Grouping and Description of Partitioned Segments

Affiliation and mailing address:
Christoph Rasche
Laboratorul de Analiza si Prelucrarea Imaginilor
Universitatea Politehnica din Bucuresti
Bucuresti 061071, RO
email: rasche15@gmail.com

Abstract

A methodology for the detection and geometric characterization of groups of segments is introduced. One set of groups focuses on a precise geometric characterization of the alignment of two and four segments; and on a geometric characterization of shapes up to five corners, whose outlines are obtained from isocontours. Another set of groups focuses on a loose geometric characterization of three or more segments. The grouping processes occur relatively fast as only keypoints are used, such as the segments end- and midpoints. The grouping output is tested in an image classification task, evaluated on three image collections (Urban&Natural, Landuse and Caltech 101), whereby a structural as well as a statistical form of representation is tested. The classification accuracy is comparable to other approaches.

Keywords: perceptual organization, feature representation, structural, image classification

1 Introduction

Perceptual organization was once believed to be a keystep in a systematic scene reconstruction, which would lead to a semantic scene interpretation (Lowe, 1985; Witkin and Tenenbaum, 1983; Sarkar and Boyer, 1993; Mohan and Nevatia, 1992). But the approach has lost its momentum for two reasons: one is that it has become clear that for the 'mere' purpose of classification for instance, an accurate scene reconstruction is not necessary, e.g. not all structural relations are necessary to 'just' classify image content, e.g. (Oliva and Torralba, 2001; Renninger and Malik, 2004); a second reason is that the input to perceptual grouping, namely appropriately partitioned contour segments, was never properly developed. The latter was addressed in our previous study on curve partitioning. In this study, we develop grouping procedures, that group two and more segments and we propose descriptors for their representation. We apply the methodology to the task of image classification and demonstrate that it almost performs as well as other methods.

The following methodology is based on a rather exhaustive geometric parameterization of the segments' alignments. This may raise some skeptisism as it seemingly lacks a compact formulation, but for the goal of complete image understanding such a description is desired in any case. As we intend to exploit the parameters for the task of semantic image classification, one may further wonder if the diversity of parameters is indeed necessary for the mere purpose of semantic classification. We believe, that this issue is underestimated by recent classification approaches: even the task of semantic classification may require a more specific preprocessing than for instance the mere histogramming of intensity gradients; yet it may not require the 'perfect' perceptual organization as some of the early approaches of scene classification sought. In some sense, we present an intermediate form, a perceptual organization that is just sufficient for classification, but for detailed image understanding it required further analysis, for which however our methodology is an ideal starting point.

1.1 Previous Grouping

Following Sarkar and Boyer's proposal for a classificatory structure, one can distinguish between four levels of groupings for 2D organization (Sarkar and Boyer, 1993): the signal, the primitive, the structural and the assembly level. The signal level organizes pixels or interest points and can aid texture analysis for instance; the primitive level groups edgels into contours and leads to the segmentation of contour and/or region boundaries; the structural level organizes contour segments into groups such as corners, polygons, closed regions, etc; the assembly level identifies arrangements of the structural-level groups. As our study focuses on the structural level only, we review only studies of that level, namely 2D groupings for segments.

Lowe had developed a set of groupings such as vertices (intersecting segments) and clusters of parallel segments and exploited them as pointers to determine the exact pose of an 3D object (Lowe, 1985). Mohan and Nevatia designed a system (Mohan and Nevatia, 1992), called CANC2, "to handle scenes of unknown curved objects imaged from arbitrary viewpoints" (p. 616); it is assumed "that the scene is composed of opaque surfaces" (p. 628). The features they extract is a "hierarchy of collated features [...] suitable for handling objects whose projected surfaces can be described by combinations of ribbons" (p. 620). Clearly, their underlying vision is to reconstruct the outlines of scene surfaces. We elaborate on their study as it is closest to our grouping goals and to the methodology presented here.

CANC2 begins with contour detection, possibly at the finest scale only - judging by their figure 9, although their figure 11 suggests that contours from multiple scales were obtained (authors are not explicit on that). Contour segments are then grouped exploiting the cocurvilinearity grouping principle (consisting of the combination of the continuity and proximity principle) and then partitioned into smooth curves at curvature extremas. The grouping so far can be assigned to the signal and primitive level (according to Sarkar and Boyer's nomenclature).

On a structural level, CANC2 analyses the symmetric axis for each pair of curve segments. Firstly, curves are pairwise analyzed, whereby only those pairs are considered that fulfill two conditions, namely minimal length ratio and minimal overlap. The selected curve pairs are called symmetries and for each pair a few symmetric points are computed that represent their symmetric axis. The symmetries are then further selected by a constraint satisfaction network that analyses various 'mutual' characteristics between symmetries. At the end of this selection step, 'u axes' are formed. Symmetries and u-axes are then combined to detect ribbons (enclosed regions). A ribbon is abstracted by a number of structural parameters such as its width, length, orientation, etc.

CANC2 was specifically applied to the tasks of image segmentation and stereo matching. The system pursues what one could term a perfect reconstruction. But for the purpose of image classification, the 'mere' assignment to a category label, this perfect reconstruction is too costly and not suitable for other scenes (that contain other types of structures).

Later work in this direction refined some of the grouping methodology. For instance Jacobs developed a grouping algorithm that concentrated in particular on convex groupings (any non-concave polygon) (Jacobs, 1996). This type of grouping can be understood as a more general form of closure, e.g. as a closure of multiple curves. Borrar and Sarkar compare various structural grouping algorithms on an aerial image collection (Borra and Sarkar, 1997). Iqbal and Aggarwal designed a system, which detects groupings that are typical for large man-made objects such as buildings, towers, bridges and other architectural objects (Iqbal and Aggarwal, 2002). Their system extracts for instance L features and U features and determines their statistics for each image. The system is applied on high resolution images (512 pixels and larger for each side) in an image-retrieval task.

In the studies mentioned so far, the diversity of extracted groups have been rather limited and the comparison between structures is not always explicitly addressed. For instance, in Jacobs' study it is unclear how to compare the convex groupings. In contrast, recent object-detection studies using contour segments have addressed these issues slightly more elaborate, but their extent of grouping remains rather limited. For instance, Ferrari et al. use only pairs of adjacent segments and use them as templates (Ferrari et al., 2008; Ferrari et al., 2010); Zhu et al. consider any pair - not just adjacent (local) but also distant (global) ones - for finding the outline of objects but do not exploit them for matching (Zhu et al., 2008). We concur with Zhu et al.'s assumption that any pair is potentially category-specific, and while this assumption aggravates the combinatorial challenge, the selection of hypothesized groups requires correspondingly stringent criteria.

1.2 Overview

Commonly detected groups are pairs of segments and clusters of segments. With the term 'cluster of segments' we specifically mean now three or more segments; they can be aligned starwise as in a vertex feature ('Y', '+',...) or more generally formulated they can be aligned with their pairwise intersection points clustered; or they can be aligned in parallel. We had initially attempted to detect vertex features specifically, but realized that they occur in fact rarely outside room scenes or scenes with tools. Instead, what is more valuable for scene classification is a thorough description of the alignment of two segments, a *pair* descriptor, as we will present in section 2.

One challenge of grouping is to find the appropriate selections as the combinatorial analysis of hundreds of segments is computationally costly. For images smaller than 200 x 300 pixels approximately - which return several hundred segments for different spatial (image) scales - , the computations pose little problems for a nowadays computers, but image sizes beyond that can lead to memory shortage during pair analysis. It is therefore necessary to consider selections without compromising classification accuracy as in principle any pair of segments is potentially useful for discriminating categories.

Building on the pair descriptors, we generate a *quad* descriptor, which is a geometric characterization of two pairs (section 3). Its creation is analogous to the pair descriptor.

It would also make sense to form groups of three segments in order to describe triangles or any type of regularity as it may occur in shapes. We have not developed the methodology so far yet, but we will present a relatively simple radial description of near-closed shapes obtained from iso-contours, which can be regarded as a first step into that direction (section 4).

In section 5, descriptors for clusters of segments are introduced, whereby the clusters are defined predominantly by proximity and multitude and less so by exact, geometric alignment aspects as with the pair and quad descriptors, for which we use a precise number of segments and determine a precise geometry.

We further tested a set of texture descriptors whose parameters were taken from regions outlined by arcs (curved segments), certain pair (biases) and isocontours (shapes) (section 6). The parameters were taken from the output of various types of image preprocessing.

The input to most grouping processes are a list of partitioned segments $\mathcal{S}$, partitioned by the method introduced in (Rasche, 2010). The segments (coarse) geometry can be straight, curved or elongated (amplitude larger than chord length) - hereafter the term 'curved' includes the elongated case. We use the term straight for straight segments and arc for curved segments and avoid the term straight arc. For each segment we use only 3 coordinates (its endpoints and midpoint) and 2 parameters (length l and bendness b) as a starting point; the entire grouping procedure therefore has complexity $O(N^2K)$ only, with N the number of segments and K the number of operations to determine the parameters. For small images K dominates, for larger images N^2 dominates. In the latter case, N can be prohibitively large and as pointed out above, it requires selection criteria to reduce the complexity. The input to the shape description process is a list of isocontours $\mathcal{I}$. Algorithm 1 overviews the processes developed and tested in this study. In section 7 a few implementation details are given.

The grouping output is applied in an image classification task, for which we introduce the classifiers we had tested in section 8. The evaluation takes place in section 9.

Algorithm 1 Grouping processes tested in this study. segs: segments. $\mathcal{P}$: list of pairs; $\mathcal{Q}$: quads; $\mathcal{F}$: shapes; $\mathcal{C}$, $\mathcal{B}$: cluster descriptors; $\mathcal{P}^x, \mathcal{F}^x$: texture descriptors derived from corresponding descriptors outlining a region.

Input : - $\mathcal{S}$, list of partitioned segments
- $\mathcal{I}$, list of isocontours

Output: lists of vectors for different group types:
$\mathbf{p}_k(\mathcal{P}), \mathbf{q}_l(\mathcal{Q}), \mathbf{f}_i(\mathcal{F}), \mathbf{c}_j(\mathcal{C}), \mathbf{b}_m(\mathcal{B})$
$\mathbf{t}^{\mathsf{a}}, \mathbf{t}^{\mathsf{p}}(\mathcal{P}^x), \mathbf{t}^{\mathsf{s}}(\mathcal{F}^x)$

1) PAIRS($\mathcal{S}$)$\rightarrow \mathcal{P}, \mathcal{P}^x$: accurate alignment of 2 segs
CLUST($\mathcal{S}$)$\rightarrow \mathcal{B}, \mathcal{C}$: coarse alignment of ≥ 2 segs
SHAPE($\mathcal{I}$)$\rightarrow \mathcal{F}, \mathcal{F}^x$: up to 5 corners described
2) QUAD($\mathcal{P}$)$\rightarrow \mathcal{Q}$: accurate alignment of 4 segs

2 Pairs

The geometric characterization of pairs of segments we pursue, follows the observation that many objects and structures in nature are symmetric or at least partially symmetric, e.g. (Willmer, 1990); many man-made objects are symmetric as well. This (2D) symmetry consists of an even alignment of segments whether it be bilateral (mirror, reflection), rotational or radial. In case of pairs it is particularly the bilateral symmetry that is of interest.

Even if a pair of segments may be regarded as a simple structure, it affords different measures to arrive at a measure of symmetry. This maybe obvious for the cases of bilateral and radial symmetry: bilaterality is determined in reference to an axis, radiality is determined in reference to a center point. Yet for bilateral symmetry, there are cases for which it is more efficient to employ different measures. Hence, we will use different operational definitions of symmetry (for pairs). We deduce this by looking at combinations of straight and arc segments: bilateral symmetries are composed of a number of simpler symmetry measurements, which we call *evenness* measures. We will define the evenness measures such that they produce graded values, with which in turn one can generate complex symmetry measures by merely correlating the evenness measures. All these measurements can be performed in relatively short time by using only few conspicuous segment points (and not all points).

Since structures can also be distorted and an object - that normally appears as symmetric - thus frequently appears as partially symmetric only. This is a challenge for recognition because it means that a symmetry measure must be robust to small structural variations - and also the structural description of an object class needs to exhibit this robustness. Hence a symmetry measure should produce only degraded values in presence of structural distortions and not lead to immediate null values.

