Curve Partitioning and Abstraction with the Local/Global Space

Christoph Rasche

Affiliation and mailing address:

Image Processing Laboratory Universitatea Politehnica din Bucuresti Bucuresti 061071, RO email: rasche15@gmail.com fax: +4021 402 4821, phone: +4021 402 4645

Abstract—A novel method for the partitioning and geometric characterization of curves is presented. The method starts by generating a multi-resolution amplitude space using labeling functions which identify curved, inflexion and straight segments. Partitioning is based on the detection of curved segments, by analysing the change in amplitude along the resolution axis. Abstraction of partitioned segments occurs by forming spectra of the space (along the resolution axis), from which global and local attributes are derived (alternating, angularity, jaggedness, etc). The attributes can be used to span a low-dimensional space in which a wide range of curve geometries and hence features is expressed. The methodology is essential for the rapid construction of mid- and higher-level features.

Index Terms—curve description, curve partitioning, multiresolution, feature representation, low-dimensional

I. INTRODUCTION

The goal of curve partitioning is to compartmentalize an (open or closed) curve into segments such that it permits the proper grouping (clustering) with other curve segments, which in turn would lead to a part-based shape recognition. This goal was originally pursued using the curvature scale space (e.g. [1], [2], [3]), but because this space can not deal well with wiggly curves, the approach was never developed to its full potential. Instead, the methodology for point-topoint (correspondence based) matching of curves was elaborated, for example for open (or closed) curves [4]; for shape retrieval [5], [6] (closed curves); and for object recognition [7], [8]. Correspondence-based matching is computationally demanding, with $O(N^2)$ and $O(N^3)$ complexity for open and closed curves, respectively (N=number of curve points), but the method can be very accurate. Yet, for exact grouping and hence construction of mid- or higher-level features, it requires the precise detection and localization of 'elementary' segments, e.g. approximate circular arcs and straight lines; and it requires a precise estimation of the segment's geometric parameters.

The goal of curve abstraction is to find a compact representation which is suitable for fast comparison of curves (in particular the elementary segments), e.g. a low-dimensional space in which curves can be compared by a mere distance measure. For instance, Gorman et al. represent curve segments by a few Fourier descriptors [9], which represents the desired compactness, as matching in this case is reduced to $O(K^2d^2)$ complexity (*K*=number of segments; *d*=dimensionality; *Kd* often < N). There exists a number of other curve descriptions based on polygons, e.g. the shape tree [5], the curve evolution for closed curves [10]; or the one presented in McNeill and Vijayakumar's study [6], which also exploit a multi-resolution 'view'. These descriptions do not really represent an abstraction, as some of them do not really reduce information. But all these approaches lack the possibility of part interpretation (with exception of [10]), so do the above mentioned studies pursuing correspondence-based matching.

Here we present methodology, which pursues both goals using a novel multi-resolution space, namely an amplitude space, where the amplitude is the distance between a selected subsegment and its chord. The amplitude space allows a precise localization of elementary segments and a characterization of the segment's geometry, which is suitable for a lowdimensional representation. The methodology was previously introduced as part of an image classification study [11], but rather in its infancy. Meanwhile it has matured and has also been applied in video indexing [12], image retrieval [13], gesture (posture) identification [14]; here we add in particular results from shape retrieval and classification of land-use satellite images.

A. Overview Methodology

To quicker introduce our methodology it is useful to mention how the popular curvature scale space is generated: a curve is repeatedly low-pass filtered and at each point in the space the curvature is determined [1], [2], [15]. However, the low-pass filtering operation represents a loss of structural information which "smears" a corner's precise position and consequently does not permit proper discrimination between 'noisy' shapes, e.g. between a jagged circle and a jagged square. Related criticism was already given by Fischler and Bolles and those researchers proposed an alternative space in which a curve is systematically labeled for increasing window sizes *omitting* the low-pass filtering operation [16]. We call this the local/global space - as opposed to the fine/coarse scale of the curvature scale space. The local/global (LG) space was recently suggested again by Cremers' group (subsection 4.1 in [17]). The term multi-resolution space is sometimes associated with the low-pass filtering operation, that is why we prefer the term local/global space, which should strictly stand for varying window size without any modification of the curve. In the LG space, the amplitude (for a subsegment) is measured at each point.

Partitioning was typically understood as the segregation (splitting) of curves at points of highest curvature into disjoint sets [16], [1], [2]. We attempted to pursue this direction in our previous study, but meanwhile we discovered that the resulting segments were often not homologue across instances of the same class (which we found out on the shapes of set B of the MPG7 collection). By homologue we mean that the partitioned segments are the same across instances despite the presence of intra-class variability in structure, as it exists in the MPG7 classes. Here, we pursue a novel partitioning procedure that focuses on the detection of curved arc segments (or simply 'arcs'). An arc is defined as an (approximate) circular arc or a corner (L feature) of varying angle; any curve possesses such arc types, unless it is just a straight line. We can detect such segments relatively easy by analyzing the LG space along the resolution axis and that results in much more homologue partitioning.

To pursue abstraction, the space requires some form of simplification from which global geometric attributes can be derived (section IV). For that purpose, the LG space is converted into various spectra, which capture the alternating course of a curve (akin to the Fourier spectrum for sinusoidal signal analysis). From those, a number of local and global attributes are derived, which then are employed as components of a low-dimensional vector. For these attributes a real value is determined, in order to avoid a strong classification of curve geometry. This prevents an 'early' feature classification and allows to build more flexible representations that can deal with the structural variability (see also [11] for arguments).

In summary, for a given, unpartitioned, planar curve (open or closed), the partitioning and abstraction process proceeds as follows:

- 1) Generation of the LG space for a list of 2D coordinates.
- 2) Detection of (curved) arcs by analyzing the space along the resolution axis.
- Abstraction of arcs by selecting the corresponding subspace of the LG space generated in step 1 and by manipulating it to derive a geometric characterization.

Sections II, III and IV correspond to these 3 steps. In section V we report about the shape retrieval results.

II. THE LOCAL/GLOBAL SPACE

For the purpose of structural description, a planar curve can be regarded as an alternating sequence of non-disjoint arcs and inflexions (change of sign; transition): for instance a wiggly (often natural) curve consists of an irregular alternation of arcs and inflexions; an 'oscillating' curve consists of an even alternation; an arc or L feature consists of only a single bow (with no inflexion). To detect this alternation (or lack of), the curve is transformed with labeling (kernel) functions, which identify bows and inflexions for different window sizes (explained in subsection II-A). These transformations take place on a curve (contour) that has been extracted as a list of 2D coordinates $\{x_i, y_i\}, i = 1, ..., n_{Coord}$, but whose output - a signature - is set in relation to the arc length variable v. For a given window size, the labeling leads to a signature (formalized in subsection II-B); for a bandwidth of window sizes, this leads to the LG space (subsection II-C). We give an example of an implementation of the space in subsection II-D.

A. Labels

A (open) curve c(v) is transformed with three labeling functions, the 'bowness'-, the inflexion- and the straightnesslabeling function. The bowness- and inflexion-labeling functions are necessary to determine the exact course of the curve; the straightness-labeling function is necessary to discriminate between a straight segment and an alternating ('oscillating') segment.

A labeling function L selects a window of the curve, determines whether the selected subsegment¹ is of a certain geometry and returns its magnitude. The bowness-labeling function L_{β} determines whether the subsegment is a bow and thus possibly part of an arc -, and returns its amplitude as a magnitude measure. The term **bow** is specifically used when referring to any bend outlined by a bowness block. The inflexion-labeling function L_{τ} determines whether the subsegment is an inflexion and returns also the amplitude. The straightness-labeling function L_{γ} determines whether the subsegment is sufficiently flat and returns a magnitude which is proportional to the amplitude.

The amplitude is measured as follows (see also figure 1a and b): given a subsegment $s = [v_1, v_2]$ of fixed length ω , the amplitude κ is defined as the maximal displacement between the subsegment and the chord, normalized by ω . To determine whether the subsegment is straight, one can define a maximal amplitude value. But to discriminate between the bow and the inflexion geometry, the course of the subsegment needs to be analyzed more specifically. In principal this can be done by determining the laterality of the subsegment in reference to the chord: if all points lie on the same side, then the subsegment is labeled bow; if the points of the first and second half of the subsegment lie on opposite sides, then the subsegment is labeled inflexion. Due to presence of irregularity and noise, it is necessary to relax these laterality conditions and include a certain tolerance T.

For the bowness label (figure 1a), this tolerance is included by determining laterality for only a central part of the subsegment, $s_{cen} = [v_{c1}, v_{c2}]$, for which all points have to lie exclusively on the same side of the chord, short-noted as $s_{cen} \triangleq [\frown]$. The length of this central part represents the tolerance T_{β} : if T_{β} is too large then the labeling is intolerant to any noise; if T_{β} is too small, the labeling is unspecific and does not allow to find arcs. In addition to the laterality condition, the amplitude κ requires a minimum displacement larger than a threshold S_{κ} , otherwise the bowness-labeling function L_{β} is set to 0:

$$L_{\beta}(s) = \begin{cases} \kappa & , s_{\text{cen}} \triangleq [\frown] & \bigcap & \kappa > S_{\kappa} \\ 0 & , \text{else.} \end{cases}$$
(1)

To relax the laterality condition for the inflexion label L_{τ} , only a proportion T_{τ} (the tolerance) of each half is required

¹In our previous publication we used the term segment for what is now called more specifically the *subsegment*, because the term segment is also used to refer to elementary curve segments.

to lie on opposite sides of the chord. The inflexion condition is now short-noted as $(s_{h1} : s_{h2}) \triangleq [\sim]$, whereby s_{h1} and s_{h2} are the two segment halves $(s_{h1} = [v_1, v_m]$ and $s_{h2} = [v_m, v_2]$; see figure 1b). If the condition is true, then the magnitude of the inflexion-labeling function L_{τ} corresponds to the amplitude:

$$L_{\tau}(s) = \begin{cases} \kappa & , (s_{h1} : s_{h2}) \triangleq [\sim] \\ 0 & , \text{else.} \end{cases}$$
(2)

For the straightness label L_{γ} , the tolerance T_{γ} sets the maximal allowable displacement, which is dependent on the given window size ω . In addition, the straightness label is only present, if at the same location no bowness label is present:

$$L_{\gamma}(s) = \begin{cases} \frac{T_{\gamma} - \kappa}{T_{\gamma}} & , \kappa < T_{\gamma} \quad \bigcap \quad L_{\beta} = 0\\ 0 & , \text{ else.} \end{cases}$$
(3)

The bowness and inflexion label are mutually exclusive, as well as the bowness and straightness label; the inflexion and straightness label can co-occur.



