuous which results in occlusion (see Fig. 1c). Another case of occlusion can appear when the camera cannot capture the whole MRZ into one image and some letters are cropped (see Fig. 1b). In these cases we need an algorithm for partially occluded object recognition.

Images in Fig. 1 were captured by a Voskuhler 1.3MPx camera, where the MRZ was illuminated by the visible light. Images were obtained from an image acquisition product called DocuCenter, which is now widely used at national airports, country borders and police stations.

Many more examples can be found in the 3D computer vision. Partially occluded object recognition is needed in the fields of forensic applications, medicine and astronomy – objects very often overlap in these areas. The automatized recognition of the occluded objects is usually the first stage of the process. After the recognition we can observe how the object changes in time.

* Corresponding author. Tel.: +420 728 526 720; fax: +420 286 890 378.
E-mail addresses: filip@utia.cas.cz (F. Krolupper), flusser@utia.cas.cz (J. Flusser).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.12.021

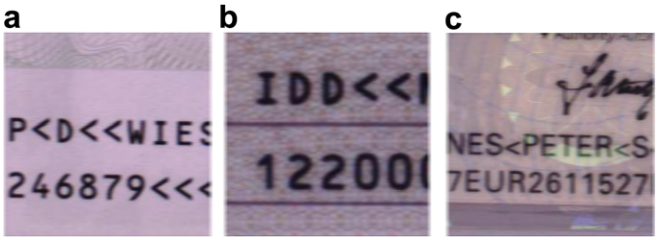

Fig. 1. Images from the DocuCenter: (a) A part of the MRZ correctly captured. (b) The MRZ with cropped last characters. (c) Several letters are made defective by the hologram reflection.

1.2. Our research

The task of the partially occluded object recognition algorithm is to match a single object against a database of objects. An algorithm should be able to confirm, whether the occluded and transformed single object is one of the objects in the database. In this paper, objects are represented only by their binary contours and an affine transformation is considered because it is a good approximation of perspective transformation in most 3D computer vision tasks.

We focus on a reliable description (features extraction) of the object. To be robust to occlusion and to be invariant to the basic transformation, the description should have a local character. Moreover the description should be so accurate that it should match just the object it represents, and none other.

There are three basic approaches, that handle the problem of object description. The first approach is a string feature characterization of the objects (the string contains several features for every point). The well known local differential invariants from Weiss (1992) belong to this category. Weiss suggested geometric invariants, which contain high order derivatives and are sensitive to noise. Similarly Pikaz and Dinstein (1995) used a total curvature description with a fast sub-curve matching, which also belongs to this category.

The second approach is the object description by important points such as minimal/maximal curvature points, or curvature inflection points. Tsang et al. (1994) suggested a polygonal approximation approach and used weighted distance transform to get a similarity score. The Han and Jang (1990) method stems from the Duda and Hart (1973) polygonal approximation which was extended to be faster and used a compatibility indexing for the matching. Lamdan et al. (1988) used a hash table for a fast triplets correspondence. Werman and Weinshall (1995) introduced a new measure between two sets of points and used it for object recognition.

The third approach is the boundary approximation by splines used by Cohen et al. (1992, 1995). They use B-splines for the approximation, because if a control points correspondence is well defined, an affine transformed B-spline becomes the B-spline again.

There are a few other papers, like the one by Turney et al. (1985), suggesting algorithms that do not belong to any of the above categories. Their method uses a saliency measure(matrix of numbers) as a descriptor. A shock graph from Siddiqi et al. (1998), codons from Richards and Hoffman (1985) and well known projective invariants from Rothwell et al. (1991) are the other possibilities for object description. Unfortunately the shock graph is not affine invariant, codons give us a weak object description and the projective invariants from Rothwell are not suitable for complex objects. From the other approaches we should mention the genetic algorithm of Kawaguchi and Nagao (1998) and the wavelet based approach by Tieng and Boles (1997). Mokhtarian and Abbasi (2002) use evolution CSS image matching.

This paper contains a new method for object shape description. The main advantage of the method is that it creates the affine invariant description for objects without computing parameters of the affine transformation. It is based on detection of inflection points and can make the description by inflection points as detailed as required. This makes the description quite robust to both – additive noise and inaccuracy in inflection point detection.

Our method can be categorized as the polygonal shape approximation based on the inflection point detection, that is it can be included in the second category described above.

Let us assume, that our objects have a continuous and piecewise smooth border. For a parametric curve of the object border $y = y(t)$, $x = x(t)$ the inflection point is defined:

$$\ddot{x}\dot{y} - \dot{x}\ddot{y} = 0$$

Many papers have been published about the detection and matching of dominant (especially inflection) points. See for instance Ansari and Delp (1990), Teh and Chin (1989) or Tsang et al. (1994). Much work has also been done on the polygonal approximations. Most methods are based on the Duda and Hartmethod (1973) but almost none of them take into account the invariance to the affine transformation.

Methods for dominant point detection are presented for example in works of Mannocha and Canny (1992) and Mokhtarian and Mackworth (1986).

The affine invariant coordinate system, which is the keystone of our object description , is mentioned in Section 2. Section 3 shows how to obtain the representation by polygonal approximation. The similarity score for object matching and object matching itself is presented in Section 4. Finally, experimental results, comparison with the Lamdan method, real world experiments, and algorithm limitations are mentioned in Section 5.

