

Artistic Line-Drawings Retrieval Based on the Pictorial Content

THOMAS HURTUT, Université Paris Descartes
 YANN GOUSSEAU, Télécom ParisTech
 FARIDA CHERIET, Ecole Polytechnique de Montréal
 FRANCIS SCHMITT, Télécom ParisTech

This article is dedicated to the late Francis Schmitt who sadly passed away at the end of this project. He was a friendly mentor and a generous colleague to whom we owe great inspiration.

In this article, a general framework for the retrieval of artistic line-drawings is introduced. It relies on the pictorial content, defined as a combination of the stylistic content and the visual features of the represented subject. First, we propose an automatic method for the extraction of stroke contours in line drawings, relying on a filtering of the level lines of images. Next, the radius of the drawing tool is estimated from these segmented strokes. This information then efficiently tunes the extraction of several geometric features, including the distribution of curvature, endpoints, junctions and corners of strokes. The efficiency of the proposed method is illustrated with several experiments on two classified databases of artistic line-drawings, and compared with an approach based on the curvature scale space (CSS). Retrieval experiments suggest that the proposed framework is able to handle the pictorial effect delivered by line drawings to a human observer.

Categories and Subject Descriptors: I.4 [Image Processing]: I.5.4 [Pattern Recognition]: Applications—*Computer vision*; J.5 [Computer Applications]: Art and Humanities—*Arts, fine and performing*

General Terms: Algorithms, Design, Documentation, Experimentation, Human factors, Measurement, Performance, Theory

Additional Key Words and Phrases: line-drawings, visual art analysis, image analysis

ACM Reference Format:

Hurtut, T., Gousseau, Y., Cheriet, F., and Schmitt, F. 2011. Artistic line-drawings retrieval based on the pictorial content. *ACM J. Comput. Cult. Herit.* 4, 1, Article 3 (August 2011), 23 pages.
 DOI = 10.1145/2001416.2001419 <http://doi.acm.org/10.1145/2001416.2001419>

1. INTRODUCTION

Cultural institutions have, over the last decades, implemented a policy of digitally safeguarding their artwork collections. One of the goals is to improve their accessibility to the public and to professionals through image repositories (e.g., the online graphical art image database of the Louvre contains

Authors' addresses: T. Hurtut, LIPADE, Université Paris Descartes, 45 rue des Saints Pères, 75006 Paris, France; email: thomas.hurtut@parisdescartes.fr. Y. Gousseau and F. Schmitt, LTCI CNRS, Telecom ParisTech, 46 rue Barrault 75013 Paris, France; F. Cheriet, LIV4D, Ecole Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3A7, Canada. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2011 ACM 1556-4673/2011/08-ART3 \$10.00

DOI 10.1145/2001416.2001419 <http://doi.acm.org/10.1145/2001416.2001419>

150,000 drawings [Louvre 2010]). Works of visual arts are quite distinct from natural images in that they are often stylized. This distinction influences the understanding of the scene through possible ambiguities, as well as the visual effect delivered to an observer. Our visual attention toward an artwork is often based on some aesthetical considerations. For instance, “in an abstract painting, ideas, emotions, and visual sensations are communicated solely through lines, shapes, colors, and textures that have no representational significance” [Owen 2007]. In this article, we take interest in the automatic analysis of the *pictorial content* of line-drawings, defined as the combination of the artistic style and the visual features formed by the set of strokes composing the drawing. The visual effect considered here is strongly related to *preattentive* perception [Julesz 1986] prior to recognition by the visual system of the possible realistic subjects depicted in the artwork. The analysis of the pictorial content is useful to retrieve artworks delivering a similar visual effect to the observer [Eakins et al. 2004; Chen et al. 2005; Stanchev et al. 2006] and to explore large amounts of data. It is also of high interest to enable the transfer of some pictorial characteristics from real artworks to non-photorealistic rendered (NPR) drawings in Computer Graphics [Jodoin et al. 2002; Bae and Durand 2004; Grabli et al. 2004].

1.1 Problem Statement and Article Scope

The pictorial content is related to the artistic style. The style detection problem has been tackled in many references [van den Herik and Postma 2000; Li and Wang 2004; Yan and Jin 2006]. These approaches are usually based on a list of low-level features and a few definitions of some modern art movements such as cubism, impressionism, etc. The main problem is that artistic style inherits from many definitions. According to the *American Heritage Dictionary* [Pickett et al. 2000], style is “the combination of distinctive features of artistic expression, execution or performance characterizing a particular person, group, school or era.” It is worth noticing that both the style and the depiction of a given subject share many common visual elements. Style recognition is thus a very difficult task, requiring the knowledge of numerous art historians and experts. For instance, an artwork from the blue period of Picasso is recognizable not only because of its blue tonality, but because of many semantic features and iconographical cues. As long as computational methods do not succeed in extracting the representational content, style cannot be fully analyzed. The semantic gap is particularly important in visual arts images since artworks are often stylized. Moreover the depicted reality of artists is highly variable. We therefore limit our study to a less ambitious but more reachable task: the pictorial content analysis of artistic line drawings. Precisely, the analysis of line drawings is performed through low-level features extracted after a careful segmentation of stroke contours. Again, these features are inherited from both the represented subject and the artistic style. The choice to rely on stroke contours instead of any medial axis model will be discussed in Section 1.2.3.