Fig. 1. Labeling a subsegment of a window. The selected subsegment ranges from v_1 to v_2 . κ : maximal displacement (normalized by window size). **a**. For the bowness label, the central part of the subsegment, $[v_{c1}, v_{c2}]$, needs to lie exclusively on one side of the chord. **b**. For the inflexion label, the first half $[v_1, v_m]$ and the second half $[v_m, v_2]$ are required to lie on opposite sides $(v_m = \text{subsegment midpoint})$.

B. Signatures and Block Characterization

The labeling functions are shifted along the curve $\mathfrak{c}(v)$ returning the bowness, inflexion and straightness signature, β , τ and γ . For instance for the bowness signature we have

$$\beta(v) = L_{\beta}(s)\mathfrak{c}(v)\mathrm{d}v,\tag{4}$$

where s is - as defined above - the neighborhood (window) of the labeling function of size ω ; dv denotes the shift. The signature is set to zero for the boundaries where the chord cannot be applied. The inflexion and straightness signatures, $\tau(v)$ and $\gamma(v)$, are generated analogously. All three signatures can be called amplitude signatures. An interval, where the signature has support, is now called *block* or signature block. It is particularly the *bowness block* (range denoted as v_{\Box}), which is of interest. A block $\beta(v_{\Box})$ is geometrically characterized by a number of parameters, which later are employed for partitioning and abstraction. Two measures are taken:

• **Circularity**, ζ^{\square} : is defined as the integral of the bowness signature:

$$\zeta^{\Box} = \int_0^{l_{\Box}} \beta(v_{\Box}) \mathrm{d}v, \qquad (5)$$

which is an approximation because the signature block does not span the entire outlined arc, but the computation of this measure is fast.

• Edginess (angularity), ϵ^{\square} : the measure allows to distinguish to what degree the block is a L feature or a circular arc. It is determined by multiplying the derivate of $\beta(v_{\square})$ by a normalized, ramp function $r(v_{\square})$, whose width is equal to the block length (with center value equal 0):

$$\epsilon^{\Box} = \int_0^{l_{\Box}} \beta'(v_{\Box}) r(v_{\Box}) \mathrm{d}v.$$
 (6)

The edginess value is largest for a L feature, it is 0 for a circular arc and negative for a flat arc (e.g. the elongated half of an ellipse). It would be sensible to separate the positive and negative range and call the negative range the degree of 'flatness' for instance.

A bowness block has a minimum length, set by a tolerance $T_{l_{\square}}$, which depends on window size ω . This minimum length is necessary because the above geometric description for very short segments is meaningless and setting null values for the geometric attributes would distort some of the spectra and hence the abstraction.

C. The Complete LG Space

To generate the LG space, the above signatures are computed for a set of window sizes, $\omega \in [\omega_{\min}, l_c]$, with ω_{\min} the minimum window size and l_c the total arc length of the curve:

B
$$(\omega, v)$$
, **T** (ω, v) , **T** (ω, v) , $\omega \in [\omega_{\min}, l_{\mathfrak{c}}]$

Because we will later also operate with the signature blocks, in particular the bowness block space, we use

$$\mathbb{B}(\omega, v)$$
 for blocks \sqcap of $\mathbf{B}(\omega, v)$,

and refer to those as the block space (for minimal-length blocks only). When the individual block characteristics are accessed, we refer to them as

$$\mathbb{B}^{\zeta}(\omega, v), \quad \mathbb{B}^{\epsilon}(\omega, v),$$

for circularity and edginess. In analogy to the expression 'curvature' scale space, one could call these spaces amplitude scale spaces.



Fig. 2. Local/global (LG) space for a complex shape. **Top right**: center point marked as gray (green) asterisk; detected arcs are outlined by their region (shading $\propto \xi^{\Box}$); squares denote elementary straight segments, which are placed at the segments' midpoints. **Left column**: LG space: signatures $\beta(v)$ (black), $\tau(v)$ (grey) and $\gamma(v)$ (stippled) for 14 different window sizes [sizes from 5 to 319 pixels; x-axis=arc length variable v]. The curve was extended on each side by 210 pixels due to closeness. (Minimal-length) bowness block characteristics: (blue) triangle= ϵ^{\Box} (upward-pointing=positive value; down-ward pointing=negative value); (green) diamond= v^{\Box} ; plus sign= ζ^{\Box} . Detected arcs are outlined with horizontal (dashed, magenta) line at ca y=0.4. Bottom plot summarizes the degree of angularity (triangles) and the straight segments (squares) at y=0. **Graph** Φ : solid line, black circles= Φ^{β} ; gray stars= Φ^{τ} ; dotted, black circle: max_v **B**; dotted, + marker: $Z(\omega)$. **Partitioned**: 'Summary' of the detected arc segments (black=arc; gray=straight). Triangles and squares at y=1.1 and y=1.3 represent angularity and straightness. **Selected**: An alternate 'summary' which outlines the exact range of each detected arc by a horizontal black bar (gray for straight segments). β , τ and γ , ϵ : space integrated along window (resolution) axis, bowness (black), transition (gray) [top panel]; edginess (blue) and straightness (gray) [bottom panel]. **Lower right panel**: some attribute values.

D. Implementation and Results

Window sizes were generated in increments of $\sqrt{2}$. The subsegment was merely selected in pixel units for simplicity and thus suffers from the aliasing problem. The smallest window sizes were always $\omega = [5, 7, 9, 11, 13]$ for scale $\sigma = 1$ and $\omega = [5, 9, 13]$ for coarser scales; scale here refers to the degree of image smoothing for the purpose of contour extraction and has no relation to ω . The labeling tolerances were $T_{\beta} = \omega \cdot 0.5$, $T_{\tau} = \omega \cdot 0.25$ and $T_{\gamma} = \omega \cdot 0.05$. Those choices are not sensitive but systematic testing was not carried out. S_{κ} was set to a low, fixed threshold, $\sqrt{(2)/2}$, to allow the detection of wide circular arcs. Minimum block length tolerance $T_{l_{\square}}$ was set to $\omega/2$ for the first half of the window sizes (for a given LG space) and to $\omega/4$ for the second half; thus for larger windows the tolerance for accepting blocks is higher.

The LG space for simple curve types (arc, inflexion and wiggly) was already shown in the supplementary material in [11] (figures 1-3 therein). The left column of figure 2 shows the LG space for a complex MPG7 shape shown in the upper right (shaded areas correspond to the consistency measure).

The computation of the LG space takes ca. 900ms for a typical silhouette of the MPG7 shape database (several hundred pixels) on a 2.66 GHz Intel Pentium, implemented in Matlab, whereby the amplitude signature is generated exploiting matrix operations. For all the contours obtained from a gray-scale image of size 200x300, the computation duration is only 100-200ms as contours are typically shorter.

III. PARTITIONING

To understand how arcs can be detected, it is useful to regard the space as a landscape surface. For an approximately circular arc, e.g. a quarter circle, the surface describes an increasing plane towards the global window level. If the circular arc is embedded in a curve, for instance as in an Ω shape, then the surface resembles a mound. For a semiturn (U or V feature) the surface describes a peak; for a wiggly curve, the surface consists of irregularly placed narrow mounds and small peaks. It is tempting to think that one could discriminate between different arc types by constructing a twodimensional filter. But due to the large variety of possible curve geometries, the partitioning operations should not rely on predefined geometries. Instead, one merely determines the change in amplitude along the window dimension (resolution axis) and if it is small, then there exists an arc type. In other words, arc detection intentionally lacks any specific geometric filter in order to flexibly detect any arc type. The measure of change is specified in subsection III-A and is called the 'consistency' measure.

That only arc types are detected - and not arbitrary forms - is guaranteed by the use of the laterality condition, $s_{cen} \triangleq [\frown]$, in equation 1. Arcs can be non-disjoint, e.g. an inflexion segment consists of two overlapping arcs. Thus, instead of pursuing the traditional strict segregation, the understanding of partitioning here is that of a process that localizes non-disjoint partitions (arcs) without segregating the curve.

In summary, an **arc** (**segment**) is thus more specifically defined as a piece of curve, whose points lie on the same side

of the segment's chord; it is approximately symmetric and can consist of a wide range of geometries, for instance a L feature, a circular arc or a semi-turn (U or V feature). Arcs with an absolute amplitude larger than half the chord length are called **elongated** (U or V shapes for instance).

Arc detection could be driven to the lowest possible resolution, that is until the smallest arc has been extracted. It thus requires a stopping rule that terminates the partitioning process because the generation of unnecessary small segments is a computational burden for the matching process that is preferably avoided. The stopping rule would ideally distinguish between jaggedness that is caused by noise (e.g. imprecise edge detection or natural contours) and wiggliness (or irregularity) that is characteristic to the category (e.g. person's silhouette contour, see figure 8 in [11]). In case of jaggedness, the algorithm would stop; in case of wiggliness, the algorithm would extract elementary segments that may be useful for grouping. Such a decision is best made after a perceptual organization and in combination with an analysis of the spectra introduced in section IV. We therefore defer this issue and assume for now that some jaggedness threshold Θ_{iag} exists, which represents a window level (size).

A. Consistency

Consistency is determined by comparing the amplitude for a given bow with the amplitude at adjacent levels observed along the resolution axis. If no bows can be found at localer (more local) and globaler (more global) levels at the same spatial position, then the bow has zero consistency and is accidental and part of a wiggly segment. The consistency is maximal if the bowness signature is the same at localer and globaler levels: it is then a part of an arc. Or expressed in bowness space, the bow consistency ξ^{\Box} is inverse proportional to the change within a limited interval $[\omega_{\Box} - a, \omega_{\Box} + a]$ across the window dimension, with ω_{\Box} the window length (level) of the selected block \Box :

$$\xi^{\Box} \propto \left(\int_{v=0}^{v=l_{\Box}} \int_{\omega_{\Box}-a}^{\omega_{\Box}+a} \frac{\partial \mathbb{B}(\omega,v)}{\partial \omega} \mathrm{d}\omega \mathrm{d}v\right)^{-1}.$$
 (7)

The measure of bow consistency is not sufficient in itself for reliable arc detection, because it can lead to an emphasis of local bows in a jagged curve for instance. To avoid the selection of such small arcs, the consistency measure is correlated with block length l_{\Box} and this new measure is called the block energy η^{\Box} :

$$\eta^{\Box} = \xi^{\Box} \cdot l_{\Box}. \tag{8}$$

The block energy measure captures the 'significance' or confidence of the block within the LG space; the more consistent and longer the block, the larger the chance that it outlines a distinct part of the structure. The energy space is abbreviated as \mathbb{B}^{η} .

A consistency measure for straightness is more difficult to develop, because smooth curves are locally straight.