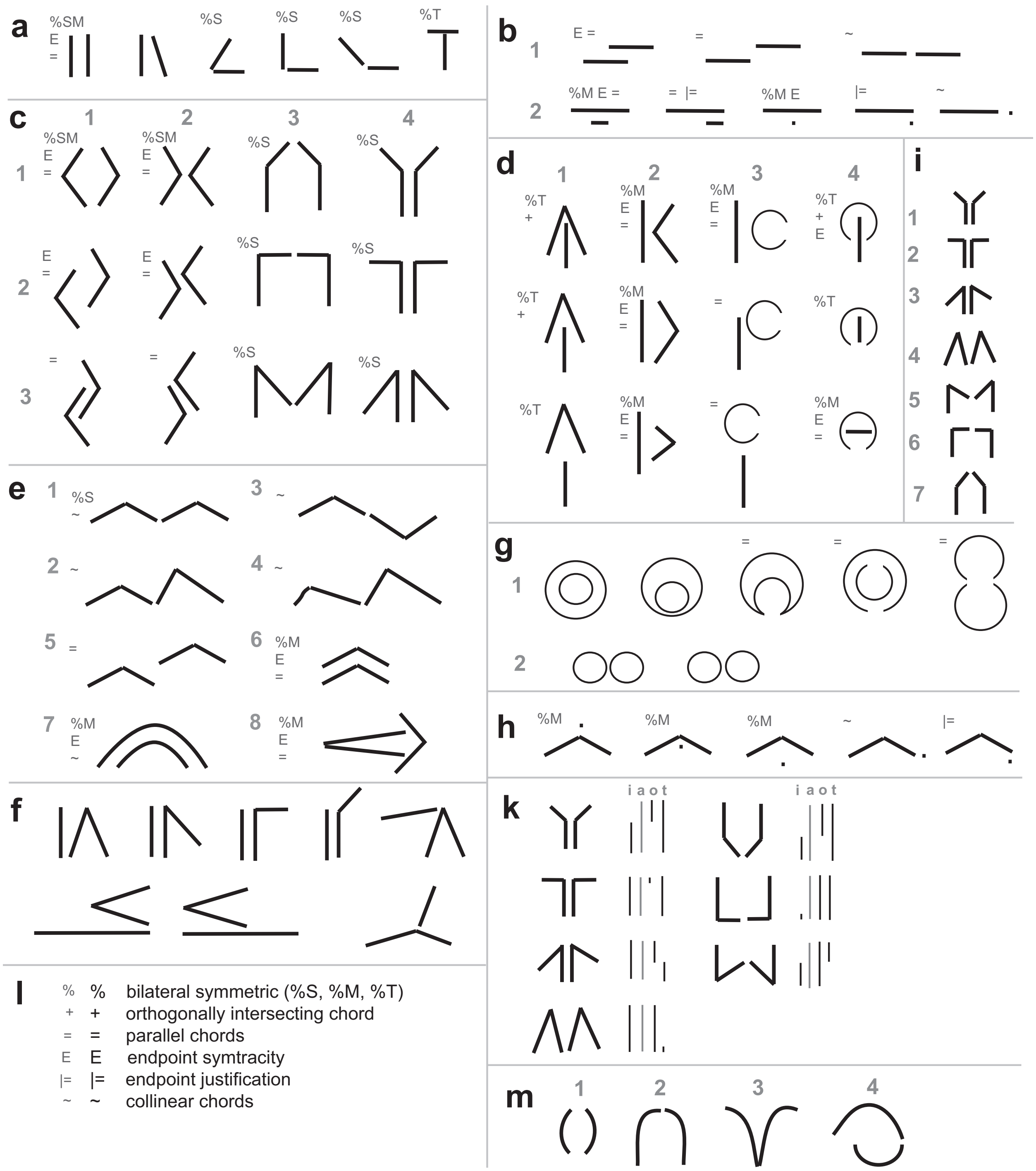

Figure 1: Pair combinations of elementary segments (straights and arcs, sometimes dots). **a.** Pairings with straight segments. **b.** Straight/straight and straight/dot pairing. Row 1: parallel to collinear; Row 2: parallel and dot pairings. **c.** Pairs of arcs. Column 1: from closure to interlocked. Column 2: from hyperbola to one-branch adjacent. Column 3: from U shape to M shape. Column 4: From Y shape to W shape. **d.** Column 1: arrow shapes; Column 2: K and D shapes; Column 3 and 4: straight/circle segment combinations. **e.** Parallel and collinear chords (for arcs). **f.** Pairs with straight and partially aligned arc. **g.** Pairs of circular (or near-circular) segments. **h.** Relations of points and arcs. **i.** Wing beat (downward motion). **k.** Axis length ratios in wing beat. **l.** Legend of symbols. **m.** Seemingly different pairs, but existent in previous subpanels already (merely with a low degree of angularity).

2.1 Overview

Figure 1 shows combinations of pairs of segments, which were organized according to certain alignment aspects. For completion, we also added segment-dot relations, but have not made an explicit effort to model those separately. Figure 1a and b contains straight/straight combinations only (and some straight/dot combinations); 1c and e exhibit arc/arc combinations; 1d and f show straight/arc combinations; 1g contains circle/circle (or near-circle) combinations; 1h exhibits arc/dot combinations; 1k shows combinations which are present already in other subpanels (to be revealed later). Most of the pair combinations have at least one type of symmetry (simple or complex). Each subpanel contains one or more sets of pairs for which a geometric attribute (dimension) gradually changes:

Panel 1a: in these combinations the intersection angle changes and the endpoint alignment varies; the combinations can be already considered traditional (e.g. (Lowe, 1985)): parallel, converging, 3 L features (acute, right and obtuse) and T feature (from left to right).

Panel 1b: contains various alignments along the parallel: row 1 contains parallel overlapping (or parallelogram), parallel non-overlapping and collinear; row 2 has parallel pairs of uneven length, with the 2nd one showing justified endpoints; the last 3 are equivalent to dot/straight combinations.

Panel 1c: pairs of the first two columns are shifted along their parallel chords: in column 1 the pairs face each other, shifted from closure to interlocked; in column 2, the point away from each other, shifted from hyperbola to one-branch adjacent. In those two columns the geometry of a single arc does not change but is merely shifted along the chord line. In columns 3 and 4, the angle of a segment changes (from acute to obtuse). In column 3, pairs range from U shape to M shape; in column 4 from Y shape to W shape.

Panel 1d: column 1 has arrow shapes of increasing spacing between the two segments; column 2 has K and D shapes; in column 3 and 4 are straight/circle combinations analogous to row 2 in panel b.

Panel 1e: Parallel and collinear chords (for arcs). For pairs no. 1 to 4 and 7, the chords are collinear and the pairs differ in amplitude (2 vs. 4) or in the coarse direction of their face angles (1 vs. 3). For pairs no. 5, 6 and 8, the chords are parallel but differ in alignment.

Panel 1f: Pairs with one curved and one straight arc, whereby one branch of the arc is aligned parallel or justified by endpoint with the straight segment. The change includes an increasing angle from left to right in the first row, and different justification in the second row.

Panel 1g: Pairs of circular (or near-circular) segments. Should the segment be only near circular, meaning containing a gap, then a number of different alignments are possible.

Panel 1h: Relations of points and arc segments. In the first three, the dot is justified to the segment's midpoint but at different sides or with different separation. In the last three are analogous to the last three cases in panel b, 2nd row.

Panel 1i: Legend of symbols for some measures.

Panel 1k: These combintations already exist in previous subpanels already, but have a different degree of edginess (angularity). Pairs no. 1 to 3 correspond to pairs in row 1 of subpanel c. Pair no. 2 corresponds to the pair row=1/column=3 in panel c. Pair no. 4 is a degenerate form of pair no. 1.

All these combinations can be determined relatively accurately and fast using only a few conspicuous points. Starting with only three conspicuous points per each curve segment - its two endpoints and its (curve's) midpoint -, one proceeds to determine the three symmetric points and some other key points for each pair (to be detailed in subsection 2.2).

Given the array of pair combinations, we identify three types of bilateral symmetry:

- **Sym-Ax Bilaterality** is the symmetry in reference to Blum's symmetric axis (sym-ax; (Blum, 1973)) of the pair (figure 4a). Such pairs are marked as %S in figure 1, e.g. 1st row in panel a and c, or 3rd and 4th column in panel c.
- **Middle-Ax Bilaterality** is the symmetry in reference to the 'middle' axis that runs through the two curves' midpoints, such as in a D shape ('|' and ')') or in a K shape ('|' and '<'). Pairs are marked as %M in figure 1, see for instance column 2 in panel d.
- **Segment Bilaterality** is the symmetry in reference to one (elongated) curve segment of the pair, such as in a T shape. Pairs are marked as %T, see for instance 1st column in panel d.

Three pairs show both bilaterality types, sym-ax and middle-ax bilaterality, see panel a and c. They differ from each other just by the distance of the segments' midpoints. One could also define a radial symmetry measure, for instance by measuring to what extent the segments' endpoints are equidistant from the pair's center point. But that is a rare case we do not pursue here. Some combinations bear also rotational symmetry, which also is not treated here as we pursue mainly image classification for which there exists a strong orientation dependence.

The bilateral symmetries can be broken down into a number of simpler symmetries such as parallel chords, equal separation between endpoints and so on. These simple symmetries are now called *evenness* measures; they can be regarded as attributes of the 'full' symmetry. Some of these evenness measures are ordinary Gestalt groupings. They are developed next (subsection 2.3), after some terminology and definitions was introduced (subsection 2.2). Then, the structural biases are developed (subsection 2.4): they are often composed of evenness measures and some of them represent the bilateral symmetries.

2.2 Terminology and Definitions

For a single segment - whether straight or curved - we determine its halfpoint and face angel:

- *Halfpoint*, $\mathsf{p_h}$: the midpoint of a curve segment's chord. If the curve segment is straight, then midpoint and halfpoint are the same.

- *Face Angle*, ϕ : is the directional angle of the line segment pointing from a segment's midpoint to its halfpoint (if classified as curved arc).

The following terminology for a pair of segments is chosen for the case of L features and aims at simplifying the understanding of the measurements. For other alignments, such as parallels, T features, closure, some of the terms may become irrelevant. The labeling is illustrated in figure 2a and b, figure 3, and is summarized in table 1:

- *Connecting segments*: line segments, that connect the corresponding conspicuous points: the *corner* segment, connecting the two proximal endpoints $\overleftrightarrow{\mathsf{p_{c1}p_{c2}}}$; the *middle* segment, connecting the two midpoints $\overleftrightarrow{\mathsf{p_{m1}p_{m2}}}$; the *open* segment, connecting the two distal endpoints (open points) $\overleftrightarrow{\mathsf{p_{o1}p_{o2}}}$. The corresponding segment orientation angles are o_c, o_m and o_o ($\leq \pi$). The midpoints of those segments correspond to symmetric points (sym-points) and are called corner point, middle point and open point, $\mathsf{p_c}$, $\mathsf{p_m}$ and $\mathsf{p_o}$, respectively. The segment lengths are the distances d_c, d_m and d_o.

- *Axes*: three sym-axis approximations are formed (see figure 3a and b): the *inner* axis, $\overleftrightarrow{\mathsf{p_c p_m}}$; the *outer* axis, $\overleftrightarrow{\mathsf{p_m p_o}}$; and the *total* axis $\overleftrightarrow{\mathsf{p_c p_o}}$.

- β angles: are the intersecting angles of the connecting segments and the sym-axes (inner, outer and total axis). The angles β_c, β_m and β_o are formed between the connecting segments and the inner and outer axes, see figure 3a. Angles β_{tc} and β_{to} are formed between the corner and open connecting segments and the total axis (figure 3b).

- *Extension distance:* the distance between the intersection point and the proximal segment endpoint, $d_{\mathsf{e1}}=\overline{\mathsf{p_i p_{c1}}}$ and $d_{\mathsf{e2}}=\overline{\mathsf{p_i p_{c2}}}$, respectively.

Table 1: Labeling in a pair of segments. dist: distance; seg: segment; pt: point.

1	l_1, l_2	segment curve lengths, $l_1 \geq l_2$
2	ℓ_1, ℓ_2	segment chord lengths
3	$\mathsf{p_h}$	halfpoint: midpoint of chord
4	ϕ	face angle: direction from mid- to halfpoint, $[0, \pi]$
5	κ	chords' intersecting angle, $[0, \frac{\pi}{2}]$
6	$\mathsf{p_{c1}}$, $\mathsf{p_{c2}}$	proximal endpoints
7	$\mathsf{p_{m1}}$, $\mathsf{p_{m2}}$	midpoints
8	$\mathsf{p_{o1}}$, $\mathsf{p_{o2}}$	open (outer) endpoints
9	$\mathsf{p_i}$	intersection point of the chords' line equations
10	d_{e1}, d_{e2}	extension distances: $\overline{\mathsf{p_i p_{c1}}}$, $\overline{\mathsf{p_i p_{c2}}}$
11	d_{f1}, d_{f2}	full distances: $\overline{\mathsf{p_i p_{o1}}}$, $\overline{\mathsf{p_i p_{o2}}}$
12	d_{sh}	shift dist: $\overline{\mathsf{p_{h1} p_{ah2}}}$, $\mathsf{p_{ah2}}$ on chord 1
		connecting segments
13	$\overleftrightarrow{\mathsf{p_{c1}p_{c2}}}$	corner segment
14	$\overleftrightarrow{\mathsf{p_{m1}p_{m2}}}$	middle segment
15	$\overleftrightarrow{\mathsf{p_{o1}p_{o2}}}$	open segment
16	$\mathsf{p_c}$	corner point (midpt of corner seg)
17	$\mathsf{p_m}$	middle point (midpt of middle seg)
18	$\mathsf{p_o}$	open point (midpt of open seg)
19	d_c	corner distance, $\lvert\overleftrightarrow{\mathsf{p_{c1}p_{c2}}}\rvert$
20	d_m	middle distance, $\lvert\overleftrightarrow{\mathsf{p_{m1}p_{m2}}}\rvert$
21	d_o	open distance, $\lvert\overleftrightarrow{\mathsf{p_{o1}p_{o2}}}\rvert$
22	$\overleftrightarrow{\mathsf{p_c p_m}}$	inner axis
23	$\overleftrightarrow{\mathsf{p_m p_o}}$	outer axis
24	$\overleftrightarrow{\mathsf{p_c p_o}}$	total (complete) axis
25	β	angles between connecting segments (13-15) and sym-axes (19-21)

- *Full distance:* the distance between the intersection point and the distal endpoint, $d_{\mathsf{f1}}=\overline{\mathsf{p_i p_{o1}}}$ and $d_{\mathsf{f2}}=\overline{\mathsf{p_i p_{o2}}}$, respectively.