Line-drawing practice has long been considered as one of the most important requirements in artist works [Kandinsky 1979; Klee 2004]. It is also often used for illustration in print publications. Sketch books often reveal useful information on an artist vision process. As revealed by a Clouzot documentary [Clouzot 1954], Picasso, for instance, starts the first layers of a painting with line strokes. The analysis of underlying preliminary drawings in paintings is also a topical subject [Kammerer et al. 2003]. Art history gives numerous examples of strikingly powerful line drawings based on a very few number of lines. Several past exhibitions focused on line drawings [Farine 2005; Labrusse and de Chassesey 2002].

This article makes the following contributions.

- We present an approach to detect closed contours using a level lines selection method. This selection method uses the probabilistic background of Desolneux et al. [2001] which is recalled in Section 2.1.
- Inspired by this framework, we propose a new algorithm, specifically suited to line-drawings, to filter the tree structure of the level lines (Section 2.2).

—We present a set of methods to extract and characterize the following geometrical features: curvature zero-crossings, corners, junctions, endpoints (Section 3).

Retrieval experiments on two classified databases of artistic line-drawings are used to demonstrate the efficiency of the proposed features (Section 4). We also compare our method to an approach which uses the curvature scale space (CSS) to detect geometrical points of interest. This article builds on a previous conference paper [Hurtut et al. 2008], providing detailed description of the strokes detection method and quantitative results.

1.2 Related Work

1.2.1 Automatic Analysis of the Pictorial Content of Artistic Line Drawings. The line content in artworks has been rarely investigated. Berezhnoy et al. proposed a method using a circular filter to extract brushstrokes which are then either interpolated by a third-degree polynomial function [Berezhnoy et al. 2005] or estimated by a multilevel thresholding [Berezhnoy et al. 2009]. These approaches are adapted to very small strokes such as Van Gogh's. Li and Wang [2004] studied ancient black and white ink drawings using wavelets and hidden Markov models. Their method is yet close to texture analysis in paintings such as in the approaches developed in Johnson Jr et al. [2008]. Onkarappa and Guru [2007] studied the spatial mutual arrangement of strokes in line-drawing images to achieve similarity retrieval.

1.2.2 Graphics Recognition. Analysis of technical drawings such as maps and engineering drawings has received considerable attention during the last decades [Tombre and Chhabra 1998]. Next open issues in this field are presently vectorization and symbol recognition of handmade sketchy drawings [Tombre 2006]. Presently, vectorization techniques are often applied to binary images and this problem is divided into the separated recognition of straight line and circular arcs which are the two mostly used shapes in technical documents. Interestingly, cultural heritage documents have been recently stressed as being one of the next emerging topic of interest by this community for indexing applications [Tombre 2006].

1.2.3 Cursive and Offline Writer Recognition. The neurobiomechanical process of writing is close to the one of drawing. Cursive alphabets are a kind of line drawings, and sometimes can be considered as such. Yet, writing inherits several geometrical constraints. Alphabets variability is bounded to a list of characters which have more or less the same size and spatial frequency. One of the biggest difficulties in writer recognition is the allograph variability. This means that a same letter can be written using a limited list of different symbols. Although artists may share some common representation techniques inherited from their visual arts education (e.g., perspective), they are at least since the 20th century renowned to break any pictorial convention. The line *alphabet* of an artist can thus intuitively be considered as much broader than letter alphabets. The cognitive process of drawing and writing are moreover quite different. Writing is a spontaneous task involving no continuous judgment of the written [Schomaker 1991] whereas an artist has to continuously analyze his drawing while doing it [Van Sommers 1984]. Nevertheless, prior to recognizing the objective content or characterize the subjective content in cursive handwritten document, several approaches choose to segment the strokes which is also our goal. Skeletonization [Lam et al. 1992] is classically used. Due to the lack of one unique mathematical definition and the need for some regularization, there exists hundreds of skeletonization methods. Skeletonization usually slightly distorts strokes [Pavlidis 1993]. These methods are moreover quite sensitive to local geometric variations. We thus choose to rely on stroke contours.

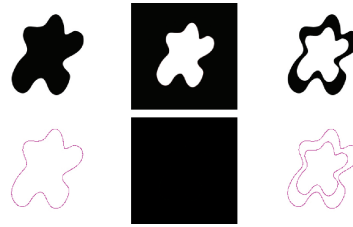


Fig. 1. Negative and positive level lines. Top row shows synthetic images and bottom row shows the level lines corresponding to the central object. Left, the contrast sign when going toward the interior of the level line is negative: this is a negative line. Middle, opposite situation, the level line is positive. Right, closed synthetic stroke made of two level lines which have opposite signs.

2. LEVEL LINES AND STROKE SEGMENTATION

Henceforth in the paper, a stroke made by the artist in a line artwork will be called a **stroke**. A set of $N \geq 1$ connected strokes crossing each others will be called a **stroke cluster**. Our first goal in this paper is to detect and represent the contours of such stroke clusters. Well-known local edge detection methods such as in Deriche [1987] create open contours which are unsuitable for our application. Methods using deformable models require more or less hazardous initializations and often impose strong regularity conditions on detected contours. We prefer to use an approach initially proposed by the mathematical morphology school and relying on level lines [Serra 1982; Caselles et al. 1999]. More precisely, we make use of several contributions on the computation [Monasse and Guichard 2000], filtering [Desolneux et al. 2001], and regularization [Moisan 1998] of level lines. The basic elements on which we build our analysis of line drawings are the *meaningful lines* as defined in Desolneux et al. [2001]. These are also the building blocks of several recent works on shape recognition [Musé et al. 2006; Cao et al. 2008]. The corresponding notions are recalled in Section 2.1. Then, in Section 2.2, we introduce an original filtering of the tree of level lines, specifically adapted to line drawings and permitting the segmentation of contours of stroke clusters.