2. Affine invariant coordinates

Affine transformation can be defined by the following equations:

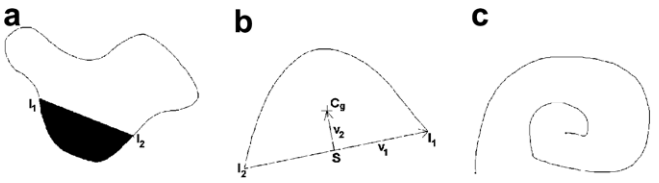

Fig. 2. (a) A new object defined between two inflection points. (b) Affine coordinates system. (c) If such a curve is a part of our contour, this method of approximation is unusable.

$$x^t = ax + by + c \quad y^t = dx + ey + f \tag{1}$$

where x^t, y^t are the transformed coordinates, x, y the original coordinates and a, b, c, d, e, f the transformation coefficients.

Let us take a digitized contour of the object. The contour is naturally split into several parts by inflection points $I_1, \ldots, I_n$. We will deal with the part of the contour that begins and ends in inflection points. Let us assume that the object given by (I_1, part of border, I_2, line (I_2, I_1)) is convex (see Fig. 2b for an example and Fig. 2c for a counter example) and call such an object a *shred*.

The next step is to compute the center of gravity of the shred. The center of gravity (C_g) is invariant to affine transformation. The new affine coordinate system is defined by the origin S and two vectors v_1, v_2 (see Fig. 2b). S is the point between I_1, I_2, such that $S = (I_1 + I_2)/2$. Vectors v_1, v_2 are defined as $v_1 = I_1 - S$ and $v_2 = C_g - S$.

Relative coordinates are preserved, that is the coordinates of each point, which belongs to the border in the new coordinates system is affine invariant.

3. Polygonal approximation

The shred description is based on the hierarchical shred approximation in the affine coordinate system. The first phase of approximation is the parallelogram defined by vectors v_1, v_2. The first line of the parallelogram is parallel to the vector v_1 and goes via point S. The second line is also parallel to the vector v_1, but touches the shred on the other side. The third and the fourth lines are both parallel to the vector v_2 and also touch the shred at separate points. The points of intersection of these lines represent the corners of the parallelogram (see Fig. 3a).

The second phase involves the four corners of the parallelogram and the approximation is improved in the following way. For every corner B_i take the vector v_i defined as $B_{i+1} - B_{i-1}$. Take the line in the direction v_i and such that it touches the shred (it need not be tangential because points I_1, I_2 do not define tangents). This line intersects the parallelogram(generally a polygon) and defines a new approximation(see Fig. 3b). The procedure is described in detail in the next paragraph.

The algorithm description:

1. Fix the number of iterations *NoI*
2. Let $P := \{B_1, \ldots, B_m\}$ be the set of corners from the previous phase approximation.
3. *For* every point B_i *do*
   (a) Compute direction $v_{ni} = B_{i+1} - B_{i-1}$ of a new tangent.
   (b) Find the translation of the tangent, that means find the exact equation of the tangent with v_{ni} direction.
   (c) Cut off the polygon and get a new list of the corners P'.
   (d) *endfor*
4. If the loop (steps 2–4) has been proceeded less than *NoI* times, set $P = P'$ and go to step 2.

The new boundary representation is compiled from the corners in the list P, defined by the algorithm in the affine coordinate system (see Fig. 5).

3.1. Discussion

In this section we discuss the advantages and the disadvantages of our boundary representation. The first advantage is that the boundary can be described as accurately as required. The approximation error decreases very fast. The second advantage is, that the intersection of the lines of the polygonal approximation is robust to noise although the point of touch of the line and the original contour is unstable.

Our object description has several disadvantages. The main disadvantage is sensitivity to accuracy of inflection point detection. However this is a general property of the methods which use inflection points and fortunately the sensitivity of our algorithm is not critical.

An experiment to illustrate In algorithm behavior under these conditions is described in Section 5.5.

The second disadvantage is the error propagation. That means, if some inaccuracy is made in the intersection finding process, the error distributes to the next approximation phase. The last disadvantage is that it is not possible to detect whether the affine coordinates should be defined by (v_1, v_2) or by $(-v_1, v_2)$ (see Fig. 2b) because affine transformation allows mirroring.

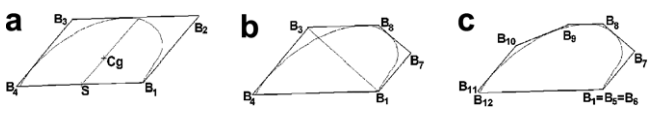

Fig. 3. (a) The basic parallelogram. (b) One step of the cutting algorithm. (c) The approximation after the second phase.

4. The similarity score

Let us take two shreds with its affine invariant polygonal approximations ap_1 and ap_2, as described above. In this section, we will show two simple penalty functions for the shreds matching. At the end of this section we give consideration to the matching algorithms for the complex objects.

4.1. The first penalty function

Let us define characteristic function $F(i,j)$ for polygon ap:

$$F(i,j) = 1 \Longleftrightarrow (i,j) \in ap$$
$$F(i,j) = 0 \Longleftrightarrow (i,j) \notin ap$$

Let the functions F^1, F^2 be characteristic functions of ap_1, ap_2. Define the penalty function $F(ap_1, ap_2)$ to be a symmetric difference $F^1 \div F^2$.