B. Arc/Straighter Detection Algorithm

The arc detection procedure successively selects the bow with the maximal energy value from \mathbb{B}^{η} until the entire, unpartitioned curve has been checked for arcs, resulting in a list of arc partitions \mathcal{E}^{arc} . For each selection, a corresponding neighborhood (ω_k, v_k) in \mathbb{B}^{η} is suppressed (set to 0). Selected bows, whose window level is smaller than the jaggedness threshold Θ_{iag} , are omitted. After all curved arcs have been analysed $(\mathbb{B}^{\eta} = 0 \text{ everywhere})$, there may remain unpartitioned curve segments, that typically are straight or wiggly, but that can also be slightly curved. We call these segments straighter (list denoted as \mathcal{E}^{str}), as they are straighter in comparison to their adjacent, curved segments. In figure 2, the belly of the animal is outlined by a slightly curved segment, which remained after arc detection as straighter segment. Or exemplifying the algorithm on the rectangle shape: after all four corners were extracted, there remain the two longer sides (in this case truely straight segments). As boundaries for the straighter segments, we take the midpoints of the adjacent, detected arcs. We summarize the algorithm as follows (algorithm 1):

Algorithm 1 Arc/straighter detection exploiting energy (block) space \mathbb{B}^{η} . See subsection IV for jaggedness threshold Θ_{jag} . $v_{\hat{n}}$ and $\omega_{\hat{n}}$ are the center of the detected arc.

-Input $: \mathbb{B}^{\eta}, \Theta_{jag}$: \mathcal{E}^{seg} -Output {list of segment ranges} $\mathbb{B}^{\eta}=0$ for $\omega = \omega_{\min}$ and $\omega = \omega_{l_s}$; {1st and last level to 0} $\mathbb{B}^{\eta}=0$ for ω where $\Phi^{\beta}=1$; {omitting circular arcs} repeat $(\omega_{\widehat{\sqcap}}, v_{\widehat{\sqcap}}) \leftarrow \operatorname{argmax}_{\omega, v} \mathbb{B}^{\eta}[\omega, v];$ $\omega_k = [\omega_{\widehat{\sqcap}} - a, \omega_{\widehat{\sqcap}} + a];$ $v_k = [v_{\widehat{\square}} - b, v_{\widehat{\square}} + b];$ $\mathbb{B}^{\eta}[\omega_k, v_k] = 0;$ {suppression} if $\omega_{\widehat{\sqcap}} < \Theta_{jag}$ then continue; end $\mathcal{E}^{arc} = \mathcal{E}^{arc} + \{v_k\}$ until all $\mathbb{B}^{\eta}=0$; $A(v) = \sum \mathcal{E}^{\operatorname{arc}};$ {detect arc 'coverage'} $\mathcal{E}^{\text{str}} \leftarrow \text{where } A(v) = 0;$ {straighter where no arcs} $\mathcal{E}^{\text{seg}} = \mathcal{E}^{\text{arc}} + \mathcal{E}^{\text{str}}$

The output of the algorithm is the combined list of arc and straighter ranges (v_k for each segment), meaning the abstraction is the same for both types of extracted segments.

C. Implementation and Results

Bow consistency (equation 7) was implemented by merely correlating the signed bowness signature of the next globaler and localer window level:

$$\xi^{\sqcap} = \frac{1}{l_{\sqcap}^2} \sum_{v=1}^{v=l_{\sqcap}} \operatorname{sgn} \left[\mathbf{B}[\omega-1,v] \right] \cdot \operatorname{sgn} \left[\mathbf{B}[\omega+1,v] \right] \quad (9)$$

The suppressed neighborhood $[\omega_k, v_k]$ has a spatial width equals the (detected maximum) block width $(b = l_{\Box}/2)$; the window width was set with a = 2 (values larger or smaller than 2 resulted in lower performance).

The shading of the bowness blocks (black silhouette) corresponds to the consistency value. For the two most global window sizes (no. 13 and 14), no consistency was detected and those bowness blocks have therefore a white face color. The straightness label's maximum is set to a value of 0.5 for illustration purposes (to lower confusion with the other labels).

To visualize a detected arc, its region (closed by the arc's chord) is shaded (upper right graph of figure 2). Elongated arcs are for instance the snout and tail of the animal shape. An elongated arc can for instance be a V, U or half-rectangular shape. As elongated we define if the (absolute) amplitude exceeds the radius treating the arc segment as a circular arc.

The two center columns of figure 3 show the arc/straighter detection output for some more shapes (left and right column respectively). (The shapes in that figure were selected to demonstrate a variety of spectrum courses, see section abstraction IV). Straighter segments are sometimes detected in the center part of circular arcs that subtend a small angle.

The performance was verified on the entire MPG7 collection (set B with 70 classes, 20 instances each, with each shape being a closed silhouette). Figure 4 shows detected arcs for some classes, whereby we selected preferably animal shapes as they are more difficult to partition than simple geometric shapes. Partitioning performance is largely homologue, but there remains occasional lack of homology, such as the 'miss' of detected arcs at a global level. For instance, the 4th face (no. 684) in figure 4 lacks the detection of the forehead as a whole (compare to others); the 4th turtle lacks the detection of the back. These global 'dishomologies' are caused by the larger increments for more global window sizes (increment of $\sqrt{2}$).

The results for all 1400 shapes can be found in the supplementary material. The arc-detection procedure successfully detects all spiral shapes (see class 'spring', figure 2 in supplementary material, row no. 64).

D. Discussion

The occasional global 'dishomologies' may be avoided by several, different measures. One measure could be the use of a smaller increment, which however would increase computation time. Another possibility is the development of an 'adaptive' bow consistency measure that takes the increment into a account, e.g. a suppression range ω_k that is a function of the increment. Lastly, partitioning may be optimized by a perceptual grouping process, in which case it becomes rather an issue of 'mid-level' processing or shape recognition.

IV. ABSTRACTION

We nominally distinguish between global and local attributes (subsections IV-B and IV-C, respectively). The global attributes describe the curve geometry as a whole and so far we have concentrated on three principal geometric types: *arc*, *inflexion* and *alternating*, which correspond to a segment with zero, one and multiple inflexion(s), respectively. These types represent extrema in a continuum of different geometries, thus segments are not classified into these types but are given a graded value expressing the degree of these geometries; we call this *typification* or *typified*. Type arc is found during the arc detection loop (algorithm 1), but we exlcude elongated arcs



Fig. 3. Spectra and detected arcs of some complex shapes. Left most column: Similar to graph Φ in previous figure: solid line, gray circles= Φ^{β} ; solid line, gray circles with black face= $H(\omega)$; dashed, x marker (blue): R^{Φ} ; red dashed, vertical line: Θ_{jag} . 2nd column: detected arcs outlined as quasi circular segment. The shading indicates the degree of block consistency ξ^{\Box} . 3rd column: detected straighter segments. Right column: bowness blocks of detected arcs (for selected window; equals the graph 'Partitioned' in the center of right column of figure 2).



Fig. 4. Arc detection in MPG7 shapes (row-wise: bat, bird, camel, carriage, helicopter, face and turtle). A shaded region outlines an arc; the darkness of the shading corresponds to the consistency measure (dark=consistent).

as the abstraction derived from the LG space is not suitable for them (elongated shapes represent rather shapes). Types inflexion and alternating are found with the straighter segment detection step. The local attributes rather describe the fine structure of a curve such as bendness, jaggedness, edginess etc.

The abstraction is generated for the subspace as outlined by the detected arc range. More specifically, for the list of segment ranges \mathcal{E}^{seg} , the abstraction is generated for the subspace from the LG space of the unpartitioned segment given by each range v_k . If a range covers the entire unpartitioned curve - and is thus a single arc or straight line -, then the entire space is used (and no subspace is selected).

A first step towards abstraction is the creation of some spectra, which are formed by integration along the arc length variable. The spectra are necessary to derive some of the geometric attributes and are therefore introduced first (subsection IV-A). Then, the definitions of the global and local attributes are explained (subsection IV-B and IV-C). While the definitions are intuitive given the way the LG space and the spectra are built, it required substantial heuristic tuning for some of the parameters, partly due to structural exceptions like the spiral shape. Tuning was performed and verified

with a set of artificially generated curves (subsection IV-D, see also supplementary material). To test the representation performance on real (actual) curves, curve sorting (search) was carried out (subsection IV-E), as well as shape retrieval, which is reported in a separate section (V).

A. Spectra

A spectrum is generated by performing an operation along the arc length dimension of the LG space, e.g. an integration $Q(\omega) = \int_{v=0}^{v=l_c} \mathbf{B}(\omega, v) dv$. For each of the three spaces (**B**, **T**, **T**) a spectrum is formed, also called *fraction functions* (IV-A1), which allow already a typification of the curve's global geometry. To estimate in particular the alternating characteristics, two *alternating* spectra are formed (IV-A2), the ratio function and the energy spectrum, which can be loosely understood as frequency spectra.

1) Fraction Functions: They are called such, because their spectrum is normalized by the curve length,

$$\Phi^{\beta}(\omega) = \frac{1}{l_{\mathfrak{c}}} \int_{0}^{l_{\mathfrak{c}}} \mathbf{B}(\omega, v) \mathrm{d}v, \qquad (10)$$

as an example for the bowness fraction function. The inflexionfraction function is formed and labeled analogously: $\Phi^{\tau}(\omega)$. The course of the fraction functions differs as follows (see also figures 1-3 in supplementary material of [11]). For an arc, Φ^{β} increases with increasing window size, whereas Φ^{τ} decreases; the rate of increase and decrease depends on the degree of smoothness (or 'wiggliness') of the arc. For a curve consisting of a single inflexion, the course of the bownessand inflexion-fraction function is reversed (Φ^{β} decreasing, Φ^{τ} increasing). For an irregular (wiggly) curve, the bownessfraction function describes a bump with its maximum located at a medium window level, whereas the maximum for the inflexion-fraction function occurs at a higher level. Thus, the course of the fraction functions expresses the global curve geometry and is suitable for defining the global attributes (subsection IV-B).

2) Alternating Spectra: The two spectra allow to estimate a jaggedness threshold and the extent of alternation, on a local and global level. Initially, we intended to do this with a single spectrum only, but eventually developed two spectra, the ratio function R^{Φ} and the energy spectrum H. Each one has benefits.