2.1 Background on the Topographic Map and Meaningful Level Lines

2.1.1 The Topographic Map. Given an image u , the level lines are defined as the connected components of the topological boundary of the so-called level sets $\chi_\lambda(u) = \{x \in \mathbb{R}^2, u(x) \leq \lambda\}$, for all $\lambda \in \mathbb{R}$ [Serra 1982]. Level lines are closed curves except for the ones meeting the image borders. These lines can easily be closed by framing the image with an artificial margin of width 1 pixel and with gray value -1 . This trick is used in this paper, and henceforth all level lines will thus be considered as closed. One can therefore define the interior and the exterior of level lines. When the gray level value along the interior side of the line is brighter than the value on its exterior side, the line is said to be *positive*. In the opposite case, the line is said to be *negative*. These notions are illustrated on Figure 1.

The whole family of the level lines of an image is called the topographic map [Caselles et al. 1999]. It can be efficiently extracted from an image using the Fast Level Lines Transform (FLLT) [Monasse and Guichard 2000]. This representation enjoys several important advantages. First, it provides a complete representation of an image. Indeed, it is easily seen that an image can be reconstructed from all its level lines [Serra 1982]. It also has a hierarchical structure, since the interiors of two different lines are either included one in the other or disjoint. Lines can therefore be given a tree structure [Caselles et al. 1999; Monasse and Guichard 2000]. Last, object contours often locally coincide with level lines [Serra 1982; Cao et al. 2008]. Yet, the topographic map contains redundant level lines following the image contours and many lines correspond to texture information or noise. The first goal of this paper is to

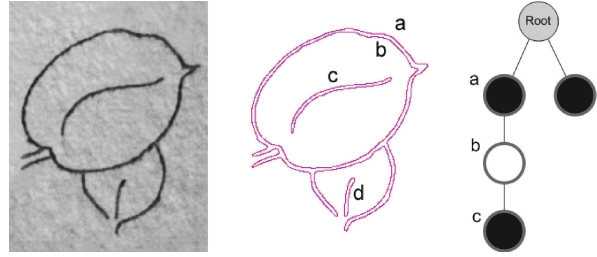


Fig. 2. A simple line-drawing made of two leaves (left). Four level lines are manually chosen to best represent the strokes contours (middle). On the tree structure of these four level lines (right), a positive and a negative level line are represented by a white and a black node respectively. The unique positive level line b describes the inner contour of the biggest leaf. We show in Section 2.2 how to achieve this representation automatically.

filter the tree of lines in order to reach a one-to-one correspondence between the contours of stroke clusters and level lines that are kept. An example of a line artwork is presented on Figure 2. Four level lines of this artwork are manually chosen to best represent strokes contours. The tree structure of these four level lines is also shown. Positive and negative level lines are represented by white and black nodes respectively and inclusion is bottom-up: a child level line is included in its parent. The goal is to automatically build such a tree from the original tree of all level lines (which on this drawing, for instance, is made of more than 26000 lines).

The first step in order to reach a representation such as the one shown on Figure 2 is to consider lines that are long and contrasted enough, the so-called *meaningful level lines* as defined in Desolneux et al. [2001], which are briefly presented in the next section.

2.1.2 Meaningful Level Lines. In order to filter the redundant set of level lines (the topographic map) of an image u , Desolneux et al. rely on an *a contrario* approach. The basic idea is to keep only lines whose high contrast is very unlikely under some null hypothesis. The null hypothesis is that the gradient magnitudes at points along a level line are independent and identically distributed random variables. This yields the following line selection criterion. A line \mathcal{L} is selected if $NFA(\mathcal{L}) \leq \epsilon$, where

$$NFA(\mathcal{L}) = N_u \times H\left(\min_{x \in \mathcal{L}} |\nabla u(x)|\right)^l, \quad (1)$$

with N_u the number of lines in u , $\nabla u(x)$ the gradient at point x (computed using a 2×2 scheme) and H the empirical distribution of the gradient on the image. This empirical distribution is simply computed as

$$\forall \mu > 0, H(\mu) = \frac{\#\{x, |\nabla u| > \mu\}}{\#\{x, |\nabla u| > 0\}}, \quad (2)$$

where $\#$ denotes the cardinality symbol.

The criterion $NFA(\mathcal{L}) \leq \epsilon$ ensures that there are less than ϵ false detections when testing N_u lines following the null hypothesis. More details can be found in [Desolneux et al. 2001]. A line for which this criterion is true is said to be ϵ -meaningful. In practice, and this will be the case in this paper, it is safe to set $\epsilon = 1$, in which case one simply speaks of *meaningful level lines*. Let us add that the value $NFA(\mathcal{L})$ provides a measure of the meaningfulness (quality) of the level line \mathcal{L} . The smaller it is, the more meaningful the line is (the less probable it is to detect such a line by accident).

Meaningful level lines are empirically well localized along contours but are still largely redundant. Indeed, digital sampled images present thick edges where many parallel meaningful level lines take place. This effect is illustrated on Figure 3. This is the reason why, in order to further filter the tree of



Fig. 3. The set of meaningful level lines from the same drawing as on Figure 2 (left) and a detailed window (right). The tree structure contains 717 level lines. The full topographic map contains more than 26000 level lines. The gray levels of the original drawing are coded on 8 bits.