The symmetric difference $F(ap_1, ap_2)$ for polygons ap_1 and ap_2 can be also defined following way:

$$F(ap_1, ap_2) = \sum_{i \in I} \sum_{j \in I} R(i,j)$$

where

$$R(i,j) = 1 \Longleftrightarrow (i,j) \in ap_1 \quad \text{and} \quad (i,j) \notin ap_2$$
$$R(i,j) = 1 \Longleftrightarrow (i,j) \in ap_2 \quad \text{and} \quad (i,j) \notin ap_1$$
$$R(i,j) = 0 \Longleftrightarrow (i,j) \in ap_1 \quad \text{and} \quad (i,j) \in ap_2$$
$$R(i,j) = 0 \Longleftrightarrow (i,j) \notin ap_2 \quad \text{and} \quad (i,j) \notin ap_1$$

4.2. The second penalty function

Compute the distance from every point of ap_1 to the second polygon and from every point of ap_2 to the first polygon. The sum of the distances is quite suitable penalty function. Its complexity is $O(n^2)$ where n is the size of the bigger set of ap_1, ap_2.

The second penalty function gave us worse discriminability than the previous one, but it is faster. We have chosen the first penalty function in our experiments.

In fact, the penalty function has to be $\min(F(ap_1, ap_2), F(ap_1, \text{mirror}(ap_2))$, where $\text{mirror}(ap_2)$ is the mirror reflection image of ap_2 with the centerline (S, CG) (see Fig. 2b). The penalty function has to be defined this way because of considering affine transformation.

Two polygonal approximations ap_1, ap_2 are labeled as *corresponding* if their penalty function is lower than given threshold.

4.3. On matching the complex objects

We have proposed an algorithm to represent an object by pieces of its borders – so called shreds – and a similarity score to mutually match these pieces. Now we will introduce a method to match the complex borders of the whole objects based on proposed representation.

Let us take template object from our database O_1. Its set of inflection points $I_1 = \{i_1^1, \ldots, i_1^n\}$ divide the border into n independent parts $P_1 = \{p_1^1, \ldots, p_1^n\}$. Parts p_1^i have corresponding lengths L_1^i in pixels and applies $L_1 = \sum_i L_1^i$ where L_1 is the length of the complex border. For every part p_1^i let us denote the polygonal, affine invariant representation of this part ap_1^i. Finally, object O_1 has its polygonal affine invariant representation $AP_1 = \{ap_1^1, \ldots, ap_1^n\}$.

Let us take the unknown object O_2, set of its inflection points $I_2 = \{i_2^1, \ldots, i_2^m\}$, independent parts $P_2 = \{p_2^1, \ldots, p_2^m\}$ and affine invariant representation $AP_2 = \{ap_2^1, \ldots, ap_2^m\}$.

We try to match every part from AP_1 to every part of AP_2 with the key condition, that the order of the shreds must be respected. That means the parts $ap_1^1, ap_1^2, ap_1^3, ap_1^4, ap_1^5$ can for example correspond to $ap_2^2, ap_2^3, ap_2^4, ap_2^5, ap_2^6$ (only if ap_1^1 *corresponds* to ap_2^2, ap_1^2 *corresponds* to ap_2^3 and so on). Cyclic correspondence and a mirrored correspondence can occur. $ap_1^1, ap_1^2, ap_1^3, ap_1^4, ap_1^5$ can correspond to $ap_2^5, ap_2^6, ap_2^7, ap_2^1, ap_2^2$ (cyclic correspondence) or $ap_2^8, ap_2^7, ap_2^6, ap_2^5, ap_2^4$ (mirror correspondence) or cyclic and mirror correspondences can occur together.

For every set of corresponding pairs

$$M = \{ap_1^{i1} - ap_2^{j2}, ap_1^{i2} - ap_2^{j2}, \ldots, ap_1^{ik} - ap_2^{jk}\}$$

We compute the similarity score for the set M

$$\text{SSC}_M(O_1, O_2) = \frac{1}{L_t} \sum_{m \in i1 \ldots ik} (L_1^m) * 100$$

The similarity score between two objects O_1, O_2 is defined as the maximum of all similarity scores of the allowed correspondences.

$$\text{SSC}(O_1, O_2) = \max_M \{\text{SSC}(O_1, O_2)_M\}$$

4.4. Occlusion

We still have not discussed the problem of the occlusion. We have described the polygonal approximation algorithm and the matching algorithm for object recognition. But if the object is not occluded, many other object descriptors that are more suitable can be used (e.g. moments). Occlusion however makes them totally unusable.

In order to make possible recognition of the occluded objects, we changed the matching algorithm from the previous sections slightly. Let us take two borders of objects O_1, O_2 and their affine invariant representations $AP_1 = ap_1^1, \ldots, ap_1^n, AP_2 = ap_2^1, \ldots, ap_2^m$. Again, every part of AP_1 is matched to every corresponding part of AP_2, respecting the shreds order and cyclic and mirroring correspondence. But unlike the previous case, every correspondence must take into account the fact that some parts may be occluded. For example correspondence $ap_1^2, ap_1^4, ap_1^5, ap_1^8, ap_1^9$ to $ap_2^1, ap_2^3, ap_2^4, ap_2^7, ap_2^8$ is permitted (we assume that shreds No. $1,3,6,7$ from the first object and shreds No. $2,5,6$ from the second object are occluded in this correspondence). Missing shreds are permitted in cyclic and mirrored correspondence as well. This extension of correspondence makes the matching algorithm more time consuming but applicable for the partially occluded object recognition.