- *Shift distance:* expresses to what degree two (approximately) parallel line segments are shifted with respect to their chords: $d_{\mathrm{sh}}=\overline{\mathsf{p_{h1} p_{ah2}}}$, whereby $\mathsf{p_{ah2}}$ is the point on chord line 1, for which the separation between half point 2 and chord line 1 is shortest.

2.3 Evenness Measures

In this subsection, the term similarity is synonym for evenness. The measures are defined as ranging between 0 and 1, so are typically the angles (unit turn) - we give radians sometimes for clarity. By using turn range, we can conveniently correlate the measures with each other to generate more complex measures.

- **Equi-Gap**, $\epsilon_{\shortparallel}\epsilon_{\shortparallel}$: measures the similarity between the corner and open distances (between the

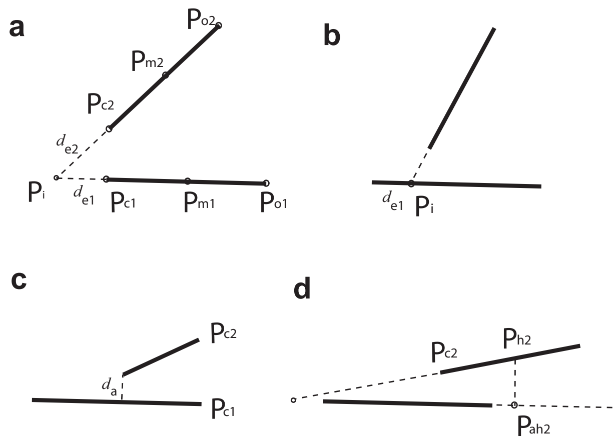

Figure 2: Ambiguous spatial alignments of two contour segments. (p_i=intersection point; p_{c1},p_{c2}=proximal endpoints; p_{m1},p_{m2}=midpoints; p_{o1},p_{o2}=open (outer) endpoints). **a.** An intermediate form of a converging and an L feature. d_{e1} and d_{e2} denote the distances from the endpoint to the intersection point. **b.** An intermediate form of an L and a T feature. **c.** An intermediate form of a T and parallel feature. **d.** An intermediate form of a L and parallel feature. Note in this case, that the intersection point is distal from the corner point.

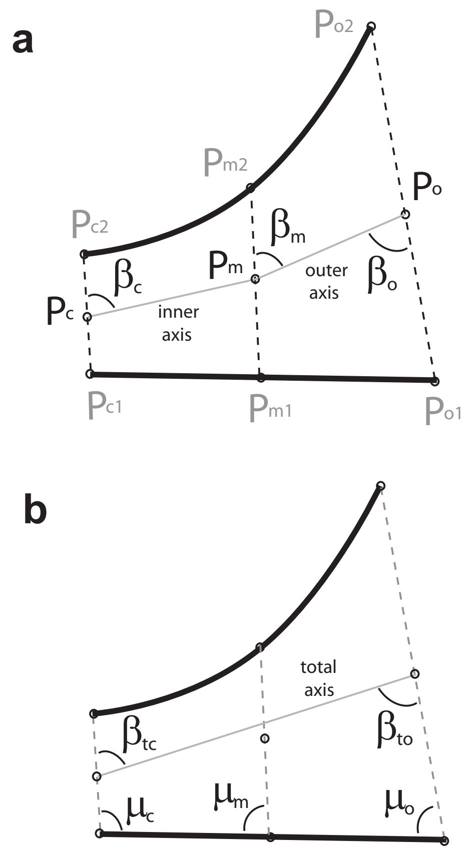

Figure 3: Measurements in a pair of segments. **a.** p_c=corner point; p_m=middle point; p_o=open point. Dashed: connecting segments. Gray: axis segments. **b.** Total axis (gray) and its related β angles; δ angles.

two gaps of the pair):

$$\epsilon_{\shortparallel} = 1 - \tanh\left(\frac{d_c - d_o}{l_2}\right), \tag{1}$$

where l_2 serves as a scaling factor.

• **Visavis**, $\epsilon_{\div}$: describes to what extent two segments are opposite (vis-a-vis) of each other by computing the degree of even alignment between the two curves' *half* points. d_{sh} is the shift distance and $T^{\div}$ is a tolerance dependent on the shorter chord length ℓ_s:

$$\epsilon_{\div} = \begin{cases} \frac{T^{\div} - d_{\text{sh}}}{T^{\div}} & , d_{\text{sh}} < T^{\div}, \quad T^{\div} \propto \ell_s \\ 0 & , \text{else.} \end{cases} \tag{2}$$

For a zero shift distance the measure is maximal; for increasing shift distance the measure decays. The measure selects approximately parallel segments or T features, but ignores L features.

• **Chord Orthogonality and Parallelism**, $\upsilon_{\perp}$ and $\upsilon_{\shortparallel}$: are measures that express how orthogonal or parallel the two chords are, for which the intersection angle κ (turn range) is used:

$$\upsilon_{\perp} = \kappa^2 \tag{3}$$

$$\upsilon_{\shortparallel} = (1 - \kappa)^2, \tag{4}$$

whereby the square operation serves as an accentuation.

• **Face Equi-Directionality and Oppositionality**, $\omega_{\twoheadrightarrow}$ and $\omega_{\leftrightarrow}$: describe to what extent the two face angles point into the same direction or into opposite direction. With $\Delta\phi$ as the difference in face angles ([0..π], normalized to turn range resp.), the two measures are defined analogous to chord orthogonality and parallelism:

$$\omega_{\twoheadrightarrow} = (1 - \Delta\phi)^2 \tag{5}$$

$$\omega_{\leftrightarrow} = \Delta\phi^2. \tag{6}$$

Note that with the two measures one cannot discriminate between closure and hyperbola features, for which it requires additional measurements (to be further specified in 2.4.2).

• **Chord Intersectionality**, υ_{+} : expresses to what degree the two chords intersect, which occurs when an arc intersects with a segment as shown in figure 1d, columns 1 and 4 for instance. The measure is maximal if the chords intersect at their midpoints (bisect each other). This can be determined using the extension distances (in which case they are not actual 'extensions'):

$$\upsilon_+ = \begin{cases} \left(1 - \frac{d_{f1} - d_{e1}}{l_1}\right)\left(1 - \frac{d_{f2} - d_{e2}}{l_2}\right) & , c_+ \\ 0 & , \text{else.} \end{cases} \tag{7}$$

where c_+ is the condition that the intersection point p_i lies on both chords. For intersection points which move away from either midpoint, the value decreases toward 0.

• **Chord Collinearity**, $\lambda^\frown$: collinearity measures to what extent the two chords lie on the same straight line. The measure is composed of two conditions and a correlation. One condition is that the distance between midpoints, $d_{\overline{hh}}$, is larger than half the sum of the chord lengths; the other condition is that the distance does not exceed a tolerance dependent on the shorter segment, $T_{\lambda^\frown} \propto l_2(1-\chi)$. Specifically, the measure is:

$$\lambda^\frown = \begin{cases} \left(\frac{(T_{\lambda^\frown} - d_{ah})}{T_{\lambda^\frown}} \cdot \upsilon_{\shortparallel}\right)^2 & , \; d_{\overline{hh}} > (\ell_1 + \ell_2)/2 \\ & ...d_{ah} < T_{\lambda^\frown} \\ 0 & , \text{else,} \end{cases} \tag{8}$$

where squaring serves to accentuate the collinear alignments.

• **Ratio Curve Lengths**, r_l : The similarity of segment lengths is expressed by a simple ratio:

$$r_l = l_2/l_1. \tag{9}$$

2.4 Structural Biases

Structural biases are organized into three groups. One group consists of simple or preparatory biases, which are later used for constructing bilateral symmetries and complex biases. They are treated next. The other groups are the bilateralities (subsection 2.4.1) and the facing biases (subsection 2.4.2).

Due to the formulation of the previous evenness measures, many of the biases treated subsequently can now be expressed relatively easily by the evenness measures, for instance by a mere multiplication.

• **Lean Proximity**, χ^g : describes the proximity between two segments (and is thus not an actual structural bias). To allow for the possibility of global pairings, it is defined rather lax but starts to decline rapidly after approximately 5 times the length of the shorter segment l_2:

$$\chi^g = \begin{cases} 1 & , l_2/d_m > 1 \\ 1 - (1 - l_2/d_m)^5 & , \text{else.} \end{cases} \tag{10}$$

The measure is used in many structural biases, because without it, many pairs consisting of short, distal segments would exhibit some type of symmetry that would make them a prefered choice and which therefore would render the description unspecific.

• **Endpoint Symtracity**, $\epsilon_{\overline{\wedge}}$: measures to what extent the segments' endpoints form an isosceles trapezoid - also called *symtra* by Halsted, pp. 49 (Halsted, 1896):

$$\epsilon_{\overline{\wedge}} = \epsilon_{\shortparallel} \cdot \upsilon_{\shortparallel} \cdot \chi^g. \tag{11}$$

The symtra is the least symmetric of the endpoints-quadrilateral, but the endpoints could also form more symmetric alignments such as a parallelogram, rectangle or square. The measure is a component of the mid-ax bilaterality measure (see also panel figure 1c, row 2).

2.4.1 Bilateralities

• **Sym-Ax Bilaterality**, y_s : is measured by adding the β intersection angles. For perfect sym-ax bilaterality, the angles are all orthogonal and the segment geometries are the same. For increasing differences in segment geometries, the angles decrease, but only gradually and the measure is thus robust (tolerant). Also for reason of robustness, the individual evenness measures are added and not multiplied. For instance, an L feature may not be perfectly aligned in its corner (vertex) and yet still be relatively symmetric, compare figure 4a and b. A similar case is shown for a hyperbola feature, compare figure 4c and d. Thus, if one angle showed a near-zero value due to a slight misalignment, a multiplicative measure would drop the measure to near zero. Additional robustness is provided by adding the angles from the total axis,

$$y_s = (\beta_c + \beta_m + \beta_o + \beta_{tc} + \beta_{to})/5 \cdot \chi^g \cdot \breve{s}_{\varkappa}, \tag{12}$$

whereby the measure is normalized by the number of summands.

• **Mid-Ax Bilaterality**, y_m : can be expressed by the correlation of the symtracity and the visavis evenness (equation 2),

$$y_m = \epsilon_{\overline{\wedge}} \cdot \epsilon_{\div}. \tag{13}$$

Robustness is naturally provided by the evenness measures.

• **Segment Bilaterality (T bias)**, $\breve{s}_{\mathsf{T}}$: if the intersection point lies on the chord of either segment, then the alignment is of type 'T'. In this case, one extension distance and one full distance correspond to the two 'branches' of the intersected segment. If the intersection point lies exactly on a halfpoint, then the T bias is maximal. It decreases with increasing distance $\overline{\mathsf{p_i p_h}}$, that is, if the intersection point moves toward either endpoint:

$$\breve{s}_{\mathsf{T}} = \begin{cases} 1 - \frac{2\overline{\mathsf{p_i p_h}} d_{\mathrm{a}}}{\ell_2} & , c_{\mathsf{T}} \ \cap \ d_{\mathrm{a}} < \ell_2 \\ 0 & , \text{else}, \end{cases} \tag{14}$$

whereby c_{T} is the condition that the intersection point $\mathsf{p_i}$ lies on either chord:$((d_{\mathsf{e}1} + d_{\mathsf{f}1}) < \ell_1 \cup (d_{\mathsf{e}2} + d_{\mathsf{f}2}) < \ell_2)$. d_{a} is the displacement between the segment endpoint - that is closest to the intersected segment - and the chord of the intersected segment (figure 2c).

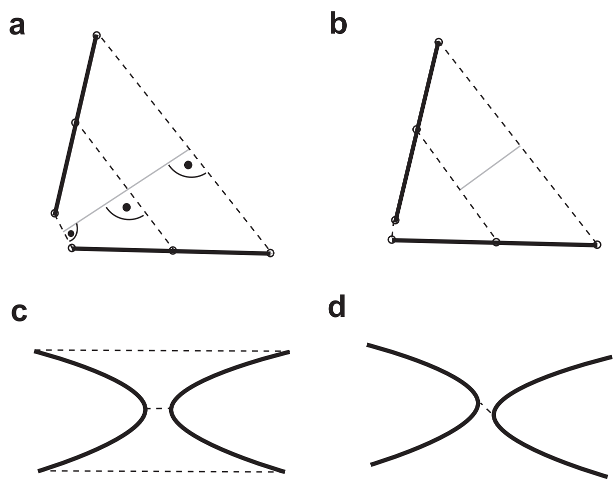

Figure 4: Symmetry in pairs. **a.** Symmetric L feature: angles between axis (gray) and dashed line segments are orthogonal. **b.** Slightly asymmetric L feature: β_c misaligned. **c.** Symmetric hyperbola feature. **d.** Slightly asymmetric hyperbola feature: β_m misaligned.