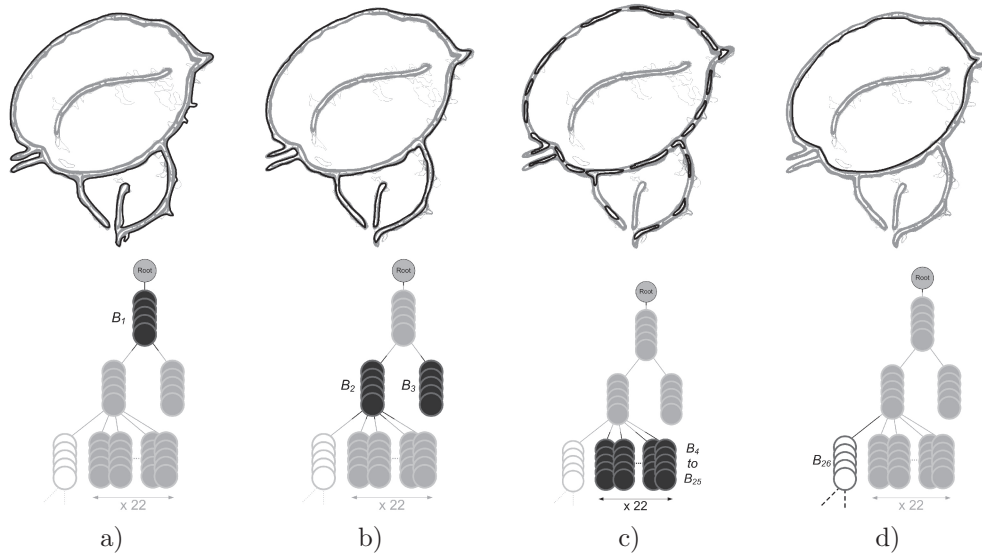


Fig. 4. Tree-structure of the set of meaningful level lines. Each series of nodes represents a maximal monotone branch, which is illustrated on the drawing by a thick black line. a) at the first level, a maximal monotone branch encompasses all the strokes. b) the last node of the first branch encompasses two strokes that are disjoint. The small isolated stroke (the vein of the small leaf) is represented by the branch B_3 . c) the branch B_2 encompasses 22 negative branches due to contrast variation along the strokes and 1 positive branch due to the inner contour of the leaf. d) the branch B_{26} made of positive level lines represents the inner contour of the big leaf. The rest of the tree (the inner part of the big leaf) is not detailed here.

lines, Desolneux et al. [2001] introduced the concept of *maximal* meaningful lines, as recalled in the next section.

2.1.3 Redundancies and Maximality Principle. First observe that the hierarchical structure of the level lines representation is preserved when selecting meaningful level lines. These lines are therefore organized as an inclusion tree. A monotone branch is a branch made of n nodes $\{\mathcal{N}_i\}_{i=1\dots n}$ of the tree where all nodes are of the same type (positive or negative) and such that, for all $i \geq 2$, \mathcal{N}_i is the only son of \mathcal{N}_{i-1} . A maximal monotone branch is a monotone branch that is not strictly contained in another monotone branch. A meaningful level line is said to be maximal [Desolneux et al. 2001] if its NFA is minimal on the maximal monotone branch to which it belongs. This selection method is illustrated on Figure 4. On this figure, each series of nodes represents a maximal monotone branch, illustrated on

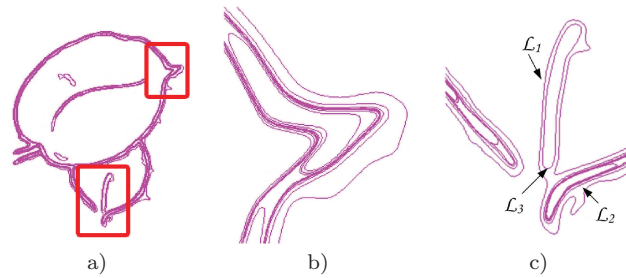


Fig. 5. Same example as on Figure 3 using Desolneux et al. branch maximality principle (left). Two details are also displayed (center and right). The resulting set contains 54 level lines. The number of lines has been dramatically reduced, but redundancy remain along stroke contours.

the drawing by a thick black line. On Figure 5, resulting maximal meaningful lines are displayed for two details of the line drawing of Figure 2. One can observe that, even though the set of lines has been dramatically reduced, there are still redundant lines. Level line \mathcal{L}_1 which is selected in branch B_1 , for instance, is redundant with level lines \mathcal{L}_2 and \mathcal{L}_3 from branches B_2 and B_3 respectively (Figure 5(c)). To cope with this fact, we introduce in the next section a new maximality principle, specifically adapted to strokes in line drawings.

2.2 Stroke Cluster Extraction

Inspired by the preceding method we now propose in this section a new algorithm to filter the tree of meaningful level lines which is specifically suited to line drawings. We first describe in Section 2.2.1 a new maximality principle. This parameter-free principle filters the meaningful level lines by taking advantage of the specific structure of line drawings. Next, we show in Section 2.2.2 that thanks to the hierarchical structure of the level lines and to line drawing properties, it is possible to group lines related to every stroke cluster.

2.2.1 A New Maximality Principle: MMT-Filtering. The geometrical analysis that we will present in Section 3 relies on stroke contours. It is therefore essential that each contour be described by a unique level line. To eliminate redundancies we propose a new maximality principle which takes advantage of the topological structure of line drawings.

We define a monotone tree to be a subtree of the set of meaningful level lines containing only positive or only negative nodes. We call maximum monotone tree (MMT) a monotone tree which cannot be included in a bigger monotone tree. On Figure 4, for instance, the first negative MMT would correspond to the union of the 25 shown negative maximal branches labeled B_1 to B_{25} .