The ratio function is built from the ratio of the inflexion and bowness fraction functions. Firstly a simple ratio ρ is generated for those values of the inflexion fraction function, that are smaller than the values of the bowness fraction function, as an emphasis of the bowness 'behavior' is sought:

$$\rho(\omega) = \begin{cases} \Phi^{\tau}(\omega)/\Phi^{\beta}(\omega) &, \Phi^{\tau}(\omega) < \Phi^{\beta}(\omega) \\ 1 &, \text{else.} \end{cases}$$

The simple ratio is then directly weighted by the bowness fraction function to form the actual ratio R^{Φ} :

$$R^{\Phi}(\omega) = \rho(\omega)\Phi^{\beta}(\omega). \tag{11}$$

The ratio R^{Φ} can be a multimodal function. If the curve is aliased, due to the use of a simple edge-detection and

extraction algorithm for instance, then the first mode reflects the aliasing noise. If the curve is alternating such as in a sinusoidal or in a natural, wiggly curve, then the second mode (or first mode when no aliasing present) reflects the 'actual' frequency respectively. For complex shapes, there can be more than two modes, which are not necessarily clearly distinguishable. The first mode of the ratio function is abbreviated as,

$$R^{\Phi}_{\max 1} = \max_{\omega} R^{\Phi}(\omega) \tag{12}$$

and will be later used for determining attributes jaggedness and alternating.

The location of the alternating mode depends on the alternating frequency. For a frequently alternating curve, the mode occurs at a lower window level; vice versa, for a slowly undulating curve, the mode occurs at a higher window level. The amplitude depends on the extent of alternation at that window level: for instance a segment, that contains only a piecewise high-frequency alternation, has a mode occurring at a lower window level and with lower amplitude. The ratio function is used to estimate the amount of alternation at a global and local level (attributes alternate and jaggedness respectively).

The second alternating spectrum, the energy spectrum $H(\omega)$, is generated by integrating the energy values:

$$H(\omega) = \int_0^{l_{\epsilon}} \mathbb{B}^{\eta}(\omega, v) \mathrm{d}v.$$
(13)

The energy spectrum can also be multimodal, but we found that it allows to estimate a jaggedness threshold better than the ratio function, because the first mode seems to appear more distinct and more appropriately located than the first mode of the ratio function (neither too close to 0 nor too large). The jaggedness Θ_{jag} threshold is defined slightly different depending on the number of modes:

$$\Theta_{jag} = \begin{cases} \min(\operatorname{argmax}_{\omega}^{1} H(\omega), \Theta_{jag}^{upp}) &, \text{ H is unimodal} \\ \min(\operatorname{argmin}_{\omega}^{1} H(\omega), \Theta_{jag}^{upp}) &, \text{ else.} \end{cases}$$
(14)

If unimodal, then the window size of the first (local) maximum is taken, unless it is larger than an upper threshold, $\Theta_{jag}^{upp} = l_c/8$, where the value 8 represents the minimum number of cycles (for fewer cycles one would interpret the curve rather as alternating). If *H* is multimodal, then the window size of the first (local) minimum is chosen.

A consistency spectrum alone is limitedly meaningful because it can be dominated by jaggedness such that other modes are difficult to detect.

3) Implementation and Results: Φ is determined as the fraction of signature values that are above 0 for each window size,

$$\Phi^{\beta}[\omega] = \frac{1}{l_{\mathfrak{c}}} \sum_{v=1}^{v=l_{\mathfrak{c}}} \operatorname{sgn}[\mathbf{B}[\omega, v]], \qquad (15)$$

as an example for the bowness-fraction function.

The two graphs in the center of the right column of figure 2 display the spectra. The graph labeled Φ contains the bowness-, transition- and straightness-fraction functions as well as the ratio function R^{Φ} and the energy spectrum $H(\omega)$.

The left column of figure 3 shows the spectra of a set of shapes that produce relatively different spectra. The jagged, hexagonal star shape (row 2) generates a bimodal energy spectrum (solid circles), whose modes are distinct, of which the first mode clearly reflects jaggedness. The ratio function (x) and the inflexion fraction function (gray) also reflect the jaggedness, but we found that their modes were not as distinct and hence easily separable than the ones in the energy spectrum.

B. Global Attributes

• Arc, a: a curve is typified as arc if its bowness-fraction function has a value larger zero for the largest window size (l_c) :

$$a = \Phi^{\beta}(l_{\mathfrak{c}}). \tag{16}$$

As stated previously, the typification occurs not only for circular arcs, but also for L features or polygons that appear as approximately circular.

• **Transition**, *t*: analogously to the arc attribute, a curve is typified as inflexion (transition), if its inflexion-fraction function has a value larger zero for the largest window size:

$$t = \Phi^{\tau}(l_{\mathfrak{c}}). \tag{17}$$

• Alternating, w: the default value for w is equal to the transition value. If the curve is a perfect arc (a=1), then the first mode of the ratio function is taken (eq. 12). If the global curve is not a perfect arc (a <1) and shows a minimum value for the ratio function, then the value is determined as follows: the window level ω_{osc} is selected where the count of bowness blocks is largest ($n_{\Box osc}^{\Theta se}$) and its logarithm was taken:

$$w = \begin{cases} R_{\max 1}^{\Phi} &, a = 1\\ \log (n_{\square}^{\omega_{\text{esc}}}) &, a < 1 \ \cup \ R_{\max 1}^{\Phi} > 0.02 \\ t &, \text{else.} \end{cases}$$
(18)

• **Circularity**, ζ : this attribute allows to express full circles, which the bendness attribute (eq. 20) cannot distinctively capture as it also describes the amplitude of an alternating curve. It is defined as the maximum of the circularity space,

$$\zeta = \begin{cases} \max_{\omega, v} \mathbb{B}^{\zeta}(\omega, v) &, a > 0\\ 0 &, \text{else}, \end{cases}$$
(19)

because we assume that not more than one circular arc is present (in a partitioned curve), that subtends and angle of more than 180 degrees. A maximum operation should therefore suffice. Since the spectrum was generated with a maximum operation already, the value is equal the global maximum circularity value of all bowness blocks.

Case Spiral: a spiral (or G) shape shows the following characteristics. 1) Its bowness fraction function increases to a value of one (or nearly one) and then decreases again, because it represents an alternating curve at a very global level. 2) The bowness signature block gradually decreases (toward one curve end). These characteristics can be easily detected. The spiral's global parameter values are adjusted as follows: a = 1; $w = \max_{\omega} R^{\Phi}(\omega_{h1})$, with ω_{h1} being the first half of the spectrum; $\zeta = \max_{\omega,v} \mathbb{B}^{\zeta}(\omega, v)$.

C. Local Attributes

The local parameters are determined using the bowness block parameters (subsection II-B) and the spectra introduced in IV-A. One parameter describes the overall bendness of the curve, which can also be regarded as a global parameter in case the global geometry is of type arc: but it is a rather local attribute when the global type is transition or alternating. The jaggedness attribute represents the alternating characteristic at a very local level.

• **Bendness**, *b*: this attribute was somewhat misleadingly called 'curvature' in our previous study. It is essentially the normalized amplitude and approximates thus the degree of turn. It has two separate definitions, one for non-perfect arcs (a < 1) with a minimum amount of jaggedness $(R_{max1}^{\Phi} > 0.02)$ and one for remaining curves:

$$b = \begin{cases} \frac{1}{l_{c}} \int_{\omega} \left(\Phi^{\beta}(\omega) + \Phi^{\tau}(\omega) \right) \mathrm{d}\omega &, a < 1 \ \cup \ R^{\Phi}_{\mathrm{max1}} > 0.02 \\ \max_{v} \mathbf{B}[l_{c}, v] &, \text{else.} \end{cases}$$
(20)

The minimum jaggedness avoids that a too large bendness value is generated for oblique straight lines that suffer from aliasing. We had also attempted to define a bendness value, that was obtained from the global part of the space, but the definition tended to neglect high-frequency alternations.

• Edginess (angularity), e: this attribute allows to discriminate between smoothly undulating and zigzag. It is defined as in our previous study, namely as the average $\langle \rangle$ of all edginess values for all bowness blocks \mathbb{B}^{ϵ} in the entire space: $e = \langle \mathbb{B}^{\epsilon}(\omega, v) \rangle$.

• Jaggedness, j: estimates the alternating characteristic at a local level. If the curve has been globally typified as of type arc, the value equals the alternating value w (defined above). Otherwise a very local interval of the ratio function is integrated:

$$j = \begin{cases} w &, a > 0\\ \int_{\omega_{\min}}^{\omega_{\log}} R^{\Phi}(\omega) d\omega &, \text{else.} \end{cases}$$
(21)

D. Implementation and Verification

For the jaggedness attribute, the local interval ω_{loc} was set to $l_c/15$. Attribute values were normalized to unit range [0,1], when they were not defined within that range already.

A vector is formed with attributes as introduced above, $\mathbf{c}(o, l, a, w, \zeta, b, e, i, j)$, where o and l are the curve's orientation (o) and the total arc length $(l = l_c)$ - the transition attribute was not used here; i is the symmetry attribute as introduced in our previous study [11].

1) Verification: To facilitate and verify the tuning of parameters and definitions, a set of artificially generated test curves was used (see supplementary material). The coordinates of most test curves are quantized to integers in order to understand the effects of potential noise. Some curve coordinates are real-valued and those serve as a control.

The rotation of curves revealed that the amplitude of the bowness signature can be very high at lowest window levels. We refer to the supplementary material for details.

One could attempt to compensate for the irregularities caused by aliasing (such as the window length measured in



Fig. 5. Sorting curve vectors along different dimensions (attributes). In each row one dimension value was systematically increased (the found and not the preset value is displayed in the upper right of each graph). Top row: increasing (inc.) bendness value for an arc. 2nd row: inc. *b* for an angular arc (e = 1); 3rd row: inc. irregularity (symmetry; here y = i) for an arc. 4th row: inc. *b* for an alternating curve. 5th row: inc. *b* for an angular alternating curve (e = 1). 6th row: inc. *w*. 7th row: inc. irregularity (y = i) for an alternating curve.

pixel units), e.g. by using a look-up table, but this may only be necessary if high accuracy is required as for instance in an identification task.

The computation time for partitioning and abstraction takes ca. 400ms for a typical silhouette contour of the MPG7 shape database (2.66 GHz), thus less than half the duration for the generation of the LG space.

E. Results

The attributes were tested with curve sorting (retrieval), which follows now, and by looking at knock-out performance for a classification task, which follows in section VI.