2.4.2 Facing

Three biases for arc/arc combinations are developed now, which differ from each other in the way the arc segments face each other, hence they are called the facing biases. They are the hyperbola, the closure and the banana bias (figure 5). When they exhibit symmetry, they belong to the sym-ax and mid-ax bilaterality. But even if stronly asymmetric, they can be very conspicuous pairs in scenes and are therefore worth formulating with a degree of tolerance.

The three biases can be distinguished from each other by the direction of their 'faces' (the 'open' side of the segment), figure 5. For a hyperbola feature, the faces point away from each other; for a closure feature they point toward each other; for a banana feature they point into the same direction. To discriminate between these cases, four directional angles are determined, two angles for each segment. For a segment, the two angles originate from the segment's midpoint: one is the face angle ($\phi = \overrightarrow{\mathsf{p_{m1} p_{h1}}}$, dotted in figure), the other angle points to the other segment's midpoint ($\tau = \overrightarrow{\mathsf{p_{m1} p_{m2}}}$, dashed in figure). The corresponding two angles are formed for the second segment ($\angle\overrightarrow{\mathsf{p_{m2} p_{h2}}}$ and $\angle\overrightarrow{\mathsf{p_{m2} p_{m1}}}$ respectively). For any of the three structural biases, the two angles (for one segment) point in approximately opposite or same directions, abbreviated as 'opp' and 'same' in figure. In case of the hyperbola feature, the two angles point in opposite directions for each segment, an opp/opp pairing whose condition is short-noted as ϕ_{oo}; for the closure feature, both angles point in the same direction for each segment, a same/same pairing (ϕ_{ss}); and for the banana feature the pairing is of type same/opp (ϕ_{os}). The minimum tolerance for the three conditions is if the difference of the two angles (ϕ and τ) differ by more than $\pi/2$ for the opp case, or by less than $\pi/2$ for the same case.

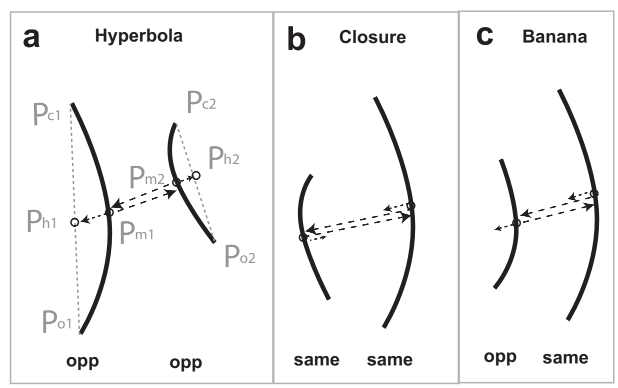

Figure 5: Conspicuous arc/arc structural biases. Face angle: directional angle (of a segment) pointing from midpoint to halfpoint. **a.** Hyperbola bias: face angles point away from each other (ϕ_{oo}). **b.** Closure bias: face angles point toward each other (ϕ_{ss}). **c.** Banana bias: face angles point the same side (ϕ_{os}); see also panel figure 1e, no. 6 and 7.

The biases are formulated as consisting of a bias that is common to all of them, the facing bias, and of the individual measures and conditions.

• **Facing**, $\breve{s}_{\varkappa}$: in all three cases, only pairs for which both segments are curved are allowed, $(a_1 > 0) \cup (a_2 > 0)$, where a is an arc parameter as defined in our previous study (a_1, a_2: the arc values for each segment). The segments are required to lie approximately vis-a-vis and we factor in lean proximity and

Table 2: Summary of evenness measures (E) and structural biases (B). See also table 1. Bilat=Bilaterality.

E	Equi-Gap	$\epsilon_{\shortparallel} = 1 - \tanh((d_c - d_o)/l_2)$
E	Vis-a-vis	$\epsilon_{\div} = (T^{\div} - d_{\mathrm{sh}})/T^{\div}$ if $d_{\mathrm{sh}} < T^{\div}, 0$ else $T^{\div} \propto \ell_s$
E	Chord Orthogonality	$\upsilon_{\perp} = \kappa^2$ κ: intersection angle of the 2 chords
E	Chord Parallelism	$\upsilon_{\shortparallel} = (1-\kappa)^2$
E	Chord Intersectionality	υ_{+} see equation 7
E	Face Equi-Directionality	$\omega_{\twoheadrightarrow} = (1 - \Delta\phi)^2$ $\Delta\phi$: diff in face angles
E	Face Oppositionality	$\omega_{\leftrightarrow} = \Delta\phi^2$
E	Ratio Seg Lengths	$r_l = l_2/l_1$
B	Proximity	$\chi^g = 1,\quad$ if $l_2/d_m > 1, 1-(1-l_2/d_m)^5,$ else
B	Endpoint symtra	$\epsilon_{\barwedge} = \epsilon_{\shortparallel} \cdot \upsilon_{\shortparallel} \cdot \chi^g$
B	Sym-Ax Bilat [L,V,(),...]	$y_s = (\beta_c + \beta_m + \beta_o + \beta_{tc} + \beta_{to})/5 \cdot \chi^g \cdot \breve{s}_{\varkappa}$
B	Mid-Ax Bilat [K,D]	$y_m = \epsilon_{\barwedge} \cdot \epsilon_{\div}$
B	Segment Bilat [T]	$\breve{s}_{\mathsf{T}} = 1 - (2\overline{\mathsf{p_i p_h}} d_{\mathrm{a}}/\ell_2)$
B	Facing	$\breve{s}_{\varkappa} = \epsilon_{\div} \cdot \chi^g \cdot r_l$ if both segments curved, 0 else
B	Hyperbola	$\breve{s}_{)(} = \breve{s}_{\varkappa} \cdot \omega_{\leftrightarrow}$ if both segments face away, 0 else
B	Closure	$\breve{s}_{()} = \breve{s}_{\varkappa} \cdot \omega_{\leftrightarrow} \cdot \upsilon_{\shortparallel}$ if both segments face each other, 0 else
B	Banana	$\breve{s}_{))} = \breve{s}_{\varkappa} \cdot \omega_{\twoheadrightarrow} \cdot \upsilon_{\shortparallel}$ if both segments face same side, 0 else
B	Ribbon	$\breve{s}_{\mathrm{Rib}} = \epsilon_{\div} \cdot \chi^g \cdot r_l \cdot \upsilon_{\shortparallel}$

length similarity:

$$\breve{s}_{\varkappa} = \begin{cases} \epsilon_{\div} \cdot \chi^g \cdot r_l & , (a_1 > 0) \cup (a_2 > 0) \\ 0 & , \text{else.} \end{cases} \tag{15}$$

• **Hyperbola**, $\breve{s}_{)(}$; **closure**, $\breve{s}_{()}$; **banana**, $\breve{s}_{))}$: the biases are considered optimal if the face angles point opposite directions ($\omega_{\leftrightarrow}$) for the hyperbola and closure bias, or if the face angles point equal directions ($\omega_{\twoheadrightarrow}$) for the banana bias. We attempted to exclude chord parallelism, as to include the largest facing variability possible, but for the closure and the banana bias, chord parallelism was necessary, otherwise some features would not resemble any facing. In summary:

$$\breve{s}_{)(} = \begin{cases} \breve{s}_{\varkappa} \cdot \omega_{\leftrightarrow} & , \phi_{oo} \\ 0 & , \text{else} \end{cases} \tag{16}$$

$$\breve{s}_{()} = \begin{cases} \breve{s}_{\varkappa} \cdot \omega_{\leftrightarrow} \cdot \upsilon_{\shortparallel} & , \phi_{ss} \\ 0 & , \text{else} \end{cases} \tag{17}$$

$$\breve{s}_{))} = \begin{cases} \breve{s}_{\varkappa} \cdot \omega_{\twoheadrightarrow} \cdot \upsilon_{\shortparallel} & , \phi_{os} \\ 0 & , \text{else.} \end{cases} \tag{18}$$

• **Ribbon**, $\breve{s}_{\mathrm{Rib}}$: The ribbon pair is very similar to the closure and banana bias:

$$\breve{s}_{\mathrm{Rib}} = \epsilon_{\div} \cdot \chi^g \cdot r_l \cdot \upsilon_{\shortparallel}. \tag{19}$$

2.5 Selection

Ideally one would consider all pair combinations for learning, as any pair can be potentially category specific and thus suitable for category representation. Practically, we are limited by storage size, meaning we can not save the long list of parameters for all segment pairs (square complexity); and we are limited by the learning duration, meaning the search for category specific descriptors requires reduced lists, otherwise learning becomes unfeasibly long. Thus, the reduction of pairs requires criteria that do not discard too many descriptors that are potentially category specific. Several criteria can be considered. One criterion is proximity, but even distal segments can be category specifics that is why we used a lean proximity bias (eq. 10). Another criterion is symmetry, which is automatically included in the formulation of our structural biases. We accommodate both criteria by simply choosing those pairs that exhibit a 'high' bias value for a *chosen segment* under investigation. We need to chose appropriate thresholds to determine what we consider high values. There are two principal cases to be considered:

- Same bias type: a group of three parallel straights with uneven spacing, e.g. '| ||' shows two high ribbon values for each segment. It is natural to describe this group with two ribbons only, the left and the right one (| | and ||, resp.), but even the third one consisting of the two outer segments can be a useful pair for category discrimination as it is equally symmetric as the other two. Thus it requires a threshold that decides which pairs of the same bias type are worth keeping. We chose one that is based on a fraction of the maximum value for a given segment i under investigation: we observe the (pair) bias values in connection with all other segments j ($j = 1, .., N; j \neq i$) in distribution $B^{\breve{s}}(j)$ for a chosen structural bias and select the threshold $\Theta_{\breve{s}s} = S_{\breve{s}s} \max_j B^{\breve{s}}(j)$, where subscript $\breve{s}s$ stands for 'same bias type' and $S_{\breve{s}s}$ is the fraction.

- Different bias type: a group with two straights and one arc, e.g. '| ()' shows three high bias values, namely for a K, a D and a closure bias. This requires a criterion to decide what other different biases we intend to permit. The criterion is based on observing all bias values for each type and determining is maximal value.

Summarized from the perspective of a (single) pair formed by segments i and j: the pair is selected if for *any* bias value $\breve{s}$ two conditions are fulfilled: 1) $\breve{s} > \Theta^{j}_{\breve{s}s}$ or $\breve{s} > \Theta^{i}_{\breve{s}s}$; 2) $\breve{s} > \Theta_{\breve{s}d}$. We have not specified yet for which biases this selection is carried out, but it is natural to take all structural biases, in particular the bilaterality and facing biases.

2.6 Pair Vector

For each selected pair $\in \mathcal{P}_k$, a vector $\mathbf{p}$ with the following dimensions is created:

$$\mathbf{p}(o, \gamma, d_c, d_m, d_o, l_\mu, r_l, b_1, b_2, y_m, y_s, ... \\ \breve{s}_{\mathsf{T}}, \breve{s}_{)(}, \breve{s}_{()}, \breve{s}_{))}, \breve{s}_{\mathrm{Rib}}, \{apps\}). \quad (20)$$

where o is the pair's total-axis orientation; γ the directional angle from the corner point to the open point - if the pair's open distance is larger than its corner distance (e.g. as in a L or V feature); l_μ the average segment length, $l_\mu = (l_1 + l_2)/2$; b_1 and b_2 are the curvature values for each segment.

Algorithm 2 Selecting Pairs.

Input : $\mathcal{S}$, list of partitioned segments
Output: $\mathbf{p}_k$, list of pair vectors
1) Select long segments using median length value $\rightarrow \mathcal{S}^{long}$
2) For $\mathcal{S}^{long}$ generate pairs (i, j) and their corresponding biases
3) Select pairs according to bias criteria $\rightarrow \mathcal{P}$
4) Create vector $\mathbf{p}_k$ for each element $\in \mathcal{P}$ (eq. 20)

A substantial set of biases was implemented, that cover many but not all pair alignments as illustrated in figure 1.

3 Quads

To understand what conspicuous alignments can be described by two pairs of segments, a quad, we observe a few pair alignments in figure 7. The first geometric characteristic is the alignment of the pairs' total axes (gray stippled). In alignments **a** to **e** the two axes lie parallel in the 2D plane; more specifically, in alignments **a** to **c** they are aligned like ribbons; in alignments **d** and **e** they are collinear; in **f** they intersect. The second characteristic in what direction a pair's total axis points to by observing a pair's directional angle γ (eq. 20) - if the pair is not a ribbon.