For example if the inflection points $I_1, \ldots, I_5$ were detected on the object border and points I_2, I_3 were occluded then the $\text{Shred}(I_4, I_5)$ and $\text{Shred}(I_5, I_1)$ will be recognized correctly by our algorithm. This is because

missing inflection points have no influence on the distant shreds.

5. Results

We have tested this algorithm on different types of curves which were transformed by affine transformation. In this section we show the separability of types of different objects. Tests were first done on parts of objects which were bordered by two inflection points.

Fig. 4 shows the images of four different objects and their affine transformation. Tables 1 and 2 contain the values of the penalty function. The shortcut "o_1" denotes the first object. The shortcut "o_1t_1" denotes the first object affine transformed by the first set of the coefficients.

Tables 1 and 2 show us an excellent discriminability of the penalty function. In the worst case the penalty function is approx. four times greater between two different shreds than between original and affine transformed shreds.

5.1. The recognition of the complex objects

The recognition process consists of three steps, as mentioned before. The first step searches for the inflection points and splits the border into segments according to the inflection points (see Fig. 6). The second step is putting every part of the border into affine invariant form as described in the previous section. The third step is matching these forms to each other.

For *the first step*, we used in our tests the automatic inflection point detection which does not need any derivatives and which is robust to noise.

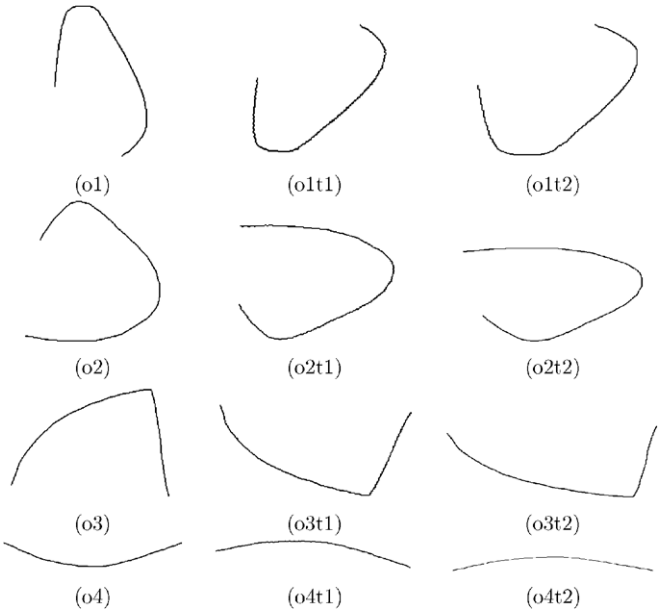

Fig. 4. (o_1–o_1t_2): The first object and its affine transformation, (o_2–o_2t_2), (o_3–o_3t_2), (o_4–o_4t_2): The second, third, fourth objects and their affine transformation, respectively.

Table 1
Penalty function of the original object and the object transformed by affine transformation

	o_1	o_1t_1	o_1t_2		o_2	o_2t_1	o_2t_2
o_1	0	784	887	o_2	0	760	874
o_1t_1	784	0	1187	o_2t_1	760	0	556
o_1t_2	887	1187	0	o_2t_2	874	556	0

	o_3	o_3t_1	o_3t_2		o_4	o_4t_1	o_4t_2
o_3	0	824	714	o_4	0	1672	1010
o_3t_1	824	0	430	o_4t_1	1672	0	1722
o_3t_2	714	430	0	o_4t_2	1010	1722	0

The penalty function is not zero due to discretisation problem by affine transformation when strong skewing presented.

Table 2
The penalty function of all original objects

	o_1	o_2	o_3	o_4
o_1	0	7824	16498	21441
o_2	7824	0	19693	25002
o_3	16498	19693	0	6117
o_4	21441	25002	6117	0

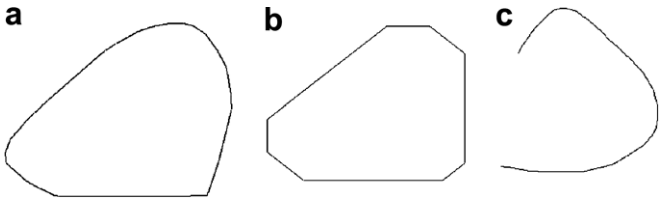

Fig. 5. (a and b) The second and third phase of the contour approximation in the affine coordinates. (c) The original image in original coordinates.

For every point of the border we can decide if the point is inflection point or not in the following way: Take the current point and insert a circle(with center in the current point) into the image. The border splits the circle into two parts – inner and outer part. The current point is the inflection point, if the areas of the inner and outer parts are the same.

The second step – putting every part of the border into the affine invariant form – was undertaken exactly as described in Section 3.

The third step – matching the complex objects was described in the previous section.

Please see Fig. 7 for the tested objects and Table 3 for the results. Numbers in Table 3 represent the percentage of the part of the original border that was matched to the border of the tested object.

If we wanted to recognize objects by maximizing the percentage of the matched border, the recognition algorithm would recognize all objects correctly. All borders had approximately 600 pixels, and as we can see, it was enough for the inflection point detection, but not enough for the correct recognition of all parts of objects. We can see at Table 3 that co_1 corresponds to cot_1 in 78.14% of

Fig. 6. Examples of two objects, on which inflection points were detected.