Curves are sorted by searching for systematically preset values in contours obtained from gray-scale images (a). To demonstrate the effect of the jaggedness attribute individually, a sorting on the MPG7 collection was carried out as well.

a) Preset: Sorting were carried out with contours obtained from images of the COREL collection, specifically from 30 or more images for each of the 112 basic-level categories (10710 images in total). Images were only 128x192 in size, since this is the typical size used in our studies [11]. The vectors were sorted (retrieved) along individual dimensions (attributes), which is shown row-wise in figure 5. For each row, a 'base' geometry was manually selected and one dimension was increased in five increments. For each increment the most similar (least distant) curve was selected and plotted. The corresponding dimension value of the found vector is given in the upper right of each graph. This search does not guarantee the optimal selection for two reasons: one is the limited amount of curve geometries in a database; the other is the presence of multiple dimensions in the distance measure. To counteract the latter, the weights for the base geometry were set to a low value, whereas the weights for the incrementing dimension was set high (vice versa for the radialbasis function). Still, for some selections an actual increment cannot be found, see for instance the second row, in which two times the same segment appears.

In the top row, an arc was selected as base geometry (e.g. a=1, w=0) and the bendness dimension was systematically increased from 0 to 1 (increment of 0.2). The bendness value of the most similar curve is noted in the upper right and corresponds to the unnormalized value, ranging from 0.15 to 0.93 in this case. In the second row, an attempt was made to select arcs with a high angularity (e=1), that is selections that are most similar to an L feature or an (open) 3-segment polygon (forming an arc). In the 3rd row, the irregularity (symmetry) value for an arc was changed, which selected 'distorted' arcs initially and more even arcs for higher values. In the 4th row, the bendness value for an alternating curve is increased, resulting in selections that range from straight line to an undulating curve. The last curve is an example of insufficient partitioning (undetected semi-turn) due to the use of subsampled window sizes. In the 5th row, the bendness value for an alternating curve with high angularity (e=1) was increased, moving so toward a zigzag line. In the 6th row, the alternating value itself was increased. In the 7th and 8th row, the symmetry value and the circularity was increased.

See also supplementary material for an alternative way of visualizing the continuity of the multi-dimensional space.



Fig. 6. Jaggedness values (sorted) for the MPG7 shapes. Ca. 12 shapes show a high degree of jaggedness, then the values drop sharply.

b) Jaggedness: The effect of the jaggedness attribute is specifically demonstrated by sorting all 1400 MPG7 shapes according to their degree of jaggedness. Figure 6 displays the sorted distribution of the jaggedness values, whereby only one jaggedness value for the unpartitioned, closed shape curve is determined - the value is thus to be interpreted as a global measure in this case. A few shapes show a high degree of jaggedness, the majority has a value smaller than 0.35. The shapes with the largest and smallest values are shown in figure 7, in the top three and bottom three rows respectively.



Fig. 7. Most/least jagged shapes (taken from sorting in figure 6). Most jagged shapes in top 3 rows - grayscale value of curve is (approximatively) proportional to jaggedness value; least jagged shapes in bottom 3 rows.

V. SHAPE RETRIEVAL

Shape matching systems for arbitrary shape retrieval use either point-based and/or segment-based matching methods. By point-based is meant the exhaustive point-to-point correspondence matching. Point-based matching is the prevailing matching principle and can be exploited for either aligning and/or for discriminating between two shapes, e.g. [18], [4], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Although powerful descriptions exist meanwhile for this type of matching, there are a number of short-comings associated with it:

1) Lack or limited robustness to fragmentation: the systems often require that the shape is a closed curve and they can therefore not be easily applied to gray-scale images, where shapes appear mostly fragmented due to 'noise'.

2) Lack of part interpretability: the systems often cannot identify shape parts and their relations, which however would be essential for manipulating the shapes (e.g. a robot trying to understand shapes for interacting with them).

3) Long matching duration: point-to-point measurements are inherently time-consuming; their algorithmic complexity is square $O(N^2)$, and if a closed curve is tested, it is of cubic complexity $O(N^3)$ (N=number points).

There are attempts to alleviate those problems. For instance, Ghosh and Petkov [29] have pleaded for testing shape recognition systems with fragmented shapes and presented the incomplete-contour representation (ICR) test; their own solution appears to be robust, but remains a point-based recognition system. Schmidt et al. [30] have developed an optimized point-based matching procedure, specifically a faster minimization method to find the proper global alignment of two shapes. Or one can use feature points (landmarks) to reduce the number of alignment matchings. Still, those attempts cannot truly shake off the above limitations.

Segment-based matching systems extract boundary segments, which are individually matched amongst two shapes (or between a shape instance and a category representation in case of classification). This is in principle a step toward escaping the shortcomings of point-based matching, but many segmentbased matching systems remain footed in the former. For instance, Latecki and Lakamper introduced a curve evolution which provides a good degree of part interpretability; their segment similarity measure is based on relating curvepoints [10]. Felzenszwalb and Schwartz's system is robust to fragmentation and was shown to work on contours obtained from gray-scale images [5]; their curve comparison also remains a point-to-point matching. Daliri and Torre's shape matching system uses point-to-point matching only for aligning two shapes, after which then the shape is divided into segments of equal lengths, which in turn are transformed into a symbolic representation by classifying the segments into 32 types made of four discretized radii and eight angles [24]. McNeill and Vijayakumar's system intentionally refrains from feature extraction and thus is not suitable for part interpretation [6].

Some segment-based systems are devoid of any point matching. The recognition system by Attalla and Siy uses a polygonal description, specifically arcs of equal length [31]. For each arc a small number of attributes is determined: the arc's chord length, its degree of curvature and its distance from the shape center. The latter attribute represents a radial description of the shape. The system however also relies on the shape being closed. And because segments are of equal length, a part interpretation is not possible and the shape description lies in some sense between point- and segment-based matching. The complexity of the system is O(N) only and thus clearly less complex than any point matching system; it is possibly the most efficient shape recognition system with respect to the speed/accuracy tradeoff.

Some of the mentioned systems are summarized in table I. The 1st column ('Perf.') denotes the retrieval score for the MPG7 shape database (set B with 1400 silhouettes, 20 instances per 70 classes; [32]). The 2nd column denotes the (principal) complexity and is sometimes estimated by us, as not all studies report an explicit algorithmic estimation. The 5th column ('Part Int') uses a simple rating to denote whether a system permits part interpretation of the analysed shape. The 6th column, represents a binary robustness 'rating', whereby a plus sign indicates when a system has been shown to perform also on shapes in gray-scale images. Clearly, all the systems have their advantages and disadvantages.

a) Distance Optimization.: Recently, efforts went into postprocessing (optimizing) the distance matrix (that is generated for pairwise retrieval). The postprocessing is a type of unsupervised learning, that seeks to decrease the distances between similar shapes - and thus within-class distances -, and to increase the distance between dissimilar classes - and thus extra-class distances. This learning comes in different forms,

e.g. [33], [34], [35], [36]. There exist two notable optimization algorithms. One is the page-rank related algorithm by Bai et al. [35], which improved Ling and Jacobs' popular innerdistance description [23] by 6.21 percent. Bai et al.'s method is simple to implement as it essentially consists of a single recursive equation, and it appears to be applicable to different tasks [35]. The other optimization algorithm is based on a modified mutual graph method, by Kontschieder et al. [34]. Their method is more complex (and thus we employed their software package²), but improved the inner-distance description by 8 percent. It appears more efficient for the task of distance matrix optimization and reports also much shorter optimization duration. We later distinguish between *geometric* and optimized distance matrix or retrieval performance, with the former as obtained without any learning (optimization), and the latter with learning (see also the horizontal division in table I).

We also evaluate our curve description methodology on the MPG7 database using analogous types of matching as the ones mentioned above. One is a space matching (subsection V-A), which would correspond to point-to-point correspondence matching. Another one is segment matching, which exploits the segment abstractions with parameters as developed in the previous section, whereby a simple radial description is employed to relate segments (subsection V-B). Finally, we also combine the analysis of the two methods, which gives us the best retrieval results on the MPG7 collection (section V-C). For each method we also apply an optimization of the distance matrix; thus in total we provide 6 entries to table I (in bold face).

A. Space Matching

The type of point-based matching that we carried out, is most similar to the matching system by Adamek and O'Connor and also analogous to curvature-scale space matching [22], [20]. Adamek and O'Connor create a fine/coarse scale space from which they derive convexity/concavity (later collectively called bulginess) by determining the spatial relation between low-pass filtered curves; they determine shape similarity by matching the signed inter-curve distance spaces and by exploiting dynamic programming to find the minimum distance. The method presented here matches the amplitude scale space. To avoid cubic matching complexity we reduce the exhaustive alignment search by using keypoints. The keypoints are obtained by partitioning the curve as discussed (section III) and use the midpoints of the partitioned segments as keypoints as those correspond to high- or low-curvature points. The segments are not used otherwise in this matching method.

1) Implementation: For each silhouette, 100 equally spaced points were selected and the bowness space **B** was generated with 10 window sizes. For matching, the space was regarded as a 1000-dimensional vector; the difference between two shapes *i* and *j* was implemented with the Manhattan distance: $d_{ij}^{\mathbf{B}} = \sum_{\omega, v} |\mathbf{B}^i - \mathbf{B}^j|$. A partitioned shape offered in average 20 keypoints (that is segments, see also section V-B). Due to various types of variability (structural, aliasing), the keypoints

TABLE I

RETRIEVAL SYSTEMS, THEIR BULL'S EYE SCORE (FOR THE MPG7 COLLECTION) AND THEIR CHARACTERISTICS. TYPE: GEOMETRIC ONLY (NO OPTIMIZATION); OPTIMIZED: DISTANCE MATRIX TUNED BY LEARNING ALGORITHM. UNIT: PT=POINT, SEG=SEGMENT. PRINCIPAL:LGS=LOCAL/GLOBAL SPACE (THIS STUDY), ID=INNER DISTANCE (INCL. SHAPE CONTEXT). PART INTERPRETABILITY. ROBUSTNESS: GRAY-SCALE APPLICABILITY.