The first observation is that, thanks to the physical profile of classic artistic tool and the averaging produced by the gesture, it is unlikely that a strong positive contrast variation occurs inside a stroke. This would correspond to some artifacts or spurious reflections. Within the tree of meaningful level lines, the contrast sign should change only if some level lines represent inner contours of clusters (as line b on Figure 2 for instance). This observation then implies that redundant level lines corresponding to a single contour are grouped inside one MMT. In order to filter redundant level lines and to obtain a single contour, we therefore consider every MMT separately.

Yet conversely, different contours can coexist in the same MMT, along with their respective redundant level lines. This situation happens in the first MMT of Figure 4 (branch B_1 to B_{25}). The outer contour of the biggest leaf (line a on Figure 2) and the contour of the small vein of the small leaf (line d on Figure 2) are in the branches B_2 and B_3 of the first MMT, respectively.

Table I. MMT-filtering

| |
|--|
| Input Meaningful level lines tree S |
| Output Filtered level lines tree |
| <i>Main Function:</i> |
| For all maximal monotone tree S_i of S do |
| filter (S_i) |
| End for |
| <i>Function filter</i> (S_i) |
| While S_i is not empty do |
| - save $\mathcal{L} \leftarrow$ most meaningful level line of S_i |
| - remove all level lines of S that are children or parents of \mathcal{L} in S_i |
| End while |

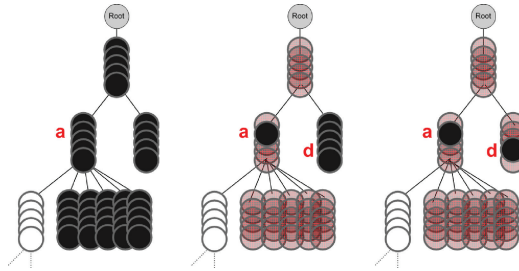


Fig. 6. MMT-filtering applied to the first maximal monotone negative subtree of the tree shown on Figure 4. Left: the most meaningful level line a is found. (a refers to the corresponding label on Figure 2). Middle: level lines that are above or below this level line in the tree are removed from the maximal monotone tree (MMT). Right: the next most meaningful level line is recursively found in the remaining tree (label d). At the end of this step, the first MMT is empty.

A last observation is as follows. Let \mathcal{L} be a level line that meaningfully corresponds to a stroke contour, and $S_{\mathcal{L}}$ the MMT in which \mathcal{L} is included. Due to the topological constraints induced by a line-drawing structure, there cannot be another level line in $S_{\mathcal{L}}$ corresponding to another stroke contour and being a children of \mathcal{L} . We thus can prune the children of \mathcal{L} in $S_{\mathcal{L}}$ since they should describe the same stroke. We can also prune parents of \mathcal{L} in $S_{\mathcal{L}}$ since they should either describe the same stroke or embrace several strokes inside the MMT.

Following the previous remarks, we propose the following method, called MMT-filtering, to filter the tree of lines. For each MMT of the tree of meaningful level lines, we first look for and save the most meaningful level line \mathcal{L} (i.e., the level line having the lowest NFA in the MMT). We then remove all level lines which are children or parents of \mathcal{L} in the MMT. These two steps are repeated while the MMT is not empty.

This algorithm is presented in Table I. Applied to the tree-structure presented on Figure 3, this algorithm returns the level lines which are shown on Figure 2(b). The filtering of the first MMT for this example is detailed on Figure 6 for the first MMT.

2.2.2 Coalescence Sets. In this section, we propose to group the maximal meaningful level lines selected by the preceding filtering algorithm that correspond to the same stroke cluster. The corresponding groups of lines will be called *coalescence sets*. We make the hypothesis that we deal with drawings made of dark strokes over a bright background. In the case of drawings made of bright strokes over a dark background, the negative image could be processed.

A stroke cluster is described by a coalescence set of one negative level line (the surrounding contour) and zero or more positive level lines. These positive level lines are necessarily children of the negative level line. Since the hierarchical structure of level lines is preserved throughout the process of line selection, it is quite simple to construct all the coalescence sets of a drawing. Every negative level line is a root of a coalescence set, and all direct positive children are in this coalescence set.

The drawing of Figure 2, for instance, has three clusters: the two small veins inside and outside the large leaf, and the large leaf contour itself. Each contour of the veins is described by a coalescence set made of one negative level line. The large leaf is described by a coalescence set made of one negative level line (the outer contour), and one positive level line (the inner contour). These coalescent sets are used in the next section to define several features characterizing strokes in line-drawings.

3. PICTORIAL CONTENT ANALYSIS

This section proposes several methods to analyze the geometrical content of coalescence sets. The tool radius is the first extracted pictorial information. Another important information is the curvature along stroke contours. These two quantities are crucial to tune several geometrical parameters in the following steps. These steps consists in detecting and analyzing curvature zero-crossings, stroke junctions, corners and endpoints. The choice of these geometrical cues is inspired from several studies on visual perception and art-related psychophysics. These references are progressively introduced in the following sections for each computed characteristic.