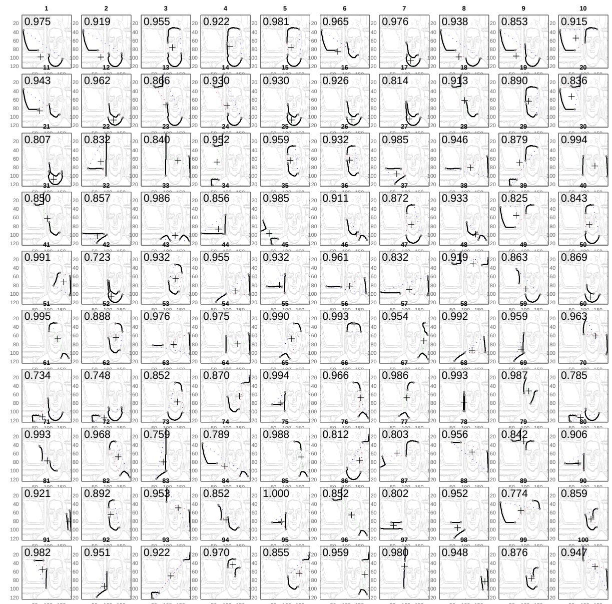

Figure 6: Pairs of a face image (see figure 1). Only the first 100 salient pairs are shown; the two segments are drawn in black, the center (middle point) is denoted by a plus sign.

If both directional angles point toward the same side, as in **b**, we label this a M bias; if the two angles point in opposite directions, as in case **c**, we label it a N bias. If one pair is a ribbon and the other a pair with an angle pointing away and a collinear ribbon axis as in case **d**, then we label it a Y bias; in case **e** the angle points toward the ribbon and we label it an arrow bias. In case **f**, the two pairs' axes form a cross, hence a cross bias; this case also illustrates that one pair can be nested within another; one form of symmetry is that the nesting occurs concentric.

And there are many alignments which are analogous to the alignments developed for pairs of segments (figure 1). We therefore use the same notation for the structural biases but for quads they are determined with the corresponding total axes. In what follows is a description of the biases typical for quads.

Whether a pair is nested within another can be conveniently determined using a pair's silhouette points only, e.g. $\mathsf{p_{Sil}}$=$\{\mathsf{p_{c1}}, \mathsf{p_{m1}}, \mathsf{p_{o1}}, \mathsf{p_{c2}}, \mathsf{p_{m2}}, \mathsf{p_{o2}}\}$ and the pole of the pair, which is defined as $\mathsf{p}_{\mathsf{Pol}} = (\mathsf{p_c} + \mathsf{p_o})/2$. For two pairs under investigation we determine also the smallest of those radii, denoted $r_{\min}$.

• **Concentricity**, $\breve{s}_{\circledcirc}$: describes how concentric two pairs are by observing the middle distance between the two axes:

$$\breve{s}_{\circledcirc} = \begin{cases} 1 - d_m/r_{\min} & , d_m < r_{\min} \\ 0 & , \text{else}, \end{cases} \tag{21}$$

The bias is maximal if the middle distance d_m is zero which means the two axes' middle points lie on top of each other. The bias decreases for increasing distance and is zero if the spacing is larger than the minimum radius; this ensures that a smaller pair within a much larger one, does not exhibit concentricity even if the smaller one sits in the corner of the larger one.

• **Crossness**, $\breve{s}_{+}$: describes to what degree the axes form a symmetric plus (or x) sign and is expressed by correlating the orthogonality and intersectionality measure:

$$\breve{s}_{+} = \upsilon_{+} \cdot \upsilon_{\perp} \tag{22}$$

• **M and N**, $\breve{s}_M, \breve{s}_N$: the biases can be conveniently expressed using the evenness measures vis-a-vis, equi-directionality and oppositionality:

$$\breve{s}_M = \epsilon_{\div} \cdot \omega^2_{\twoheadrightarrow} \tag{23}$$

$$\breve{s}_N = \epsilon_{\div} \cdot \omega^2_{\leftrightarrow}, \tag{24}$$

• **Y and Arrow**, $\breve{s}_Y, \breve{s}_{Arw}$: we intend to describe how tight a Y alignment is at its center, that is at the proximal lying axes' endpoints, in comparison to the axes' distal endpoints. For the arrow alignment we determine the inverse, namely how wide its center is. d_p is now defined as the maximum symmetric distance of the proximal lying symmetric distances, which can be either corner or open distances depending on the pair's geometry; d_d is defined as the maximum symmetric distance of the distal lying distances. The corresponding normalized ratios are taken and multiplied with the measures for collinearity and proximity:

$$\breve{s}_Y = \tanh\Big(\frac{d_d - d_p}{d_p}\Big) \cdot \lambda^{\frown} \cdot \chi^g, \tag{25}$$

$$\breve{s}_{Arw} = \tanh\Big(\frac{d_p - d_d}{d_d}\Big) \cdot \lambda^{\frown} \cdot \chi^g. \tag{26}$$

Selection and Vector Formation A selection analogous to the ones in pairs is carried out. For each selected quad $\in \mathcal{Q}_l$, a vector $\mathbf{q}$ with the following dimensions is created:

$$\mathbf{q} = (\breve{s}_{+}, \breve{s}_{\circledcirc}, \breve{s}_M, \breve{s}_N, \breve{s}_Y, \breve{s}_{Arw}, ..., \{apps\}) \tag{27}$$

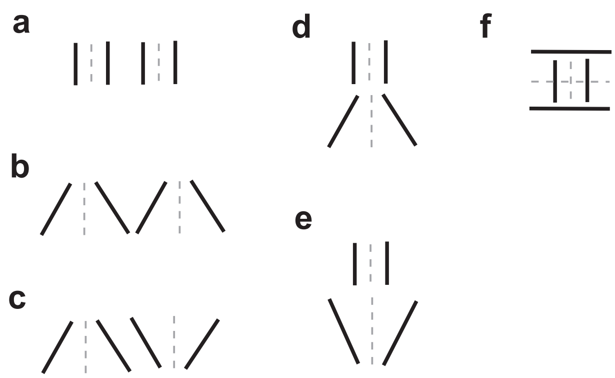

Figure 7: Some conspicuous alignments for pairs: the gray stippled axes are to be observed. **a.** Two ribbons aligned as a ribbon. **b.** Pairs with directional angle γ pointing toward the same side: M bias. **c.** γ angles pointing toward opposite directions: N bias. **d** Y bias. **e**. Arrow bias. **f**. Cross bias.

Algorithm 3 Selecting Quads.

Input : $\mathcal{P}$, list of (selected) pairs
Output: $\mathbf{q}_l$, list of quad vectors
1) Generate pairs (i, j) and their corresponding biases
2) Select quads according to bias criteria $\rightarrow \mathcal{Q}$
3) Create vector $\mathbf{q}_l$ for each element $\in \mathcal{Q}$ (eq. 27)

4 Shapes

Shapes are obtained from an isocontour analysis of the image, as opposed to an edge analysis for the contour segments used previously. Isocontours were not partitioned and were immediately analyzed for structural biases as presented below. The structural analysis is based on the isocontour's radial signature and is thus of low complexity as opposed to the partitioning and grouping analysis for edge contours.

We particularly focus on describing 'simple', planar, closed or near-closed shapes. Here, simple is understood as circular or loosely cyclic, whereby we define the latter as simple (non self-intersecting) polygons or as star shapes (star-shaped, non self-intersecting polygons). If the shape is fragmented, its gap sizes should be small only, such that the shape appears as nearly closed. In this study, only shapes with a single gap are considered, that is closed or near-closed curves. Such curves occur in abundance in gray-scale images of real-world scenes, especially at a finer scale.

The shape representation is based on the radial signature, which describes the distances of the pixels to the centroid (pole) of the shape, also called centroid-distance function or radial-distance signature in other studies. This or similar distance signatures are often used as a first step in building shape descriptions using the Fourier Transform, see (Zhang and Lu, 2005; El-ghazal et al., 2009) for reviews. But here, the radial signature itself is analyzed and is parameterized based on its extrema and other geometric information. Some of the parameters are simple shape descriptions as used previously (Peura and Iivarinen, 1997).

For a given isocontour, its pole is determined by taking the mean of all curve points. A radial signature $R(v)$ is formed, whereby v is the arc length variable; the signature is normalized by the average radius. The signature's extrema are determined and assigned to lists $R^{\max}$ and $R^{\min}$(maxima and minima respectively), with $\mathsf{n}_{\max}$ the number of maxima. The directional angles $\gamma(i)$ of the maxima are taken and their included angles $\alpha(i)$ between them determined and ordered by decreasing value ($i = 1..\mathsf{n}_{\max}$).

Five structural biases were created, which represent to what degree the shape corresponds to a specific (global) form; $\breve{s}_{\frown}$ represents curved shapes and may correspond to the bending energy (Peura and I-ivarinen, 1997); $\breve{s}_2$ coarsely represents the degree of elongation (or eccentricity in (Peura and Iivarinen, 1997)), an analogue measure to the aspect ratio; $\breve{s}_3$ represents triangle or deltoid shapes; $\breve{s}_4$ stands for a quadrilateral or astroid shape; $\breve{s}_5$ is for a pentagon or any star shape with five peaks. Thus, the bias number corresponds to the number of 'outer' corners (vertices) with angles smaller than π and we refer to those shapes sometimes as n-corner shapes.

For notational simplicity we use the following angles: supplementary angles α^c as a π-complement to the included angles α; interior angles $\beta^3 = \frac{2\pi}{3}$, $\beta^4 = \frac{\pi}{2}$, $\beta^5 = \frac{2\pi}{5}$ for expressing the degree of structural bias - they correspond to the interior angles for an equilateral triangle, a square and a regular pentagon.

- **Curved**, $\breve{s}_{\frown}$: the bias allows to express whether a shape is of type bean, bicorn, D or crescent and is only larger zero if the largest included angle $\alpha(1)$ is a reflex angle (larger than $\frac{\pi}{2}$). The bias corresponds to the difference above π:

$$\breve{s}_{\frown} = \begin{cases} \alpha(1) - \pi & , \alpha(1) > \pi \\ 0 & , \text{else.} \end{cases} \tag{28}$$

- **Two-corner**, $\breve{s}_2$: this bias is suitable to express shapes such as ovals, ellipses or U-turns. The bias is taken only if two or more maxima are present; its value depends on the complementary angles for the included angles, and is proportional to the elongation η and weight w:

$$\breve{s}_2 = \begin{cases} \frac{\pi - \alpha^c(1) - \alpha^c(2)}{\pi} w\eta & , \mathsf{n}_{\max} \geq 2 \\ 0 & , \text{else.} \end{cases} \tag{29}$$

The weight (or strength) equals the range of values for the 2nd derivative of the (normalized) radial signature: $w = \text{rng}(R'')$. The weight so corresponds to the degree of 'peakness' of the shape; the elongation is a parameter by itself and is defined later.

- **Three-corner**, $\breve{s}_3$: expresses triangles or star shapes with three peaks. The bias decreases with inreasing asymmetry from an equilateral triangle (or deltoid or equivalent star shape):

$$\breve{s}_3 = \begin{cases} \frac{\beta^3 - \sum_{i=1}^{3}(\beta^3 - \alpha(i))}{\beta^3} w & , \mathsf{n}_{\max} \geq 3 \\ 0 & , \text{else.} \end{cases} \tag{30}$$

- **Four-corner**, $\breve{s}_4$: the bias corresponds to the ratio of the fourth included angle and the interior angle for a square:

$$\breve{s}_4 = \begin{cases} \frac{\alpha(4)}{\beta^4} w & , \mathsf{n}_{\max} \geq 4 \\ 0 & , \text{else.} \end{cases} \tag{31}$$

- **Five-corner**, $\breve{s}_5$: the bias is analogous to the one for the four-corner polygon:

$$\breve{s}_5 = \begin{cases} \frac{\alpha(5)}{\beta^5} w & , \mathsf{n}_{\max} \geq 5 \\ 0 & , \text{else.} \end{cases} \tag{32}$$

In summary, the corner biases have a large value for cyclic polygons (equilateral triangle, square, rectangle, regular pentagon,...). The bias decreases the more the shape deviates from the interior angle of the cyclic polygon. The bias also decreases the less 'acute' the interior angles are such as in a squircle (four-cornered wheel). Conversely, the bias grows very large if the corners become more acute such as in a hypocycloid (deltoid, astroid,...) or any star shape - due to the use of the derivative-dependent weight value w. A circle is expressed by a zero value for all structural biases.

- **Elongation**, η: measures the spatial extent of the shape and is 0 for symmetric shapes such as circles, squares and pentagons; it is proportional to the range (rng) of radii otherwise:

$$\eta = \begin{cases} \text{rng}(R(v)) & , \breve{s}_2 > 0 \\ \max(\text{rng}(R^{\max}), \text{rng}(R^{\min})) & , \mathsf{n}_{\max} \geq 3 \\ 0 & , \text{else.} \end{cases} \tag{33}$$

For polygons with three or more corners, the range of the signature values is not a sufficient elongation measure, because an equiangular polygon (or in particular a hypocycloid) shows a positive elongation value due to its corners but does not contain a true elongation. The elongation is therefore determined with the lists of maximum and minimum radii.

• **Symmetry**, y: is determined by two measures. One is the signature's irregularity, ι (iota), which is the integrated difference between the two signature halves whereby one half is inverted, whereby the point of havening is the total maximum of the radii $R^{\max}$. The other measure is the minimum of the (absolute) derivative for the (ordered) maximum radii:

$$y = 1 - \iota - \min(\Delta R^{\max}). \tag{34}$$

The symmetry is largest for any shape, that has at least two equal angles such as an isosceles triangle, a kite shape and so on.