Т	Perf.	Complexity	Unit	Principal	Part Int	R	Authors
Optimized	95.96%	$O(N^3)$	pt	ID, Aspect	+		Ling etal,10
	95.01%	$O(N^3)$	pt/seg	LGS, segment vectors	+++		combined [proposed]
	93.40%	$O(N^3)$	pt	ID, modified mutual graph	+		Kontschieder etal,09
	93.32%	$O(N^3)$	pt	ID, LCDP, uns GP	+		Yang etal,09
	91.61%	$O(N^3)$	pt	ID			Bai etal,10
	91.03%	$O(N^3)$	pt	LGS	+++		space matching
	85.08%	O(N)	seg	segment vectors	+++	+	segment matching
Geometric Only	89.31%	$O(N^3)$	pt	contour flexibility			Xu etal,09
	87.70%	$O(N^3)$	pt/seg	hierarchical multi-resolution		+	Felzenszwalb&Schwartz,07
	85.40%	$O(N^3)$	pt	ID (inner distance)	+		Ling&Jacobs,07
	84.93%	$O(N^3)$	pt	multi-scale convexity/concavity			Adamek&OConnor,04
	84.80%	$O(N^3)$	pt/seg	LGS, segment vectors	+++		combined [proposed]
	84.33%	O(N)	pt-seg	equal-length segments; radial			Attalla&Siy,05
	81.12%	$O(N^3)$	pt	curvature scale space	++		Mokhtarian&Bober,03
	78.74%	$O(N^3)$	pt	LGS	+++		space matching
	78.38%	$O(N^3)$	pt	distance sets		+	Grigorescu&Petkov,03
	76.45%	$O(N)/O(N^2)$	seg/pt	parts	+		Latecki&Lakamper,00
	70.48%	O(N)	seg	segment vectors	+++	+	segment matching

are not at exactly the same locations (for shape instances of one class), causing a decrease in performance in comparison to exhaustive point-based matching. To compensate for this variability, the neighboring points around the keypoints were used for alignment as well: by dilating 2 pixels, the average number of keypoints increased to 30 (some segments overlap due to the use of selected window lengths), which returned a reasonable approximation to an (estimated) full matching. The algorithmic complexity is thus less than cubic, $O(< N^3)$. The resulting distance matrix between two shapes is of size $m \times n$, with m and n being the number of keypoints for each shape. A dynamic programming approach did not make sense and we simply took the minimum value of the entire distance matrix. Matching duration with 30 keypoints (along v) was ca. 80 milliseconds for a pair of shapes on a 2.66 GHz Intel Pentium.

2) Evaluation: For each shape, the remaining 1399 shapes were ordered according to their increasing distance. Retrieval accuracy was measured with the Bull's eye score, which counts the number of same-class instances within the first 40 most similar shapes (including self-similarity). This count is determined for all 1400 shape retrievals, summed and divided by the maximal possible count, namely 28000 (1400*20; [32]).

The geometric Bull's eye score is 78.74 percent, see 'space matching' in table 1. Increasing the resolution to 200 points for instance, only marginally increased the performance by ca. 1-2 percent.

The optimized score with the modified-mutual graph method is 91.03 percent (a gain of 12.29), with kNN=12 for local scaling normalization, kNN=6 and c=3.5 for the graph. The class-individual gain can be up to 50 percent; it is slightly negative for some classes and larger for another (figure 8, upper right).

The optimized score with the page-ranking related algorithm is 88.36 percent (a gain of 9.62), with kNN=15 and α =0.4 for local scaling normalization. As results with this method were consistently lower by a few percent than the modified-mutual graph method - as can be assumed from the comparison of the two methods on the inner-distance method - they are no longer reported here and only results with the modified-mutual graph method are given from now on.



Fig. 8. Optimization Analysis. Upper left: Precision/Recall curves for geometric and optimized combination of space and segment matching; ID-SC: inner-distance (& shape context) by [23]; ASC+LCDP by [36]. Class-individual gain for space matching (upper right), for segment matching (lower right) and the difference in gain between space and segment matching (lower left).

3) Discussion: The geometric retrieval accuracy lies in the range of other respectable shape recognition systems, but lies significantly under Adamek and O'Connor's system performance (84.93%), probably because their system elegantly exploits concavity and convexity information [22], which we were not able to exploit.

The increase in resolution (to gain higher performance) is not worth the square increase in matching duration. The observation that 100 points provide a sufficient recognition performance was also observed in Attalla and Siy's study [31].

B. Segment Matching

The shape curve was partitioned and abstracted exactly as discussed in sections III and IV. The partitioned segments are related to each other using a simple radial description. The pole was determined by taking the average of all shape pixel coordinates. The segments were related by three parameters: the (normalized) radius r and two types of 'orientation' angles. The two angles correspond to curved segments (arcs) and straight segments: for curved segments we determine the degree of bulginess, which represents convexity and concavity; for straight segments, we determine a tangentiality value.



Fig. 9. Two angles for a radial description ($p_c=pole$). **a.** Tangentiality angle for straight segments: segments 1 and 3 have the same tangentiality and are maximal (π); segment 2 shows minimal tangentiality. **b.** Bulginess (directional) angle for curved (bent) segments; segment 1 is maximally concave; segment 2 shows no bulginess - it is neither convex nor concave; segment 3 is maximally convex.

• Tangentiality (θ): is the (smaller) angle of the intersection of the straight segment and the radial line connecting the shape center (p_c) and the segment's midpoint (gray dashed in figure 9a). Maximal tangentiality is therefore a right angle (segments no. 1 and 3 in figure 9); minimal tangentiality (θ =0) occurs if the segment's chord aligns (parallel) with the radial line (segment no. 2). A segment is considered straight if its bendness value is below a tolerance T_b^{θ} : $\exists \theta_i \in [0, \frac{\pi}{2}] \mathbf{a}_i(b) < T_b^{\theta}$.

• **Bulginess** (δ): is the angular difference between two direction angles (both $\in [0, 2\pi]$). One directional angle is the face angle ϕ obtained from the ray pointing from a segment's midpoint to its own halfpoint (the midpoint of a segment's chord) - the dotted arrow pointing 'north' from a segment's midpoint in figure 9b. The other directional angle is the ray pointing from the pole to the segment's midpoint (dashed in figure). The angular difference lies in the interval $[0, \pi]$; convex segments have a bulginess value in the interval $[0, \frac{\pi}{2})$, concave segments a value in the interval $(\frac{\pi}{2}, \pi]$, a value $\delta = \frac{\pi}{2}$ means neither convex nor or concave (segment 2 in figure 9b). Maximal convexity and concavity are the interval's endpoints (0 and π , respectively, segments 1 and 3 in figure 9b). The

bulginess angle is determined if the segment was detected as an arc $(\Phi^{\beta}(l_{c}) > 0)$: $\exists \delta_{i} \in [0, \pi] \Phi^{\beta}(l_{c}) > 0$.

Elongated arcs were included as well and in order to characterize those segments a bit more specific, a attribute p was added that corresponded to the ratio of amplitude and chord length. Together with the above three components, the vector is now: $\mathbf{c}(\theta, l, a, b, e, \zeta, t, p, r, \delta)$, that is including the transition attributes, but omitting the jaggedness and alternating attributes as they did not contribute to an improved retrieval performance.



Fig. 10. Retrieval score (for segment matching) for varying factors with which the variance for individual attributes was multiplied. Attribute transition (7 trs) appears to have the least impact as it steadily increases.

1) Implementation and Evaluation:

a) List Matching: To match the list vectors of two shapes, the segment vectors are pairwise matched to form a similarity matrix from which the segments' best matches are selected and integrated. The similarity measure is called *congruence* measure in this study, as no scale (size) independence is included (the MPG7 collection shows little if any intra-class size variations).

Given two lists of segments, \mathbf{a}_i and \mathbf{a}_j from shapes A and B, the pair-wise (metric) similarities $\sin(\mathbf{a}, \mathbf{a}')$ of the individual vectors are taken, resulting in a $[n \times m]$ similarity matrix, $\mathbf{S} = [\sin_{ij}]$, with n and m being the list lengths. The similarity measure sim can be further refined by a weight vector $\mathbf{v}(i)$, whose components correspond to the significance of the attributes. Next, the maximum with respect to each shape is taken, returning a congruence vector \mathbf{g} ,

$$\mathbf{g}_A(i) = \max \mathbf{S}[i, j],\tag{22}$$

$$\mathbf{g}_B(j) = \max \mathbf{S}[i, j]. \tag{23}$$

A weighted sum of the individual components is taken, with the weights \mathbf{w} corresponding to the segments' significances and acting as a normalizer. The final congruence value is a multiplication of the weighted sum of both shapes:

$$\operatorname{cong} = \mathbf{g}_A' \mathbf{w}_A \times \mathbf{g}_B' \mathbf{w}_B. \tag{24}$$

The complete congruence measure is summarized as $cong_{kl}(\mathbf{a}_i, \mathbf{a}_j, \mathbf{v}, \mathbf{w}_A, \mathbf{w}_B)$, with **v** as the attribute weight vector

with dimensionality equal the number of attributes (parameters), \mathbf{w} as the segment weights with dimensionality corresponding to the list lengths (*n* and *m*), and *k* and *l* two shapes.

b) Complexity: An informal complexity estimate is provided. The most time consuming part is the generation of the local/global (scale) space for partitioning and abstraction, which is analogously complex as creating the curvature scale space, thus $O(N \log N)$.

As the partitioned segments are merely a list of parameters and selected coordinates (end- and midpoints), the segmentmatching duration is 2.8 milliseconds only and thus negligible as compared to the duration of generating the local/global space (next paragraph).

2) Implementation: No subsampling occurred, meaning the full size of the silhouette was employed. T_b^{θ} was set to 0.4. The similarity metric for two vectors (sim) is determined using a Gaussian radial-basis function, whose variance for the individual dimension is set to their overall variance of the entire collection (vector **v**). As mentioned previously, all three steps (LG space, partitioning and abstraction) take ca. 1300ms on a 2.66 GHz Intel Pentium.

3) Evaluation: The geometric retrieval score is at ca. 67 percent for unitary attribute weights **v**. A simple heuristic search was employed to optimize the weights, which increased the performance to 70.48 percent (see figure 10). The search was carried out by multiplying the (average) variance of the radial-basis function with varying factor values (x-axis in figure 10). The transition attribute appeared to have the least effect on the retrieval score, as it monotonically increased with factor size. Matching duration was ca. 20 times shorter than the one for space matching.

Exploiting distance optimization (modified mutual graph), the score rises to 85.08 percent, a gain of 14.60 percent (with kNN=20 for local scaling normalization, kNN=6 and c=2.5 for the graph). The class-individual gain in Bull's eye score can be more than 50 percent, and is negative for only one class (figure 8, lower right). To analyze more specifically the differences in class-individual gain for the two matching methods, we subtract the two sets of gain values (pairwise) and observe that there can be significant differences for some classes (plot in lower left in figure 8).

4) Discussion: The geometric retrieval score is rather low, but this type of matching is also the first of its kind: only abstracted (parameterized) segments are used and no particular dependence on the closeness of the shape exists. A possible reason for the relatively low performance may be that a radial alignment is structurally too simple for shape description. Further improvement may be obtained by grouping segments. The class-individual gain analysis in figure 8 shows that the two matching methods favor different classes and that encourages us to combine the matching methods.

C. Combined Matching

The combination of the two methods comes at little additional temporal cost, since the processes of segment partitioning, abstraction and matching occur relatively rapid. And it comes at little increase in storage size, with 22 segments in average per shape and 10 attributes for a vector \mathbf{c} , which increases the dimensionality to 1220 (1000 from the space).