3.1 Level Lines Smoothing

Curvature being quite sensible to noise, level lines of the coalescence sets needs to be smoothed. In order to maintain the hierarchical structure inherited from the topographic map, a smoothing preserving the inclusion of level lines is needed. This prevents us from using Gaussian smoothing [Cao 2003]. Instead, affine plane curve smoothing is chosen [Alvarez et al. 1993] which consists in applying the following partial differential equation (PDE) on a level line $s \mapsto \mathcal{L}(s, 0)$:

$$\frac{\partial \mathcal{L}}{\partial t}(s, t) = \kappa^{1/3}(s, t)\mathbf{N}(s, t), \quad (3)$$

where t is the scale parameter, κ the local curvature, and \mathbf{N} the normal vector to \mathcal{L} . The choice of the final scale T will be discussed in Section 4.2. Experiments are done with the practical implementation of Moisan [1998] which preserves theoretical properties of the evolution (monotonicity, affine and morphological invariance). It applies directly to polygonal models of level lines. Let us stress that only one smoothing is operated here at a very small scale. Our approach differs from a multiscale approach such as [Mokhtarian 1995], where the smoothing acts as an analysis tool. Our goal is to reduce noise at pixel scale before computing curvature along level-lines

3.2 Tool Radius Estimation

In order to estimate the size of the drawing tool, we make the following simplification that let us easily estimate the tool radius: we will consider that each **stroke cluster** Φ has been drawn with tools having the same radius R_Φ . This is somehow restrictive but a large amount of drawings, including the ones considered in our tests, respect more or less this hypothesis. As we will see in Section 4.3, although the hypothesis is not strictly observed, the tool radius estimation using this simplification is very robust. A possible extension of the method enabling to deal with a spatially variable radius would be to rely on some local analysis of the parallelism of lines.

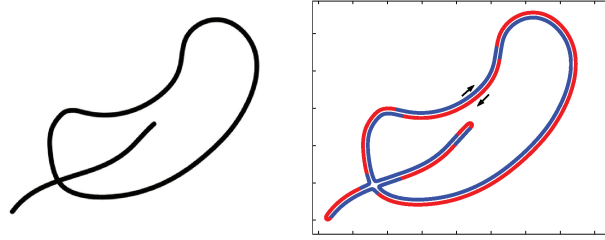


Fig. 7. During curvature computation, convention is chosen so that negative (resp. positive) level lines are traveled clockwise (resp. anticlockwise). On this example (left), one negative level line delimits the outer contour, and one positive level line delimits the inner contour (right). Arrows notify the travel direction. This convention induces negative curvature values (red lines) and positive values (blue lines) wherever they represent an inner or an outer stroke cluster contour. This convention ensures us that pictorial elements such as stroke endpoints or junctions have the same local curvature sign.

Using a constant width model of stroke made with a circular tool, for each stroke cluster Φ associated with a coalescence set made of n level lines \mathcal{L}_i , the tool radius R_Φ is then given by:

$$R_\Phi \approx \frac{\text{area}}{\text{perimeter}} = \frac{-\sum_{i=1}^n \text{sign}(\mathcal{L}_i) S_i}{\sum_{i=1}^n P_i}, \quad (4)$$

where S_i (resp. P_i) is the polygonal surface (resp. perimeter) of the level line \mathcal{L}_i .

3.3 Curvature Computation

For each curvilinear abscissa s_i along a level line \mathcal{L} , the curvature $\kappa(i)$ is estimated as

$$\kappa(i) = \theta'(s_i) \approx \frac{\theta_i - \theta_{i-1}}{(s_{i-1} - 2s_i + s_{i+1})/2}, \quad (5)$$

where θ_i is the local orientation of the segment $[s_i, s_{i+1}]$. This curvature estimation is less sensitive to noise than the ones directly based on first and second derivatives. It gives a curvature value relative to the pixel size. Practically the same sampling rate $\Delta s = 0.5$ pixels for curvilinear abscissa is used for every image.

Curvature is a signed information depending on the direction of traveling along a closed curve (clockwise or anticlockwise) and the local geometry (convex or concave). We choose the following convention. Negative level lines will be traveled clockwise. This induces negative curvature values at line convexities and positive values at line concavities. On the opposite, positive level lines will be traveled anticlockwise to invert the curvature sign. This convention ensures us that pictorial elements such as stroke endpoints or junctions have the same curvature sign wherever they are located in a coalescence set (see Figure 7). Moreover, endpoints always correspond to negative curvature values, and regions where strokes create a nonreflex angle (e.g., at junctions) hold positive values.

Curvature zero-crossings contribute to artistic visual impression and object recognition [Mokhtarian 1995]. They are estimated directly on the curvature signal, without using any threshold. An example of a drawing with detected curvature zero-crossing is shown on Figure 8(b).

3.4 Detection of Endpoints, Junctions and Corners

This section proposes to detect three different strong visual cues: stroke junctions, stroke corners and stroke endpoints, according the taxonomy presented on Figure 9-I and inspired from Plamondon and Privitera [1999]. These features all correspond to points with high curvature values. We thus first extract such points before classifying them.

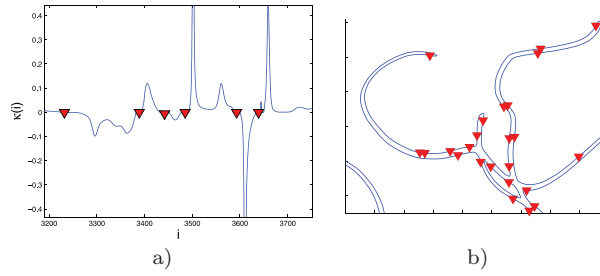


Fig. 8. Left: curvature signal along a stroke contour. Triangles represent estimated zero-crossings. Right: maximal meaningful level lines from a drawing, and its superposed zero-crossings.