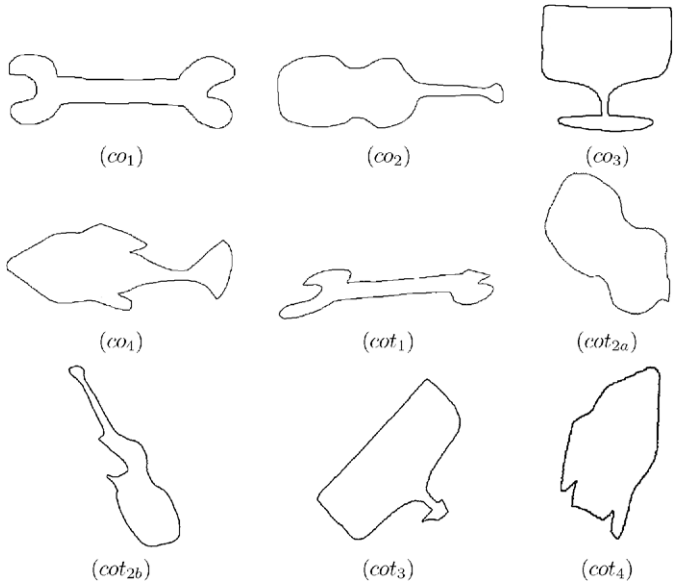

Fig. 7. (co_i) are the original complex objects, (cot_j) are the transformed and partially occluded complex objects.

Table 3
The correspondence (as a percentage) between the original and transformed and occluded objects

	co_1	co_2	co_3	co_4
cot_1	78.14	23.43	10.99	17.67
cot_2a	0	65.36	0	0
cot_2b	11.22	59.16	22.32	17.67
cot_3	23.7	32.47	64.6	24.03
cot_4	6.64	0	0	53.18

the border, although the damage to the bone border was not too severe. We would presume correspondence of about 85%.

5.2. Extensive testing on simulated data

To get the statistically reasonable results, we have suggested the following experiment. We took four test objects – templates, we changed their contour randomly and transformed them by affine transformation. The contour was impaired by 0–50% and affine transformation had coefficients $a = 1$, $b = 0$, $d \in (0, \ldots, 0.5)$, $e = 1$. This simplification of affine transformation is suitable, because scaling and rotation have only a little influence on our recognition algorithm and could make the results ambiguous.

For every contour, for every $d = 0$, 0.1, 0.2, 0.3, 0.4, 0.5 and for every contour impairment of 0%, 10%, 20%, 30%, 40%, 50%, four realizations were made which differed in the location of the contour impairment (these locations were chosen randomly). The algorithm was tested on 576 images. You can see the original images and samples of the transformed and damaged images in Fig. 8. For the results see Table 4.

This method seems to be suitable up to skew $d = 0.5$ and impairment of 20%. If the skewing is strong, the discretisation effect causes the worse recognition. There is no theoretic limitation, of course. This method and generally all methods based on inflection point detection suffer from inaccurate inflection point detection under affine transformation and low resolution.

5.3. Comparison to other methods

We have decided to compare our method to the Lamdens method (1988), because these two are the most

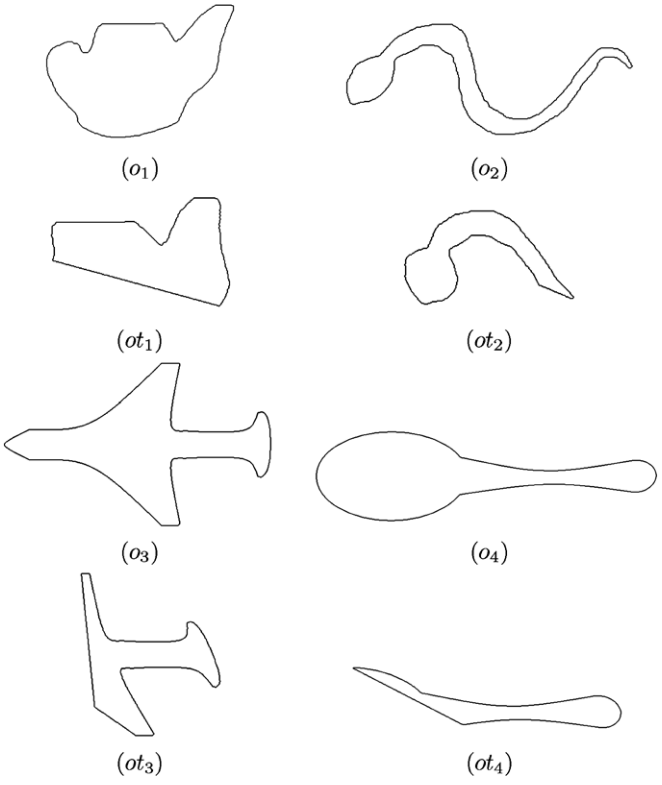

Fig. 8. (o_1) corresponds to (ot_1) where the teapot was transformed with $d = 0.4$ and 50% of the contour was occluded, (o_2) corresponds to (ot_2) where the snake was transformed with $d = 0.5$ and 50% of the contour was occluded, (o_3) corresponds to (ot_3) where the plane was transformed with $d = 0.4$ and 40% of the contour was occluded, (o_4) corresponds to (ot_4) where the teapot was transformed with $d = 0.2$ and 40% of the contour was occluded.