1) Implementation and Evaluation: The combination occurs by adding the two distance measures for each method for a pair of shapes $(d_{comb} = d_{ij}^{\mathbf{B}} + \operatorname{cong}_{ij})$. The geometric score is 84.80 percent and is thus in the range of other top-performing systems (see 'combined' in table I). The precision-recall curve for this retrieval is shown in the upper left of figure 8: it is practically equal to the one for the inner distance description (IDSC).

Applying the distance optimization method (modified mutual graph), the score rises to 93.71 percent. We also tested a combination after individual optimization, that is the optimized distance matrices for each method are added ($d_{comb} = d_{ij}^{\mathbf{B},opt} + cong_{ij}^{opt}$), then the score is marginally higher with 93.89 percent. Applying the optimization algorithm again to that combined matrix, we obtain 95.01 percent (gain of only 1.12 percent), which is the second best reported so far. The precision-recall curve for that latter combination is shown in figure 8 (upper left) and is practically equal with the one of the optimized aspect space description (dotted) [36].

This last combination thus consists of three optimizations in total, which however given the temporal efficiency of the modified mutual graph algorithm are negligible. The downside of this combination is rather that it lacks the robustness to gray-scale images, but its advantage of part interpretability remains.

2) Discussion: The combined retrieval performance (84.80%) is only 4.5% below the present benchmark of geometric scores (89.31% by [26]). The score with distance optimization (95.01%) is the second best reported so far to our knowledge (95.96% by [36]).

D. Discussion

Recent studies emphasize in particular the need for invariance to articulation (part alignment variability), e.g. [26], [36], [27]. In those studies, the problem was formulated as a tradeoff or balance between an increased articulation invariance and a decrease in class discriminability. In our description, this issue of balance was not particularly addressed. Instead, the presented description is a rather detailed description, in which articulation is expressed by a segment characterization - it is parameterized essentially. Thus, the potential problem that arises with such a detailed description is that it may be too detailed and lead to intra-class clusters, which deteriorates discrimination. Further analysis may give more clues about the exact nature of the representation, but presently the increase in discriminability appears to be elegantly provided by the distance optimization methods.

The distance optimization with the modified mutual graph method yielded higher gains in our work (12.29, 14.60 and ca. 10.0 percent for space, segment and combined matching) as opposed to the 8 percent increase in Kontschieder et al's study [34]. There are two possible reasons for this higher gain:

1) Starting with a lower geometric score - as in case of space and segment at least - allows more room for improvement, as opposed to a geometric score that is high already. 2) The local/global description bears a better separation between classes. As Ling et al. already assume, the crucial issue is that the geometric distance measure provides an overall good separation between perceptually dissimilar classes - even if the geometric score is only mediocre in comparison to some other systems (see their section 4 in [36]).

VI. IMAGE CLASSIFICATION AND RETRIEVAL

Systems for image classification are typically evaluated on either a database with fixed number of categories, e.g.[37], [38], [39], or in classification and retrieval competitions where the number of categories grows each year, e.g.[40]. We applied our method to both types of databases.

Of the fixed-categories collections, there are three notable ones, the Urban &Natural collection [38] and the Caltech101 [37]. The Urban&Natural collection contains 8 super-ordinate categories (mountain scene, forest scene, street scene, highway scene,...) and were classified with 80 percent correct using a modified Fourier transform as preprocessing [38]. The Caltech101 collection contains mostly objects in close-up view (football, ying-yang sign, accordeon,...), some are embedded in a scene (e.g. cheetahs, anchors); they were correctly classified with roughly 70 percent by different methods (see figure 6 in [41] for a summary). The Landuse collection consist of satellite images depicting 21 categories [39] (street intersections, forest, agricultural fields,...) and were correctly classified with ca. 81 percent by a bag-of-features approach, with features being SIFT features [39].

Of the classification and retrieval competitions, we participated in particular in the image retrieval task, where the dominant method is the extraction and matching of gradient histograms of fixed-size intensity patches, such as the widely used SIFT features and their modification [42].

Using our preprocessing, we applied two types of classification so far: a simple statistical one using only image vectors (subsection VI-A); and an ensemble classifier exploiting the multi-dimensional space of the individual descriptor vectors (subsection VI-B).

A. Image Vector Classification

1) Implementation: Contours were extracted with the Canny algorithm and then processed exactly the same way as was done for shape retrieval, that is LG space generation, partitioning and abstraction were not modified or adjusted at all. Contours were extracted from four different spatial scales, which returned an abundance of contours and we therefore discarded strongly overlapping segments. An average image contained several hundred segments. For each partitioned contour we extracted a number of appearance parameters (contrast, fuzziness,...), which were based on simple luminance statistics, see [11] for details.

Our reported, best-performing classification results also depended on other *descriptor types*, which describe for instance grouping, region or texture information, but describing these descriptors here would exceed the scope of the present study.

For a list of descriptors $\mathbf{a}_i(d)$ we generated a 10-bin histogram H_d for each attribute d ($d = 1, ..., n_{\text{Dim}}$, number of dimensions). The attribute histograms were then concatenated to form a high-dimensional image vector H^{I} , whose length could be several hundred components. Thus, there is no use of the multi-dimensionality of the vector space per se; the histogram is a mere statistical description of the curve attributes present in an image. The principal component analysis (PCA) was used to optimize the separability between categories. A linear discriminant analysis (LDA) worked best on the image vectors.

2) Evaluation: With a 6-fold cross validation we reached 76 percent correct classification for the Urban&Natural collection, 77 percent for the Landuse collection and 40 percent for the Caltech101 collection. In the ImageClef competition, we reached the middle of the ranking of all submitted approaches, in one task we reached the top third [13].

The significance of individual attributes was estimated by knocking out individual attributes in the statistical classification task. For an attribute knock-out, the corresponding 10 bins were eliminated. To provide a meaningful reference, a classification performance with only the segment descriptor was determined as well (horizontal, solid line figure 11; dashed lines are standard error for 6 folds), which is lower than the best-performing results just mentioned. A knockout performance below or above that reference value stands for a more or less significant attribute. The significance for individual attributes varies between image collections. For instance, the circularity attribute plays a more important role in the Caltech and the Urban&Natural collection (performance lower) than in the Corel collection (performance higher).

3) Discussion: Classification with mere image vectors was relatively competitive: the benchmark for the Urban&Natural collection was reached already. The knock-out experiments show that the significance can vary substantially between collections and one could attempt to employ a feature selection procedure.

B. Ensemble Classification

In an ensemble classifier the decision is based on several weak classifiers - as opposed to a single 'strong' classifier as in the LDA for example. To learn the weak classifiers we apply the adaptive boosting method, which specifically concentrates on the misclassified examples in the training set and learns associated weights in a systematic way [43], [44]. Viola and Jones introduced this methodology to the computer vision community with their rapid face detection system [45]. They specifically used single-node decision trees (decision stumps) as weak classifiers, but applying such decision stumps to individual dimensions did not yield good performance in our case. Instead we used a pooled decision of the decision stumps for individual dimensions. We firstly explain the classifier we built and then the boosting (learning) procedure we used.

a) Classifier: Given a single entry v of a descriptor vector **a** (from list \mathbf{a}_i), the decision stump evaluates whether the value lies on the correct side of a certain threshold θ

$$t(v, p, \theta) = \begin{cases} 1 & , v > p\theta \\ 0 & , \text{ otherwise,} \end{cases}$$
(25)



Fig. 11. Attribute significance estimated with dimension knock-out for a statistical classification task (top to bottom: Caltech101, Corel, Urban&Natural and MPG7 collection). bnd=b; eds=e.

whereby p stands for the polarity of the inequality. θ and p are adjusted during learning by seeking the optimal discrimination between one category k and all others. We then integrate the stump outcomes across dimensions to arrive at a descriptor activation value h,

$$h(\mathbf{a}, \mathbf{p}, \mathbf{\Theta}) = \sum_{d=1}^{n_{\text{Dim}}} t(\mathbf{a}(d), \mathbf{p}(d), \mathbf{\Theta}(d)), \quad (26)$$

where $\Theta(d)$ can be regarded as an hyperplane separating one category from all others, $\mathbf{p}(d)$ is the array of polarities. For a category, a number of hyperplanes n_L is determined and their activation values integrated and weighted to form a descriptor confidence value c

$$c(\mathbf{a}, \mathbf{p}_l, \mathbf{\Theta}_l) = \sum_{l=1}^{n_L} h(\mathbf{a}, \mathbf{p}_l, \mathbf{\Theta}_l) \mathbf{w}(l), \qquad (27)$$

whereby weights **w** are normalized to one, $\sum_{l} \mathbf{w}(l) = 1$. This integration corresponds to the so-called strong classifier in the terminology of ensemble classifiers. The posterior for a category is calculated as the mean confidence value of all n_{Desc} descriptors of an image,

$$P = \sum_{i=1}^{\mathsf{n}_{\text{Desc}}} c(\mathbf{a}_i, \mathbf{p}_l, \mathbf{\Theta}_l) / \mathsf{n}_{\text{Desc}}.$$
 (28)

We used two types of classification. One is used during learning, which occurs by taking the maximum value across the posterior values $\operatorname{argmax}_k P_k$. Another one is used during classification of the validation set: we found that using the posterior values for each category and descriptor type (segment, pair, texture,...) as input to the LDA yielded substantially for classification was the number of descriptor types times the number of categories $V = \{P_1^s, P_2^s, ..., P_{n_K}^s, P_1^p, P_2^p, ..., P_{n_K}^p, ...\}$ (s=segment, p=pair, etc.). This means that how an image responded to other category representations was valuable information to obtain better discrimination.

b) Adaptive Boosting: In the adaptive boosting learning procedure, a sample is reused in a weighted manner according to the evolving classification performance during training. In our case, we adjust a descriptor weight s(i) (significance) after each learning round. The weight is included when we determine the optimal threshold θ and polarity p: t takes the value of s(i) instead of only 1 in equation 25 - if the value lies on the right side of the inequality. Once an optimal threshold and polarity is found, the threshold values t are integrated across dimensions and descriptors (for a category) and that determines weight $\mathbf{w}(l)$. After each learning step l, the descriptor weights s(i) of misclassified images are adjusted by increasing their value by a small amount.

1) Evaluation: With ca. 10 learning steps (hyperplanes, n_L), we obtained a classification performance of ca. 82 percent for the Urban&Natural collection - for fewer or more learning steps the performance decreased. The optimal dimensionality for V consisted of 22 dimensions - after application of the PCA. For the Landuse collection we used ca. 15 learning steps and an optimal dimensionality of ca. 56 dimensions to achieve a correct classification of 79 percent. For the Caltech101 collection we used ca. 20 learning steps and an optimal dimensionality of ca. 150 components to achieve a correct classification of ca. 49 percent.