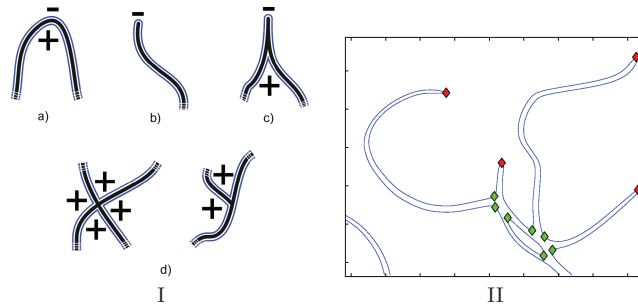


Fig. 9. Left: taxonomy of high curvature values along stroke contours: a) a simple high curvature point along the gesture trajectory, b) an endpoint, c) a stroke corner, d) two types of stroke junctions: X and T junctions. Plus and minus sign refers to the curvature sign on the stroke contour. Strokes represented by black lines, and the stroke contours are represented by blue surrounding lines. Right: curvature maxima detected with the method presented at Section 3.4.1. Red (resp. green) diamonds are positive (resp. negative) maxima.

3.4.1 *Selection of Candidates to a Taxonomy of High Curvature Points.* We first extract a large set of extrema that are candidates to the extrema point taxonomy shown on Figure 9-I. This task is related to dominant points detection in pattern recognition. We use the curvature information to extract such points. Similar approaches can be found in Ghosh and Petkov [2006], Wu [2003], and Gu and Tjahjadi [2000]. Other approaches use local measures which are different from curvature [Poyato et al. 2004; Marji and Siy 2003], Gaussian scale spaces [Mokhtarian 1995] or wavelets [Fayolle et al. 2000].

Recall that R_Φ is the tool radius. We consider every continuous portion of the curvature signal where $|\kappa(i)| > \kappa_t$ with $\kappa_t = 1/(k_c R_\Phi)$. We will discuss the choice of k_c in Section 4.4. Zero-crossings of the derivative curvature signal on each of these portions are then considered as candidates to the taxonomy. The threshold κ_t holds two objectives. First, it allows us to select the set of candidates to the taxonomy of Figure 9-I. Next, it will also be useful to select *flat parts* of the strokes. Flat parts are portions of the strokes that are coarsely flat and empty of any endpoints, corners or junctions. We will use these flat parts in Section 3.5.

Depending on the curvature sign, an extremum can be either positive or negative. An example of extrema detection is shown in Figure 9-II. Thanks to the convention on curvature presented in Section 3.3, stroke endpoints coincide with negative maxima (Figure 9-I(b)), and junctions coincide with several positive maxima (Figure 9-I(d)). Corners have one positive maximum on the concave side, and one negative maximum on the convex side (Figure 9-I(c)). This information will be useful in the next three sections to analyze stroke endpoints, junctions and corners.

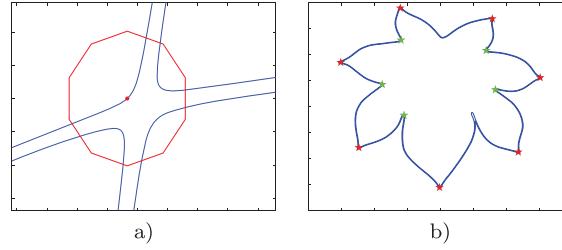


Fig. 10. Left: to state a stroke junction, a disk is centred on the positive curvature maximum. In this example, four different groups of neighboring points are covered by the disk. Right: stroke corners detected on a drawing. Red stars (resp. green) are convex line corners (resp. concave).

3.4.2 Stroke Junctions. Stroke junctions are important geometrical characteristics of the 1D content of a drawing. Each junction indicates a possible occlusion denoting a level of perspective in the depicted scene [Willats 1997]. A positive curvature maximum at a point p_m indicates a possible junction nearby its location. To state for a junction we center a disk of radius $k_j R_\Phi$ on p_m (see Figure 10(a)). If there are three or more pieces of level lines covered by the disk, a stroke junction is detected and p_m corresponds to one of the non-reflex angle of this junction. If there are only two pieces, this maximum is a line corner which will be characterized further in the next subsection. Once all positive curvature maxima in a coalescence set have been analyzed, the ones that have been detected as belonging to a stroke junction are finally merged if their mutual distances are less than $k_j R_\Phi$. The choice of k_j will be discussed in Section 4.4.

3.4.3 Stroke Corners. A positive maximum that has not been classified as a stroke junction is considered as a stroke corner. Such maxima points have a very strong pictorial impact [Leyton 2006] (see Figure 10(b) for an illustrative example). Among the characteristics that have been previously proposed in the perception literature to measure the visual strength of a corner, we use the relative surface as a corner strength description [Winter et al. 2002]. Considering a maximum p_m , we iteratively consider the polygon $p_{m-i} \dots p_m \dots p_{m+i}$ for $i > 0$. This polygon is expanded while it does not contain any point of the coalescence set other than the points $\{p_{m-i}, \dots, p_{m+i}\}$. The strength value is computed as the relative surface of this maximal polygon normalized by the total surface of the image.

If the underlying stroke describes a shape such as an object contour, a line corner is usually perceived as convex or concave, inducing a different pictorial effect. This type of effect has been recently studied [Fantoni et al. 2005; Feldman and Singh 2005]. Recognizing the type of a corner (i.e., stating if it convex or concave) can be an ill-defined problem in a line drawing. Indeed, strokes do not always describe a part of an object contour in which case open strokes can create some ambiguities. We propose a well-defined way to specify the visual orientation of a corner, that we call *sign of a corner*. We say that a stroke corner is negative if it points toward the center of mass of the whole coalescence set to which it belongs, and positive if it points toward the opposite direction. The corner orientation is estimated using the orientation of the geometrical vector $\overrightarrow{p_{m-i_\Phi} p_m} + \overrightarrow{p_{m+i_\Phi} p_m}$, where i_Φ is such that $i_\Phi \Delta s \approx R_\Phi$. A typical case where a corner sign and a corner type do not correspond is when the form describes a U-bend around the corner.