• **Gap Size**, $\omega_d, \omega_\angle$: the gap size - if present - is included as dimensions as well by the spatial and angular separation between the endpoints, ω_d and $\omega_\angle$. The angular separation may be zero but there may exist a spatial separation, in which case the curve corresponds to the beginning of a spiral shape.

• **Concavity**, $\epsilon_{\rtimes}$ The concavity bias is determined by observing the difference between maxima's halfpoints and the radii minima $R^{\min}$. The halfpoints are taken for the neighboring maxima points and their radii R_i^{mxh} to the pole measured. From those, the minima radii are subtracted and only the positive differences are integrated:

$$\epsilon_{\rtimes} = \tanh\Big(\sum_i \max(R_i^{\text{mxh}} - R_i^{\min}, 0)\Big) \tag{35}$$

The value is zero if the shape is convex and increases with concavity.

additional parameters are the degree of *concavity*, which is determined by the fraction of the signature whose course is in reverse direction to the dominant direction; the above-mentioned weight value w; the standard deviation of the angular directions of the corners; the angular gap, which is above zero in case of a horse-shoe shape; the mean, minimum, maximum and standard-deviation value for the maxima and minima radii ($R^{\max}$ and $R^{\min}$); and the mean, minimum, maximum and standard-deviation value for the maxima and minima in the second derivative.

The entire parameterization is of relatively low complexity $O(NK)$, with N the number of isocontours and K the number of operations to arrive at the variety of structural parameters.

Implementation and Selection Isocontours from different spatial (image) scales were extracted. For each one the concavity measure is determined and isocontours with large concavity values discarded. No other selection takes place - in contrast to pair or quad formation - as there is no combinatioral complexity involved.

Algorithm 4 Generating shape vectors.

Input : $\mathcal{I}$, list of isocontours
Output: $\mathbf{f}_l$, list of shape vectors
1) Determine concavity for all isocontours and select low-concavity ones $\rightarrow \mathcal{F}$.
2) Generate vector $\mathbf{f}$ for each element $\in \mathcal{F}$ (eq. 36)

Vector Formation The following vector is then formed:

$$\mathbf{f}(o, r, \eta, y, \omega_d, \omega_\angle, \breve{s}_\frown, \breve{s}_2, \breve{s}_3, \breve{s}_4, \breve{s}_5, \{apps\}), \tag{36}$$

whereby o is the angular orientation of the global maximum's directional angle; r is the average of the (unnormalized) signature, normalized here by the image dimensions.

Figure 8 shows the results of a retrieval task where different biases are preferred to illustrate that the vector space is a good approximation to the various shapes.

5 Clusters

Two types of clusters are developed, clots and bundles. Clots are clusters of short segments which frequently occur in texture or at intersections of object parts such as T or X junctions. Bundles are clusters of long segments and specifically aim at describing a group of parallel aligned segments. Pairs express the detailed geometrical alignment of two segments.

5.1 Clots

Clots are formed in two phases: first, clusters of (short) segments are identified using a traditional cluster method; then, the alignment of the clusters' segments is characterized.

1) Identification Clots are identified by a hierarchical cluster analysis of the segments' midpoints, see algorithm 5. The pairwise distances D_p between the (short) segments' midpoints is taken ($p = 1,..\mathsf{n}_{\text{Pairs}}$) and their nearest neighbor distances determined D_j^{NN} ($j = 1,..,\mathsf{n}_{\text{Short}}$; $\mathsf{n}_{\text{Short}}$ the number of sort segments). Hierarchical clustering is performed with the pairwise distances D_p using an average linking method and a cutoff distance whose value is twice the average of the nearest neighbor distances, $\frac{2}{\mathsf{n}_{\text{Short}}} \sum_j D_j^{NN}$. Only clusters of 3 or more segments

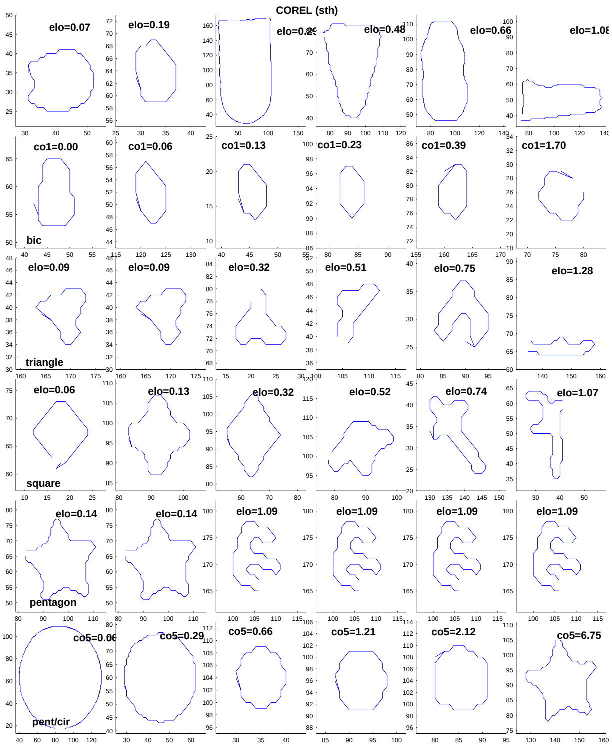

Figure 8: Sorting shape vectors along different dimensions (aspects). In each row one dimension value was systematically increased (in the upper right of each subplot the actual and not the preset value is displayed). Top row: increasing (inc.) elongation (η) value. 2nd row: inc. η value for a two-corner polygon 2; 3rd row: inc. η value for a three-corner polygon. 4th row: inc. η value for a five-corner polygon. 5th row: inc. η value for a circular shape.

are retained. This clustering procedure generates several tens of clusters per image, with each cluster consisting of a list of segments, see figure 9 for an example.

2) Alignment The alignment analysis focuses in particular on radial and orientation statistics. For the radial statistics, the cluster center, now called pole, is determined to be the average of the midpoint coordinates. The radii between the pole and midpoint coordinates is taken, $R_l, l = 1, .., \mathsf{n}_{\text{cs}}$, with n_{cs} the number of segments; the following measures were defined: the minimum radius r_{min}, the maximum radius r_{max}, the mean radius r_μ, the standard deviation of the radii r_σ. From the array of segment lengths the mean and standard deviation is taken, l_μ and l_σ resp.

In the orientation analysis, three principal types of alignments are determined: uni-orientationality, where segments have similar orientation; null-orientationality, where segment orientations cover roughly the entire range; cross-orientationality, where segments are aligned in a T or X pattern. To discriminate between these alignments a 4-bin histogram $H(o)$ is generated, whose bins are centered at 0, 45, 90 and 135 degrees.

Algorithm 5 Clot formation.

Input : $\mathcal{S}$, list of segments
Output: $\mathbf{c}_k$, list of clot vectors
1) Select short segments using median length value $\rightarrow \mathcal{S}^{short}$
2) Pairwise midpoint distances $D_p, \forall \in \mathcal{S}^{short}$
3) Hierarchical clustering with D_p and corresponding NN distances D_j^{NN}
4) Linking with average method and cutoff distance $t = \frac{2}{\mathsf{n}_{\text{Short}}} \sum_j D_j^{NN}, \rightarrow \mathcal{C}^{all}$
5) Select clusters with ≥ 3 segments: $\mathcal{C}^{all} \rightarrow \mathcal{C}^r$
Generate vector $\mathbf{c}$ for each clot $\in \mathcal{C}^r$ (eq. 40)

• **Uni-Orientationality**, $o^{\shortmid}$: measures the degree of same orientation by dividing the histogram's peak amplitude by the number of cluster segments

$$o^{\shortmid} = \max_o(H(o))/\mathsf{n}_{\text{cs}} - \tfrac{1}{4}, \tag{37}$$

whereby we subtract $\frac{1}{4}$ to account for a flat distribution. The measure does not account for unequal segment lengths.

• **Null-Orientationality**, o^* : expresses how distributed the orientations are and is maximal if all orientation occur with equal frequency in which case the value is 1,

$$o^* = 2 - \frac{\max_o(H(o))}{\mathsf{n}_{\text{cs}}/4}, \tag{38}$$

otherwise it decreases with increasing 'peakness' in the histogram. Negative values are set to zero.

• **Cross-Orientationality**, o^+ : expresses groups with two principal orientations, e.g. T or X alignments, which we can recognize by determining whether the histogram is bimodal. The measure is the ratio between the two maximal peak values

$$o^+ = \begin{cases} \frac{\max_o^2(H(o))}{\max_o(H(o))} & , H \text{ is bimodal} \\ 0 & , \text{otherwise.} \end{cases} \tag{39}$$

whereby $\max^2$ denotes the amplitude of the second mode.

If the value for uni-directionality is larger than the other two values o^* and o^+, then we set the cluster's dominant orientation angle $\hat{o}$ to the orientation angle o of the longest segment, otherwise the value is not a number.

With the above attributes a descriptor vector is formed:

$$\mathbf{c}(r_{\min}, r_{\max}, r_\mu, r_\sigma, l_\mu, l_\sigma, \hat{o}, o^{'}, o^*, o^+, \{app\}). \quad (40)$$

where $\{app\}$ represents an array of simple appearance parameters as introduced in (Rasche, 2010).

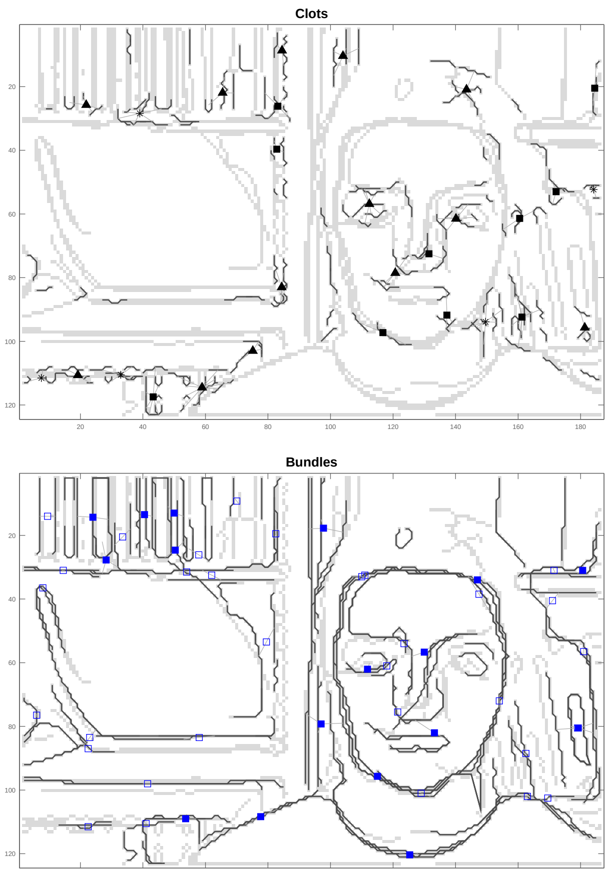

Figure 9: Example of clots and bundles for a face image of the Caltech101 collection (upper and lower graph resp). In both graphs the edge pixels are of light-gray luminance; the short and long contours of dark-gray luminance. Clots: cluster centers marked with squares, triangles and stars. Bundles: pairs marked with a square; groups of 3 or more marked with a filled square.

5.2 Bundles

Bundles are formed similar to clots, but with a focus on finding segments that lie approximately parallel. Pairwise distances D_p are again taken with the segment midpoints, but a selection of pairs is made, based on the pairwise length and bendness differences, see algorithm 6 for details:

Algorithm 6 Bundle formation.

Input : $\mathcal{S}$, list of segments
Output: $\mathbf{b}_k$, list of bundle vectors
1) Select long segments using median length value $\rightarrow \mathcal{S}^{long}$
2) Pairwise midpoint dists $D_p^m, \forall \in \mathcal{S}^{long}$
Pairwise length differences $D_p^l = \frac{l_1 - l_2}{l_2}$
Pairwise bendness differences $D_p^b = b_1 - b_2$
3) Exclusions: $D_p^m > l_2/2; D_p^l > 1; D_p^b > 1 : \rightarrow D_r$
4) Hierarchical clustering with D_r and corresponding NN distances D_j^{NN}
5) Linking with average method and cutoff distance $t = \frac{2}{\mathrm{n_{Short}}} \sum_j D_j^{NN}, \rightarrow \mathcal{B}^l$
6) Singleton clusters are excluded $\rightarrow \mathcal{B}^k$
Generate vector $\mathbf{b}$ for each bundle $\in \mathcal{B}^k$

A bundle vector $\mathbf{b}$ similar to the clot vector was formed, but which also included simplest bendness statistics of the segments:

$$\mathbf{b}(o, f, r_{\min}, r_{\max}, r_\mu, r_\sigma, l_\mu, l_\sigma, b_\mu, b_\sigma, \{app\}). \quad (41)$$

where o is the dominant orientation, f the segment frequency, b_μ the mean bendness and b_σ the standard deviation of the bendness values.