2) Discussion: The classification benchmark for the Urban&Natural collection was just exceeded; the one for the Landuse almost reached; the one for the Caltech101 collection however was not reached yet. Nevertheless, the potential of our new methodology is evident.

C. Discussion

The use of the image vector resulted in relatively good performance, that is the mere curve statistics of an image contain valuable information; it represents a fast method as learning duration and classification is short. The use of the ensemble classifier yielded better performance, but it also requires time to learn the hyperplanes. We believe that even better performance can be achieved, if we built a classifier exploiting individual descriptor vectors.

VII. GENERAL DISCUSSION AND SUMMARY

The key to homologue (class-consistent) partitioning is a procedure, which detects consistent bows in the amplitude space, allowing so to the robust identification of arcs. Partitioning is not completely homologue at a global level (figure 4), but better partitioning is more likely obtained with a perceptual organization (grouping) of the abstracted segments. Nevertheless, when tested with a radial description, the retrieval score for the MPG7 rate is at ca. 70 percent. If the distance measure is combined with a traditional correspondence-based matching method, the retrieval score becomes comparable to bestperforming systems. In image retrieval we obtained competitive results using only an image vector. In image classification we were able to exceed the benchmark for the Urban&Natural collection with an ensemble classifier. The curve description methodology is thus applicable to any database and while we have not yet outperformed the prevailing SIFT features and its variants [42], our curve methodology in turn offers in principle precise scene interpretation - whereas SIFT features do not. The challenges ahead are proper grouping algorithms and novel learning methodology.

The most time consuming part of the method is the generation of the LG space, whose complexity is O(NLW)approximately, with N the number of segment points, L the number of window levels (dependent on N), W the amount of computation for window labeling. Partitioning and abstraction depend on N and L only and take thus much less time in comparison.

The attributes and their definition can be taken as examples. The attributes jaggedness and alternating can play a role in gray-scale images (figure 11), whereas for shape retrieval in the MPG7 collection they were hardly useful. In another study, we have shown that we also reached the benchmark for the Urban&Natural collection (under review; see also figure 11), thus the methodology can be applied in gray-scale images without modification, that is, it works on fragment contour images equally well.

Whether the stopping criterion is sufficient for classification remains to be further tested. The dilemma is that jaggedness may not always be 'noise', instead there may exist characteristic segments of approximate similar frequency, that need to be extracted for proper recognition; or, the presently chosen threshold may be too low for some other curves, e.g. discriminating a square from a complex animal shape does not require detailed partitioning even if the square is jagged (e.g. contains a dent). Thus, the stopping rule is ideally set using the results of a perceptual organization or by semantic content. In other words, optimal partitioning is probably not possible without the use of semantic content (e.g. the corresponding shape class), very much like the task of image segmentation.

There are other recognition tasks where our methodology could be applied:

a) Identification: For curve identification tasks, such as in [46], [47], the present abstraction may not be sufficient. But the local/global space is a rich description from which more specific characterizations can be dervied. For instance, the parameterization of a signature block could be extended, by determining its degree of 'flatness' or its degree of distortion (asymmetry) to one side. Another possibility is to use a finer subsampling of window sizes. With the output of the arc/straighter detection algorithm (\mathcal{E}^{seg}), one could construct abstractions of (open) polygons, specifically a sequence of three segments. Such abstractions could be in particularly useful to represent symmetric alignments in polygon curves.

b) Grouping for segmentation: The majority of grouping studies uses computationally expensive point-based matchings [48], but the relatively detailed detection and characterization of arc segments with our method would allow to generate a

number of 'arc hypotheses' with which segmentation could be accelerated.

ACKNOWLEDGMENT

The study was partly supported by the Gaze-Based Communication Project (European Commission within the Information Society Technologies, contract no. IST-C-033816).

REFERENCES

- H. Asada and M. Brady, "The curvature primal sketch," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 2–14, 1986.
- [2] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Transactions* on Pattern Analysis and Machine Intelligence, vol. 8, pp. 34–43, 1986.
- [3] A. Bengtsson and J.-O. Eklundh, "Shape representation by multiscale contour approximation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 85–93, 1991.
- [4] T. Sebastian, P. Klein, and B. Kimia, "On aligning curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 116–125, 2003.
- [5] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," *IEEE Conference on Computer Vision and Pattern Recognition*, 17-22 June 2007, Minneapolis, USA, pp. 1–8, 2007.
- [6] G. McNeill and S. Vijayakumar, "Hierarchical procrustes matching for shape retrieval," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1.* Washington, DC, USA: IEEE Computer Society, 2006, pp. 885–894.
- [7] R. Nelson and A. Selinger, "A cubist approach to object recognition," in Sixth International Conference on Computer Vision, 1998.
- [8] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1270–1281, July 2008.
- [9] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, pp. 257–266, March 1988.
- [10] L. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 22, pp. 1185–1190, 2000.
- [11] C. Rasche, "An approach to the parameterization of structure for fast categorization," *International Journal of Computer Vision*, vol. 87, pp. 337–356, 2010.
- [12] B. Ionescu, K. Seyerlehner, C. Rasche, C. Vertan, and P. Lambert, "Content-based video description for automatic video genre categorization," in *The 18th International Conference on MultiMedia Modeling*, 4-6 January, Klagenfurt, Austria, Springer-Verlag LNCS - Lecture Notes in Computer Science, Eds. K. Schoeffmann et al., no. 7131, 2012, pp. 51–62.
- [13] C. Rasche and C. Vertan, "A novel structural-description approach for image retrieval," in CLEF (Notebook Papers/LABs/Workshops), 2010.
- [14] S. Oprisescu, C. Rasche, and S. B., "Automatic static hand gesture recognition using tof cameras," in 20th European Signal Processing Conference, Bucharest, RO, 2012.
- [15] F. Mokhtarian and A. Mackworth, "A theory of multiscale, curvaturebased shape representation for planar curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 789–805, 1992.
- [16] M. Fischler and R. Bolles, "Perceptual organization and the curve partitioning problem," in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, Volume 2*, 1983.
- [17] S. Manay, D. Cremers, B.-W. Hong, A. Yezzi, and S. Soatto, "Integral invariants for shape matching," *IEEE Transactions on Pattern Analysis* and Machine Intelligence, vol. 28, no. 10, pp. 1602–1618, October 2006.
- [18] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002.
- [19] C. Grigorescu and N. Petkov, "Distance sets for shape filters and shape recognition," *Image Processing, IEEE Transactions on*, vol. 12, pp. 1274 – 1286, 2003.
- [20] F. Mokhtarian and M. Bober, Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization. Norwell, MA, USA: Kluwer Academic Publishers, 2003.
- [21] A. Jalba, M. Wilkinson, and J. Roerdink, "Shape representation and recognition through morphological curvature scale space," *IEEE Transactions on Image Processing*, vol. 15, pp. 331–341, 2006.

- [22] T. Adamek and N. O'Connor, "A multiscale representation method for nonrigid shapes with a single closed contour," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 742–753, 2004.
- [23] H. Ling and D. W. Jacobs, "Shape classification using the innerdistance," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 29, pp. 286–299, 2007.
- [24] M. R. Daliri and V. Torre, "Robust symbolic representation for shape recognition and retrieval," *Pattern Recogn.*, vol. 41, pp. 1782– 1798, May 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id= 1340786.1343133
- [25] J. Xie, P.-A. Heng, and M. Shah, "Shape matching and modeling using skeletal context," *Pattern Recogn.*, vol. 41, pp. 1773–1784, May 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1340786.1340846
- [26] C. Xu, J. Liu, and X. Tang, "2d shape matching by contour flexibility," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 180–186, January 2009.
- [27] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-invariant representation of non-planar shapes," in *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 286–299. [Online]. Available: http://dl.acm.org/citation.cfm? id=1927006.1927030
- [28] K. A. Raftopoulos and S. D. Kollias, "The global-local transformation for noise resistant shape representation," *Comput. Vis. Image Underst.*, vol. 115, pp. 1170–1186, August 2011. [Online]. Available: http: //dx.doi.org/10.1016/j.cviu.2011.03.009
- [29] A. Ghosh and N. Petkov, "Robustness of shape descriptors to incomplete contour representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1793–1804, November 2005. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2005.225
- [30] F. Schmidt, D. Farin, and D. Cremers, "Fast matching of planar shapes in sub-cubic runtime," in *IEEE 11th International Conference on Computer Vision*, 2007.
- [31] E. Attalla and P. Siy, "Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching," *Pattern Recogn.*, vol. 38, pp. 2229–2241, December 2005. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2005.02.009
- [32] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for nonrigid shapes with a single closed contour," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 1, 2000, pp. 424 – 429.
- [33] X. Yang, S. Kknar-tezel, and L. J. Latecki, "L.: Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR*, 2009.
- [34] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *Computer Vision - ACCV 2009*, ser. Lecture Notes in Computer Science, H. Zha, R.-i. Taniguchi, and S. Maybank, Eds. Springer Berlin / Heidelberg, 2010, vol. 5996, pp. 655–666.
- [35] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context-sensitive shape similarity by graph transduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 861–874, May 2010. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2009.85
- [36] H. Ling, X. Yang, and L. J. Latecki, "Balancing deformability and discriminability for shape matching," in *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 411–424. [Online]. Available: http://dl.acm.org/citation.cfm? id=1927006.1927039
- [37] F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE IN-TELLIGENCE*, vol. 28, no. 4, pp. 594–611, 2006.
- [38] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
- [39] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2010, pp. 270–279.
- [40] H. Mèuller, P. Clough, T. Deselaers, and B. Caputo, *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*. Springer, 2010, vol. 32.
- [41] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian processes for object categorization," *INTERNATIONAL JOURNAL OF COMPUT-ER VISION*, vol. 88, pp. 169–188, 2010.

- [42] D. Lowe, "Distinctive image features from scale-invariant keypoints," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 60, no. 2, pp. 91–110, 2004.
- [43] R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.
- [44] L. Breiman, "Arcing classifier (with discussion and a rejoinder by the author)," *The annals of statistics*, vol. 26, no. 3, pp. 801–849, 1998.
- [45] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.
- [46] S. W. Parent, P. & Zucker, "Trace inference, curvature consistency, and curve detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, pp. 823–839, 1989.
- [47] S. Wang, J. Stahl, A. Bailey, and M. Dropps, "Global detection of salient convex boundaries," *International Journal Of Computer Vision*, vol. 71, pp. 337–359, 2007.
- [48] L. Williams and K. Thornber, "A comparison of measures for detecting natural shapes in cluttered backgrounds," *International Journal of Computer Vision*, vol. 34, no. 2-3, pp. 81–96, 2000.