Table 4
The recognition results

Skew/damage	0%	10%	20%	30%	40%	50%
$d = 0.0$	100	93.8	93.8	81.3	81.3	81.3
$d = 0.1$	100	87.5	93.8	75.0	81.3	81.3
$d = 0.2$	100	87.5	93.8	75.0	81.3	68.8
$d = 0.3$	100	87.5	87.5	81.3	81.3	68.8
$d = 0.4$	100	87.5	87.5	68.8	81.3	68.8
$d = 0.5$	93.8	87.5	93.8	75.0	87.5	68.8

For the object skewed with $d = 0.1$ and with 20% of the contour occluded, the probability of correct recognition is 93.8%(from our four template objects).
In the remaining cases the recognition process delivers false results – the percentage of matched border is higher for another template – or the similarity score for all object is zero.

similar. First let us point out the similarities and differences between these two methods.

Both methods deal with affine transformation and need to find out important points (such inflection points) for the non-convex object description.

The first and the main difference between these two methods is that our method computes approximations of all shreds and matches these approximations for two different objects by matching algorithm which is similar to substring matching. The Lamdans method computes parameters of affine transformation from two corresponding shreds and verifies if the other important points match.

The second difference (definitely less important) is the meaning of the phrase "shred correspondence". In the Lamdans paper the decision that two shreds correspond is based on radial vectors similarity of the shreds. In our paper it is a penalty function between shred approximations, what is the key to decision if two shreds correspond or not. The shred approximation contains more information about the shred and therefore leads to less false matches.

We implemented Lamdans method and tested it on the same images that were used for *extensive testing* (see 5.2 for details) of our method and with the same inflection points. Results are available in Table 5. The Lamdans method performs worse than our method, namely when skewing becomes large. In such cases the detection of inflection points becomes less accurate and this affects the Lamdans method more than ours.

Table 5
The recognition success of the Lamdans method

Skew/damage	0%	10%	20%	30%	40%	50%
$d = 0.0$	100	83.3	75.0	70.0	63.6	33.3
$d = 0.1$	66.6	50.0	58.3	30.0	45.4	00.0
$d = 0.2$	66.6	66.6	58.3	20.0	45.5	22.2
$d = 0.3$	33.3	33.3	33.3	10.0	27.2	11.1
$d = 0.4$	66.6	41.6	33.3	20.0	45.4	22.2
$d = 0.5$	66.6	50.0	41.6	30.0	54.5	33.3

5.4. Robustness to additive gaussian noise

We carried out tests with objects corrupted by additive noise. We must point out that more significant corruption near inflection points can make the description very different from the original one.

Our tests were based on adding gaussian noise to the original borders. Every point of the border was translated in the direction normal to the border. The length of translation vector was determined by a random number of normal distributions $N(0, \sigma^2)$ – see Fig. 9.

We added gaussian noise with $\sigma^2 = 1, 4, 16, 64$ to each of our four original images o_1, o_2, o_3, o_4 and made 100 realizations of every object and sigma. The results of matching of the original and noise-corrupted images are presented in Table 6.

Table 6 shows that the penalty function between any tested shreds and the corresponding corrupted ones (in the worst case corrupted by gaussian noise with $\sigma^2 = 64$) is lower than the penalty function between any tested different shreds (see Table 2). Considering the relatively low resolution of shreds, we can assert that our algorithm is robust to additive gaussian noise.

5.5. Robustness to inaccurate inflection point detection

In this section we want to discuss robustness of the polygonal affine invariant description to the inaccurate detection of inflection points.

We took four objects (see Fig. 10) and we computed the penalty function between the original object and the object which had its beginning and end (the new inflection points)

Fig. 9. Samples of the images with additive noise.

Table 6
The mean of the penalty function between the original images and 100 realizations of additive noise to images with given σ^2

	o_1 (174 × 237)	o_2 (233 × 295)	o_3 (276 × 244)	o_4 (358 × 106)
$\sigma^2 = 1$	1198.6	943.4	699.5	2506.8
$\sigma^2 = 4$	1811.9	1363.8	894.9	3494.3
$\sigma^2 = 16$	2696.9	1629.0	1387.6	4213.7
$\sigma^2 = 64$	3717.7	2073.4	1920.5	5678.6

The numbers in the brackets represent the real width and height of the original images.
Please compare the penalty function from the graph to the penalty function between different shreds (Table 2).
You can see, that the penalty function between all tested different shreds is greater than between shreds and noisy shreds although σ^2 was up to 64. This shows very good stability to additive noise.

in the highlighted areas. The border points of the highlighted areas were designed to be the original inflection point plus or minus five percent of the length of the border between the inflection points.

For the results of the penalty function see Fig. 11. The x-axis represents (in %) how much the first and the second inflection points were shifted together. The y-axis represents the penalty function. If the first inflection point was shifted by 3% (plus or minus) and the second point by 2% (plus or minus) the penalty function of the original object and the object with shifted inflection points can be found in the graph – where x is equal to 5.

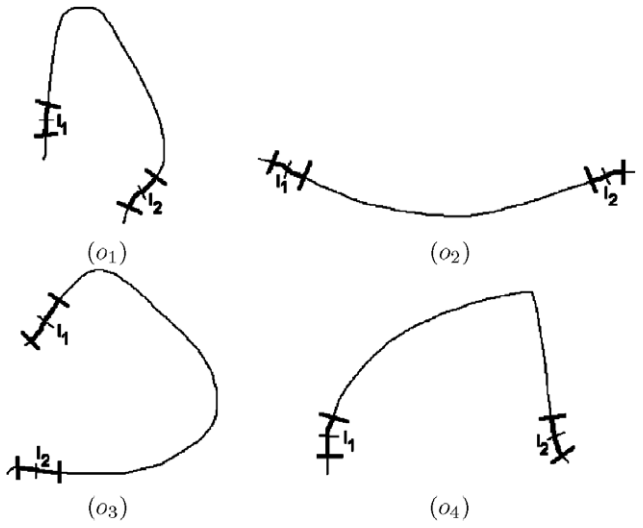

Fig. 10. I_1, I_2 labels in every image the original points of inflection. We computed the penalty function between the image with original inflection points and the images, where the inflection points were chosen from the highlighted interval. The size of the interval is 10% of the length of original border.