6 Textures

Texture descriptors are generated from regions that are outlined some descriptors. In particular, the regions from three types of descriptors are taken: from segments of minimal curvature; from pairs, whose segments are approximately vis-a-vis; from any shape. The regions are of varying size ranging from several pixels to large image patches. Motivated by the popularity of the SIFT features, we had tried a matching with histograms of intensity gradients - where gradients are taken from the region - , but it did not appear very fruitful. There may be two reasons: one is that due to the varying region sizes, it is difficult to find a meaningful distance measure, even if the histograms are normalized for region size; a second reason may be that many regions do not contain 'interesting' textures, as opposed to the ones centered around 'interest points' and thus a histogram of gradients does not properly capture the texture properties of our selected regions. We therefore use a parameterization of the output of various types of image preprocessing, called texture parameters hereafter. One could perform matching with only those texture parameters, but we had observed that if one includes some geometric parameters from the descriptor that outlines the region, that classification performance increases. We therefore append the texture parameters to the geometric parameters of the descriptors, whereby some geometric parameters are omitted. We thus generate three types of texture descriptors, arc texture $\mathbf{t}^{\mathsf{a}}$, pair texture $\mathbf{t}^{\mathsf{p}}$ and shape texture $\mathbf{t}^{\mathsf{s}}$, which all share the same set of texture parameters, but distinguish themselves from some geometric parameters that can be regarded as silhouette parameters. We firstly elaborate on the types of image preprocessing we use (subsection 6.1). Then we specify what parameters are extracted from each region (subsection 6.2). Finally we describe how regions are selected and what type of silhouette parameters are used for building the region vectors (subsection 6.3) .

6.1 Image Preprocessing

The image is analyzed in four different ways:

Intersection Count: The locations where contours intersect are detected and for each location the number of intersecting contours is determined. This is done with the contour image - the binary image with edgel information: it is convolved with a 3x3 summation mask resulting in an intersection map, whose conspicuous values are: 1, contour terminates; 4, 5 and 6: 3, 4 and 5 contours intersect, respectively. This intersection detection is done for all scales at which contours are extracted, typically σ=1,2,3 and 5.

Difference-of-Gaussians (DOGs): Three mask sizes are used: 3, 5 and 7 pixels with standard deviation $\sigma = \frac{1}{2}, \frac{5}{6}, \frac{7}{6}$ for the center Gaussian and $\sigma = \frac{3}{4}, \frac{5}{4}, \frac{7}{4}$ for the surround Gaussian. Positive and negative DOGs are formed, totaling thus 6 DOG maps.

Blobs: Blobs are defined as thresholded DOGs in order to detect true center-surround patches. The same sizes as for DOGs are used and the mean of the center is required to exceed or undershoot the positive and negative DOGs respectively, resulting in 6 blob maps.

Topology: The intensity landscape is characterized according to 3 different, basic topologies. At each pixel, the local neighborhood is analyzed for its degree of rampness, foldness and uneveness. Rampness describes whether the intensity patch is an incline. Foldness describes whether the patch contains a ridge or ravine. Uneveness describes an irregular surface. These topologies can be estimated by a mode analysis of the gradient histogram of the patch. For rampness, the gradients point into a similar direction and the histogram therefore is unimodal with a high amplitude. For foldness, the gradients point into two directions and the histogram is therefore bimodal. For uneveness, the gradients point into all directions and the histogram has a low amplitude.

6.2 Region Parameterization

For each selected region, two sets of parameters are generated, appearance parameters and topology parameters, whereby this distinction is made to emphasize the novelty of the latter set. The former set represents traditional measures of 'appearance' and are structurally not very specific, whereas the latter are novel and structurally more explicit.

Appearance: area a of the region; standard deviation σ_r of the region's intensity values; range (contrast) c_r of intensity values; the count of intersections for each conspicuous value, i_1, i_4, i_5 and i_6; average values are formed for the values for each DOG and blob map, $\{\delta_1, .., \delta_6\}$ and $\{\beta_1, .., \beta_6\}$ respectively:

$$\text{txt}^{\mathsf{app}} = \{a, \sigma_r, c_r, i_1, i_4, i_5, i_6, \delta_1, .., \delta_6, \beta_1, .., \beta_6\} \tag{42}$$

Topology: average fold value t^{f}; degree of elevation $\hat{t}$of the region's intensity landscape, which is defined as the positive intensity difference between the center values and the surround values - it is quasi a difference-of-regions, where the inside is obtained after some erosion of the region and the outside is the difference between the region and the inside; degree of sinkness $\breve{t}$, which is the negative equivalent of the degree of elevation; average uneveness value $\tilde{t}$; average rampness value t^{r}:

$$\text{txt}^{\mathsf{top}} = \{t^{\mathsf{f}}, \hat{t}, \breve{t}, \tilde{t}, t^{\mathsf{r}}\} \tag{43}$$

The two vectors are concatenated and appended to some geometric values describing the regions silhouette, which is explained next.

6.3 Selection and Vectors

For all three texture types (segment, pair or shape) a minimal area is required. For segments, arcs with a minimal bendness and a maximal transition value are selected. The segment region vector consists of selected geometric parameters (orientation, length, arc, bendness,...see (Rasche, 2010)) and the texture parameters introduced in the previous subsection:

$$\mathbf{t}^{\mathsf{a}} = (o, l, a, b, ..., \{\text{txt}^{\mathsf{app}}\}, \{\text{txt}^{\mathsf{top}}\}) \quad | \quad a > 0, t < 0.2, \tag{44}$$

Pairs, whose segments are approximately parallel, vis-a-vis and reasonably proximal are preferred. This preference is obtained by omitting pairs with a positive value for the T-bias, the collinearity bias and the intersection bias; and pairs with a low proximity value. A pair region vector $\mathbf{t}^{\mathsf{p}}$ is generated with geometric parameters omitting the biases that had been excluded; the same texture parameters are added as for the segment region vector (eq. 44). A shape region vector $\mathbf{t}^{\mathsf{s}}$ is generated analogously. Algorithm 7 summarizes the process.

Algorithm 7 Selecting regions and their texture parameters.

Input : $\mathcal{S}$, $\mathcal{P}$, $\mathcal{F}$, list of segs, pairs and shapes
Output: $\mathbf{t}^{\mathsf{a}}$, $\mathbf{t}^{\mathsf{p}}$, $\mathbf{t}^{\mathsf{s}}$
1) Image preprocessing: → intersection map, DOG map, blob map, topology map
2) Select appropriate regions from $\mathcal{S}$, $\mathcal{P}$and $\mathcal{F}$ and some of their geometric parameters (silhouette geometry)
3) Generate texture parameters for each region $\text{txt}^{\mathsf{app}}$ and $\text{txt}^{\mathsf{top}}$ from the maps
4) Create vectors $\mathbf{t}^{\mathsf{a}}$, $\mathbf{t}^{\mathsf{p}}$, $\mathbf{t}^{\mathsf{s}}$ consisting of texture and silhouette parameters (e.g., eq. 44)

7 Implementation

The entire system was implemented in Matlab. Contours are extracted with the Canny algorithm at four different spatial scales ($\sigma = 1, 2, 3, 5$) and then partitioned into segments and described as in (Rasche, 2010). A typical image contains several hundred segments, see figure 10, right column, descriptor labeled 'Seg'. For each segment, a number of appearance parameters (contrast, fuzziness,...) are extracted, which were based on simple luminance statistics, see (Rasche, 2010) for details. Those are abbreviated as $\{apps\}$ in the vector equations 20, 27, 36, 40 and 41. The total duration for generating all descriptors is 30 seconds per image on a 3.33GHz processor (see also table 3), of which half of it is taken by the grouping processes. A total of 278 parameters is generated, see lower left in figure 10; the right column also shows that the majority of descriptors are pairs and quads. As most attributes are defined in unit range, only some dimensions required normalization. The variances (left column, figure 10) show that the statistics for the collections appear the same.

8 Classification

There is a variety of classification methods that one can use for descriptions with unequal list lengths. A simple and successful principle is to search for clusters in the entire collection, in which case all lists are concatenated; a representative example would be clustering with a k-means procedure followed by a classification of the detected clusters and this is generally known as the bag-of-words approach, e.g. (Sivic and Zisserman, 2003; Perronnin et al., 2006; Philbin et al., 2007). A slightly more refined classification method is the search for unique features such as carried out by an adaboost technique - also performed on a concatenated list; the most famous example is the face

Table 3: Description and representation summary. Desc: descriptors; Att: attributes; CatEnsemb: category ensemble. n_{cat}: number of categories. Duration determined on a 3.33GHz processor.

Total No Attributes	278
Grouping Complexity	$O(N^2)$, N = keypoints
Grouping Duration Tot Dur Desc Extraction	15 sec (3.33GHz) 30 sec
Size/Image ($\approx 250\times250$ pix): No Desc × No Att	 $\approx 4000 \times 31$
Dim Img Vect → after PCA	$2780 \rightarrow \approx 2 \cdot \mathsf{n}_{\text{cat}}$
Size/CatEnsemb No HypPlanes × No Att → after PCA	 $\approx 90 \times 31$ $\rightarrow \approx 20 \cdot \mathsf{n}_{\text{cat}}$

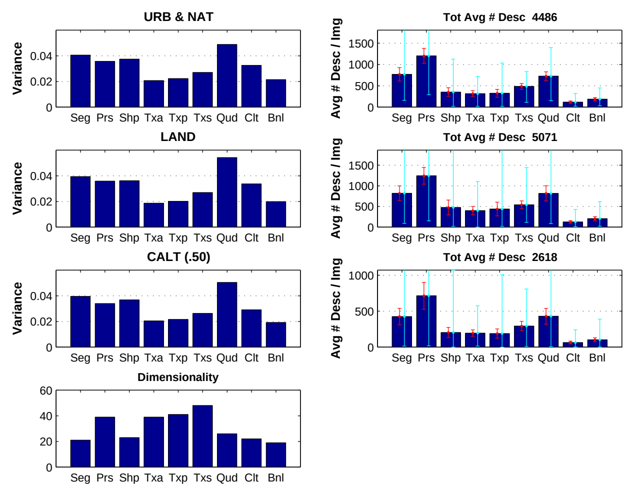

Figure 10: Descriptor variance and count per image for three collections (**first three rows**). The descriptor variance is the average across its dimensions variances. The descriptor count is per image; the small errorbars denote the inter-image variation; the large errobars denote the minima and maxima. **Lower left**: dimensionality (no. of attributes) per descriptor for comparison.

detection system by Viola and Jones (Viola et al., 2005). And there can be many combinations of the individual steps of these classification principles.

We had tried a clustering by the k-means method, but the classification accuracy was moderate only. The reason maybe that this type of quantization is too unspecific for our type of descriptors; which possess much lower dimensionality (up to ca. 40) as opposed to the typical 128-dimensional SIFT or other related descriptors (Lowe, 2004). The method of features selection using Adaboost turned out to be more succesful (subsection 8.2). But we will also report accuracies with a simple statistical classifier using only image vectors (subsection 8.1) with which one can obtain quick yet still reasonable results.

8.1 Image Vector Classification

For a list of descriptors $\mathbf{a}_i(d)$ we generated a 10-bin histogram H_d for each attribute d ($d = 1, .., \mathsf{n}_{\mathrm{Dim}}$, number of dimensions). The attribute histograms for different descriptor types were then concatenated to form a high-dimensional image vector H^I, whose size could be several hundred components. Thus, there is no use of the multi-dimensionality of the individual vectors per se; the histogram is a mere statistical description of the descriptor 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.

8.2 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 (Schapire and Singer, 2000; Breiman, 1998).

Viola and Jones introduced this methodology to the computer vision community with their rapid face detection system (Viola et al., 2005). They specifically used single-node decision trees (decision stumps) as weak classifiers, but applying such decision stumps to individual dimensions did not yield good accuracies 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.

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

$$t(v, p, \theta) = \begin{cases} 1 & , \ v > p\theta \\ 0 & , \ \text{else}, \end{cases} \tag{45}$$

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,

$$h(\mathbf{a}, \mathbf{p}, \mathbf{\Theta}) = \sum_{d=1}^{\mathsf{n}_{\mathrm{Dim}}} t\big(\mathbf{a}(d), \mathbf{p}(d), \mathbf{\Theta}(d)\big), \tag{46}$$

where $\mathbf{\Theta}(d)$ 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

$$c(\mathbf{a}, \mathbf{p}_l, \mathbf{\Theta}_l) = \sum_{l=1}^{\mathsf{n}_{\mathrm{L}}} h(\mathbf{a}, \mathbf{p}_l, \mathbf{\Theta}_l)\mathbf{w}(l), \tag{47}$$

whereby weights $\mathbf{w}$ 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 $\mathsf{n}_{\mathrm{Desc}}$ descriptors of an image,

$$P = \sum_{i=1}^{\mathsf{n}_{\mathrm{Desc}}} c(\mathbf{a}_i, \mathbf{p}_l, \mathbf{\Theta}_l)/\mathsf{n}_{\mathrm{Desc}}. \tag{48}$$

We used two types of classification. One is used during learning, which occurs by taking the maximum value across the posterior values $\mathrm{argmax}_k P_k$. 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 better accuracy; to clarify, the length of the 'feature' vector for classification was the number of descriptor types times the number of categories $V = \{P_1^s, P_2^s, .., P_{\mathsf{n}_{\mathrm{K}}}^s, P_1^p, P_2^p, .., P_{\mathsf{n}_{\mathrm{K}}}^p, ...\}$ (s=segment, p=pair, etc.). This means that how an image responded to other category representations was valuable information to obtain better discrimination.

Adaptive Boosting In the adaptive boosting learning procedure, a sample is reused in a weighted manner according to the evolving classification accuracy 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 45 - 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.

9 Evaluation

9.1 Image Collections

Evaluation took place on three different image collections, the Urban&Natural collection (Oliva and Torralba, 2001), the Caltech101 (Li et al., 2006) and the Landuse collection (Yang and Newsam, 2010). 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 (Oliva and Torralba, 2001). 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 (Kapoor et al., 2010) for a summary). The Landuse collection consist of satellite images depicting 21 categories (Yang and Newsam, 2010) (street intersections, forest, agricultural fields,...) and were correctly classified with ca. 81 percent by a bag-of-features approach, with features being SIFT features (Yang and Newsam, 2010).

9.2 Image Vector Classification

With a 6-fold cross validation we reached 75 percent correct classification for the Urban&Natural collection, 77 percent for the Landuse collection and 40 percent for the Caltech101 collection, see horizontal lines in figures 11 and 12; the third row in table 4 summarizes the classification accuracies with the image vector. The contribution of the individual descriptors was estimated once with a classification in which a single descriptor was knocked out (figure 11) and once with a classification using an individual descriptor only (figure 12).

A knockout performance below or above the performance for full-dimensionality (solid line) stands for a

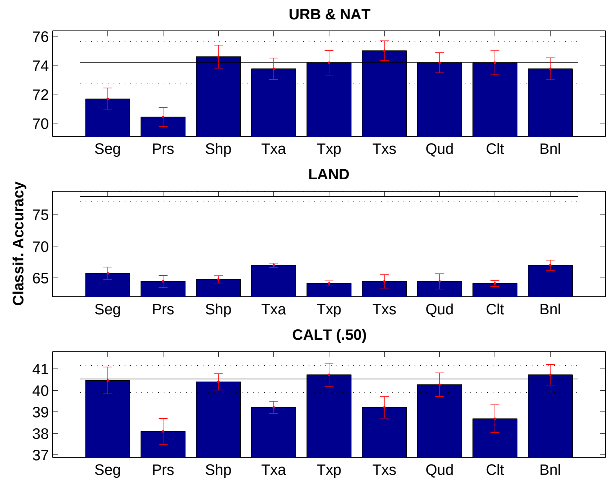

Figure 11: Classification accuracy when single descriptors are knocked out in the image vectors (Urban&Natural, Landuse, Caltech101). Horizontal solid and dotted lines represent performance for full dimensionality plus standard error for 6 folds.

more or less significant descriptor, respectively (figure 11). For the Urban&Natural and the Caltech101 collection there are descriptors that contribute significantly, e.g. the pair descriptor, or the arc descriptor for the Urban&Natural collection; for the landuse collection, all descriptors contribute significantly.

Looking at the individual-descriptor performance (figure 12), we observe that even individual performances almost reach the performance for full dimensionality for the Urban&Natural collection, but are are significantly lower for the other two collections.

As the descriptors occur with different count per image and have different dimensionality (figure 10), one can analyze how expressive the individual descriptors are. We define the 'expressiveness' as the classification accuracy - as shown in figure 12 - divided by the number of descriptor dimensions (attributes) and the number of descriptors. Figure 13 shows that the clot and bundle descriptors exhibit the largest degree of expressiveness.

9.3 Ensemble Classification

With ca. 10 learning steps (hyperplanes, n_L), we obtained a classification accuracy of ca. 81 percent for the Urban&Natural collection - for fewer or more learning steps the accuracy 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

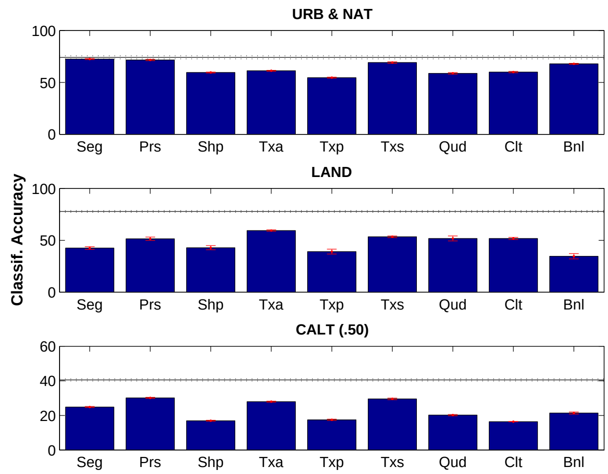

Figure 12: Classification accuracy for individual descriptors only (as image vectors).

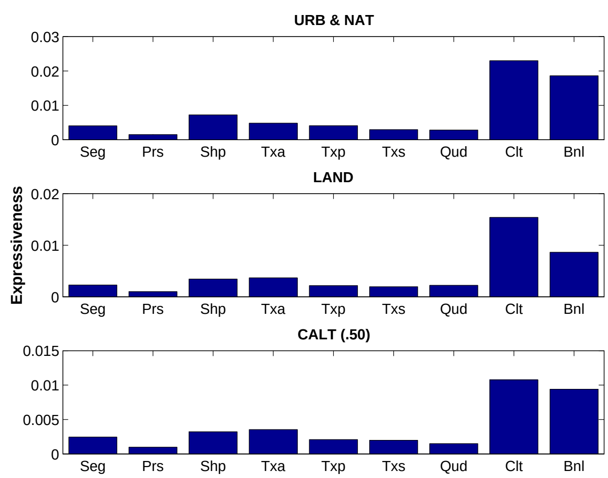

Figure 13: Descriptor expressiveness. Classification accuracy divided by dimensionality and descriptor count.

dimensionality of ca. 56 dimensions to achieve a correct classification of 82 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. 50 percent.

The fourth row in table 4 summarizes the classification accuracies.

9.4 Varia

We have not tested the contribution of the individual attributes of each descriptor type as it is computationally too demanding to carry out a feature selection procedure wrapped around the slow adaboost learning procedure. It may well be that some parameters are insignificant in one or the other image collection. But in general, the addition of parameters always improved performance. This is most evident for the descriptor knock-out for the landuse collection (figure 11), where a decrease in categorization performance by more than 10 percent occured for every omitted descriptor; it is therefore likely that the majority of individual attributes contributes to discrimination of categories in that collection. The covariance matrices for the dimensions (attributes) of each descriptor generally do not show any strong dependencies.

Table 4: Summary of classification accuracy. **Bmk**: benchmark; **Img Vect**: PCA/LDA on image vectors; **Cat Rep Desc**: hyperplanes (decision stumps) on desciptor spaces. All percentages are rounded.

Collection	Bmk	Img Vect	Cat Rep Desc
Urb&Nat	80%	75% (-5%)	81% (+1%)
Landuse	81%	77% (-4%)	82% (+1%)
Calt101	70%	40% (-30%)	50% (-20%)

10 Discussion

Some of the group types we have tested correspond to groups that had been proposed previously, namely pairs of segments and clusters of parallel segments - the latter are called bundles in our case. What is completely new is that we carry out an exhaustive geometric characterization for the purpose of matching. Other group types are completely new, such as the quad descriptor - pairs of pairs - and the shape descriptor. The clot descriptor is a mixture of a structural and a textual descriptor. The presented geometric parameterization can be regarded as an example of how an alignment of segments can be characterized. While the pair descriptor is meanwhile relatively well developed and its parameterization occurs relatively elegant by correlating symmetry measures (table 2), the other descriptors may certainly profit from refinement and further testing.

It was shown that this structural information can be exploited in a classification task. The classification accuracies are comparable to other systems. The classification with image vectors already showed respectable results and in case of the Urban&Natural and the Landuse collection, the results are near the benchmark; this type of classification has the advantage that it occurs fast. When using an ensemble classifier, the accuracies increase to the level of some benchmarks: the benchmarks for the Urban&Natural and Landuse collection were just exceeded; the one for the Caltech101 collection still lags somewhat.

The results with knock-out and individual descriptors (figures 11 to 13) have to be interpreted with care as those were obtained with image vectors only. As pointed out, such an analysis for the slow ensemble classifier is unfeasible. A feature selection based on those results may be beneficial if one intended to optimize the classification with image vectors, but for classification with ensembles exploiting the descriptor space, this may be the wrong direction for optimization. Nevertheless, we dare the following interpretations: clusters of segments, such as the clot and bundle descriptors, show a high degree of expressiveness as they are geometrically not as precise as other descriptors and thus more flexible to express category variability. This does not need to be an indication that the other descriptors should be less specific in their geometric description, as it is exactly that specificity which may be the key to increase the separation between categories. Only extensive testing and analysis can shed more light on the expressivness of the individual descriptors.

Comparison to Gradient Histogramming As classification with histograms of intensity gradients is so dominant and successful, we draw further comparison to that methodology, in particular on the aspects of descriptor generation, learning and matching duration, as well as representation size:

- Descriptor generation: the generation of histograms with intensity gradients is of low complexity. Our descriptor generation is substantially more complex: the most time consuming part is the curve partitioning process and is necessary to understand the exact 'bends' that a contour exhibits: the method has therefore the advantage that the contour segments can be immediately used for interpretation of object outlines. The duration for the grouping operations presented in this study is smaller than the one for the partitioning process, as the pairwise analysis of segments relies only on its keypoints.

- Matching duration: matching with gradient histograms can be relatively fast if indexing structures (e.g. kd-trees) are used, something which could also be tested with our descriptors in principle. But since our category representation is relatively small, the matching duration is negligible.

- Learning duration: in the gradient-histogramming approach often employs a k-means clustering algorithm to create a bag-of-words representation, a procedure of $O(N)$ complexity with N the number of descriptors. The duration in our system is not immediately comparable, but we estimate it of similar complexity as no actual distance measurements are taken in our learning scheme.

- Representation size: in the gradient-histogramming approach, image descriptions can consist of several thousands vectors. Our image description is equally vast, but our dimensionality is less than half the one of the gradient vectors taking the original histograms of dimensionality 144 as a comparison (Lowe, 2004). Recent approaches are able to reduce this dimensionality substantially and may thus be comparable to our representation in size. Our category representations are however much smaller, consisting of only tens of stumps (decision planes).

The comparison to systems classifying with gradient histograms can be put succinctly as follows: our system requires much more preprocessing, but the category representations are relatively small and classification can therefore occur relatively quickly.

References

Blum, H. (1973). Biological shape and visual science .1. *Journal Of Theoretical Biology*, 38(2):205–287.

Borra, S. and Sarkar, S. (1997). A framework for performance characterization of intermediate-level grouping modules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1306–1312.

Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, 26(3):801–849.

El-ghazal, A., Basir, O., and Belkasim, S. (2009). Farthest point distance: a new shape signature for fourier descriptors. *Signal Processing: Image Communication*, 24:572–586.

Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51.

Ferrari, V., Jurie, F., and Schmid, C. (2010). From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303.

Halsted, G. (1896). *Elementary Synthetic Geometry*. John Wiley and Sons Inc, New York.

Iqbal, Q. and Aggarwal, J. (2002). Retrieval by classification of images containing large manmade objects using perceptual grouping. *Pattern Recognition*, 35:1463–1479.

Jacobs, D. (1996). Robust and efficient detection of salient convex groups. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(1):23 – 37.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2010). Gaussian processes for object categorization. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 88:169–188.

Li, F., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 28(4):594–611.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 60(2):91–110.

Lowe, D. G. (1985). *Perceptual organization and visual recognition*. Kluwer Academic Publishers, Boston.

Mohan, R. and Nevatia, R. (1992). Perceptual organization for scene segmentation and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:616–635.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.*, 42(3):145–175.

Perronnin, F., Dance, C., Csurka, G., and Bressan, M. (2006). Adapted vocabularies for generic visual categorization. In *Proceedings of the 9th European conference on Computer Vision - Volume Part IV*, ECCV'06, pages 464–475, Berlin, Heidelberg. Springer-Verlag.

Peura, M. and Iivarinen, J. (1997). Efficiency of simple shape descriptors. In *Third International Workshop on Visual Form*, pages 443–451, Capri, IT.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Rasche, C. (2010). An approach to the parameterization of structure for fast categorization. *International Journal of Computer Vision*, 87:337–356.

Renninger, L. and Malik, J. (2004). When is scene identification just texture recognition? *Vision Research*, 44(19):2301–2311.

Sarkar, S. and Boyer, K. (1993). Perceptual organization in computer vision: a review and a proposal for a classificatory structure. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(2):382 – 399.

Schapire, R. E. and Singer, Y. (2000). BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA. IEEE Computer Society.

Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161.

Willmer, P. (1990). *Invertebrate Relationships : Patterns in Animal Evolution*. Cambridge University Press, Cambridge.

Witkin, A. and Tenenbaum, J. (1983). On the role of structure in vision. In Beck, J., Hope, B., and Rosenfeld, A., editors, *Human and machine vision*, pages 481–543. New York: Academic Press.

Yang, Y. and Newsam, S. (2010). 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*, pages 270–279. ACM.

Zhang, D. and Lu, G. (2005). Study and evaluation of different fourier methods for image retrieval. *Image and Vision Computing*, 23:33–49.

Zhu, Q., Wang, L., Wu, Y., and Shi, J. (2008). Contour context selection for object detection: A set-to-set contour matching approach. In *Lecture Notes in Computer Science*.