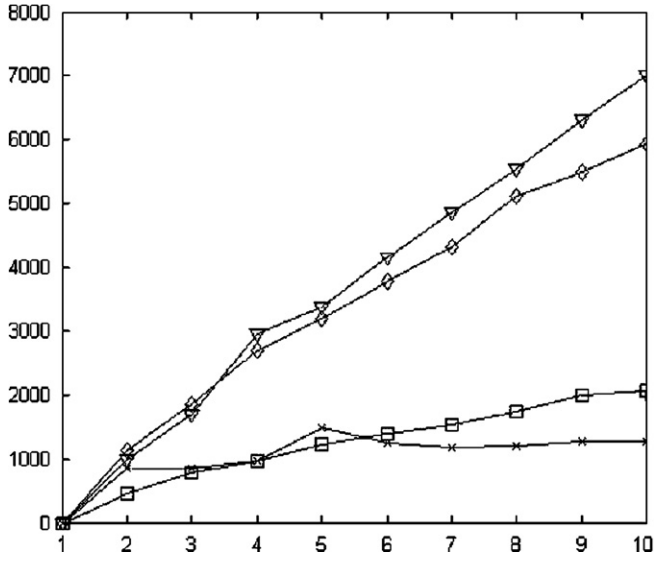

Fig. 11. This graph corresponds to Fig. 10 and shows the penalty function between the original object and the object which had its inflection points shifted. The x axis represents (in a %) how much the first and the second inflection points were shifted together. The y axis is the penalty function. Diamonds represent o_1, x-marks o_2, triangles o_3 and squares o_4.

The algorithm seems to be unaffected by small inaccuracies in inflection point detection. From Fig. 10 we could guess that objects o_1 and o_3 would give us the worst results, because of the bump near I_2 (object o_1) and near I_1 (object o_3) which is the reason for the significant approximation modification. This presumption was confirmed (see Fig. 11).

5.6. Tests on the real objects

We tested our algorithm on the real objects such as knife, screwdriver, scissors, spoon, axe, pliers, stopper, etc. There were eleven different objects in the template database. Every object consists of 4–8 shreds. Four shreds have spoon, screwdriver or stopper and eight shreds has pliers.

We took fifteen pictures of every object and although we had almost uniform red background, segmentation was correct in 10–13 cases for every object. The corrected images were our input files in this test.

We used camera with 3MPixels with a macro mode. The objects had length and width from 10 to 50 cm and depth from 3 mm to 3 cm. The macro mode uses focal length from 10 to 40 cm. The limit for camera inclination was 45°. These conditions had three different implications.

- Complex object would not appear whole in the focus plane – its border was not detected accurately.
- The object image is deformed by strong perspective transformation. See Fig. 13a1 and b1.
- When the object has non-zero depth its contour is not, from different camera inclination angles, created by the same edges. See Fig. 13a2 and b2.

Despite these bad conditions, we think that affine transformation still would be a good approximation of perspective transformation, which deforms our images.

The problem of additional edges, which create different contours for different camera inclinations, when the object has non-zero depth, is not so critical. We have to realize that in many cases the additional edges have the similar shape to the original ones and the inflection points are preserved too. However, this does not hold for the piers.

As we mentioned above, we made 10–13 pictures of every object. The occlusion was from 0% up to 50% of the object border. For non-occluded, only perspective transformed objects, the recognition ratio was 88.9%. When the occlusion was 50% of the length of the border, the recognition ratio was 43.1%. As we suspected the worst recognition ratio gave object with small number of shreds and hardly detectable inflection points (cooking spoon – Fig. 12d2) and objects with the large depth (pliers – Fig. 12a1). On the other hand the flat tool with higher number of inflection points, which were well

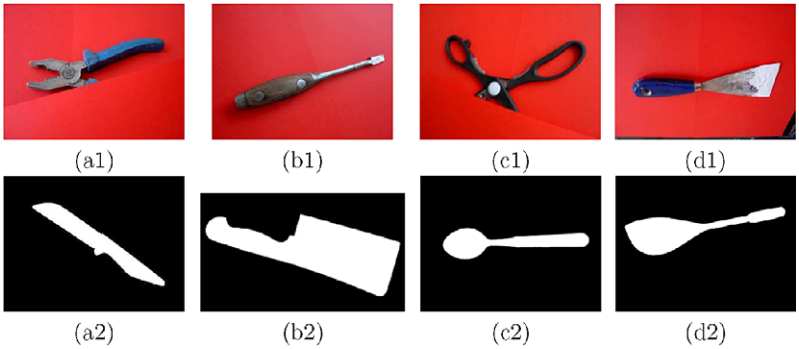

Fig. 12. (a1)–(d1) Several examples of tested objects. (a2)–(d2) Examples of several segmented objects.

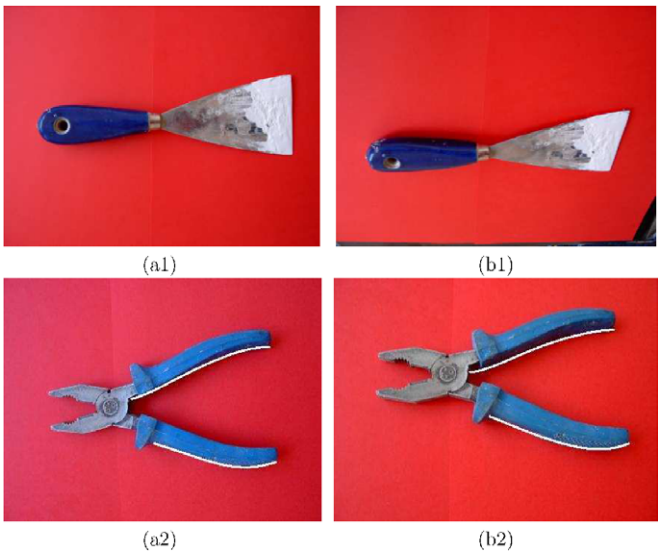

Fig. 13. (a1) The original image of the stopper. (b1) The image of the stopper under strong projective transformation. (a2) Original image of the pliers. (b2) The pliers contour change is caused by the pliers non-zero depth and relatively big angle of the camera inclination. The white curves on image a2, b2 highlight the places where the contours were most changed.

detectable (knife – Fig. 12a2), was recognized correctly in almost all the cases.

Although we used relatively small number images, the computation of the results took huge amount of time. The explanation is easy. Every picture has three mega pixels. The object border length was approximately 2000 pixels. All the preprocessing – dynamic threshold, border detection, border smoothing, inflection point detection and getting shred representation, requires many time consuming (especially in Matlab) operation. Let us assume that object A consists of a shreds and object B consists of b shreds. The substring matching algorithm takes much time too, because its complexity is huge as we mentioned in Section 4.3.

6. Conclusion

In this paper we have introduced a new method which is suitable for occluded object recognition, if the objects possess inflection points. We suggested a simple penalty function and we have presented the results of the recognition, comparison to the Lamdans method, robustness to additive gaussian noise and to inaccurate detection of inflection points.

Further development of the algorithm would involve the automatic determination of the number of iterations of the approximation algorithm. The new way of matching according to the different iteration level approximations of objects is also encouraging.

Acknowledgements

This work has been supported by the Czech Ministry of Education under the project no. 1M0572 (Research Center DAR) and by the Grant Agency of the Czech Republic under the project no. 102/04/0155.

References

Ansari, N., Delp, E.J., 1990. Detectiong dominant point. Pattern Recognition 24, 441–451.

Cohen, F.S., Huang, Z., Yang, Z., 1992. Curve recognition using B-spline representation. ACV 93, 213–220.

Cohen, F.S., Huang, Z., Yang, Z., 1995. Invariant matching and identification of curves using B-spline curve reprezentation. IEEE Trans. Image Process. 4 (1).

Duda, R.O., Hart, P.E., 1973. Pattern Recognition and Scene Analysis. John Wiley, New York.

Han, Min-Hong, Jang, Dongsig, 1990. The use of maximum curvature points for the recognition of partially occluded objects. Pattern Recognition 23, 21–33.

Kawaguchi, T., Nagao, M., 1998. Recognition of Occluded Objects by a Genetic Algorithm. ICPR98.

Lamdan, Y., Schwartz, J.T., Wolfson, H.J., 1988. Object Recognition by Affine Invariant Matching. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 335–344.

Mannocha, D., Canny, J., 1992. Detecting cups and inflections in curves. Comput. Aided Geom. Design 9, 1–24.

Mokhtarian, F., Abbasi, S., 2002. Shape similarity retrieval under affine transforms. Pattern Recognition 35, 31–41.

Mokhtarian, F., Mackworth, A., 1986. Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE Trans. Pattern Anal. Machine Intell. 8, 34–43.

Pikaz, A., Dinstein, I., 1995. Matching of partially occluded planar curves. Pattern Recognition 28 (2), 199–209.

Richards, W., Hoffman, D.D., 1985. Codon constraints on closed 2d shapes. CVGIP 31 (2), 156–177.

Rothwell, C., Zisserman, A., Forsyth, D., Mundy, J., 1991. Using Projective Invariants for constant time library indexing in model based vision. In: Proc. British Machine Vision Conference.

Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W., 1998. Shock graphs and shape matching. Computer Vision, 222–229.

Teh, C.H., Chin, R.T., 1989. On the detection of dominant points on digital curve. IEEE Trans. Pattern Anal. Machine Intell. 12, 1072–1079.

Tieng, Q.M., Boles, W.W., 1997. Wavelet-based affine invariant representation: A tool for recognizing planar objects in 3d space. IEEE Trans. Pattern Anal. Machine Intell. 19 (8), 846–857.

Tsang, P.W.M., Yuen, P.C., Lam, F.K., 1994. Classification of partially occluded objects using 3-point matching and distance transformation. Pattern Recognition 27 (1), 27–40.

Turney, J.L., Mudge, T.N., Volz, R.A., 1985. Recognizing partially occluded parts. IEEE Trans. Pattern Anal. Machine Intell. 7 (4), July.

Weiss, I., 1992. Noise resistant projective and affine invariants. Proc DARPA Image Understanding Workshop pp. 683–692.

Werman, M., Weinshall, D., 1995. Similarity and affine invariant distances between 2D point sets. IEEE Trans. Pattern Anal. Machine Intell. 17 (8), 810–814.