3.4.4 Stroke Endpoints. To detect stroke endpoints, the set of negative curvature maxima is analyzed. Stroke endpoints are important components of the pictorial content. A drawing made of dotted lines delivers a different pictorial effect than the same drawing made with continuous strokes. For each negative maximum point p_m , we consider the two points p_i, p_j (where $i < m < j$) along the level line at a geodesic distance $k_e R_\Phi$ from p_m . If the Euclidean distance between p_i and p_j is less than $3R_\Phi$,

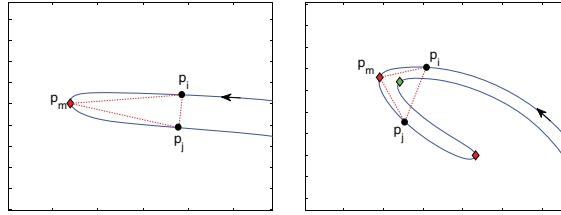


Fig. 11. Left: negative maximum is classified as a stroke endpoint. Right: some points p_q which do not belong to the level line portion $[p_i, p_j]$, are included in the triangle $p_i p_m p_j$. This maximum is thus not classified as a stroke endpoint.

Table II. Features Vector Used for Similarity Retrieval

| Features |
|---|
| 1. mean of the curvature distribution of $\kappa(i) \in [-\kappa_t, 0]$ |
| 2. standard deviation of the curvature distribution of $\kappa(i) \in [-\kappa_t, 0]$ |
| 3. kurtosis of the curvature distribution of $\kappa(i) \in [-\kappa_t, 0]$ |
| 4. linear density of curvature zero-crossings |
| 5. linear density of stroke endpoints |
| 6. linear density of stroke junctions |
| 7. linear density of convex stroke corners |
| 8. linear density of concave stroke corners |
| 9. sum of the depths of convex stroke corners |
| 10. sum of the depths of concave stroke corners |
| 11. normalized tool radius |

and if there does not exist any point p_q inside the triangle $p_i p_m p_j$ such that $q < i$ or $q > j$, then p_m is considered as an endpoint. This distance threshold approximately tolerates a 50% error on the estimation of R_ϕ . If the maximum is an endpoint, the distance between p_i and p_j should indeed be close to the width $2R_\phi$. This method is illustrated on Figure 11. We do not consider here the disk previously used in Section 3.4.2 to detected stroke junctions. The triangle $p_i p_m p_j$ lowers the risk to provoke miss detection of endpoints. However, as we will see in Section 4.4, k_e will be empirically set to the same value as k_j .

3.5 Similarity Retrieval

To illustrate how the proposed features can be used for the indexing of line drawings, a similarity retrieval framework is considered in the experimental section. Based on the pictorial content analysis presented in previous sections, eleven scalar features are computed, summarized in Table II. The three first features are computed on the distribution of curvature values $\kappa(i)$ satisfying $-\kappa_t \leq \kappa(i) \leq 0$. They are defined as the mean, standard deviation and kurtosis of such curvature values. Observe that negative curvature values let us select one half of the curvature points along the strokes (see Figure 8). We thus indirectly consider points that are related to the underlying drawing stroke. Observe also that we only consider values that respect $-\kappa_t \leq \kappa(i)$ to discard the strong curvature values that correspond to stroke endpoints. The corresponding parts of level lines are what we define as *flat parts* of the drawing. The three first features therefore give an indication of how *straight* are the flat parts of the strokes independently of the rest of the geometrical information. Curvature values are normalized by the image diagonal. These three features thus become invariant to the drawing scale. This is required if we want that two images of a same drawing with two different resolutions have the same feature values.

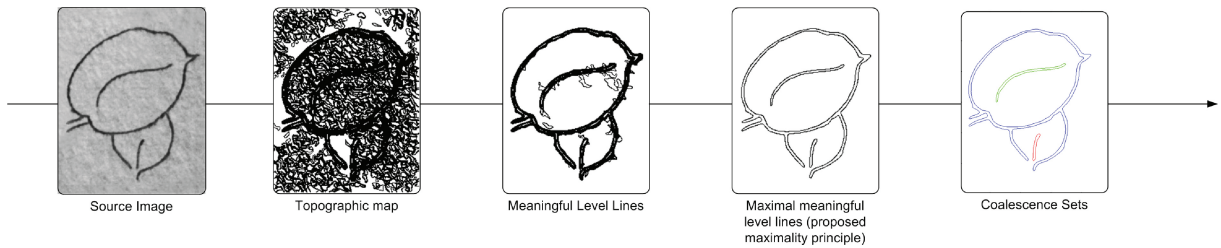


Fig. 12. Overall framework for the detection of stroke contours. First, the topographic map of the numerical image is computed. The probabilistic scheme recalled in Section 2.1.2 is then applied to select meaningful level lines. The MMT-filtering method introduced in Section 2.2.1 next eliminates redundancies. Cluster of strokes are finally segmented by grouping level lines into coalescence sets. Our contributions to this framework are the last two steps.

Features 4 to 8 are linear densities of curvature zero-crossings, stroke endpoints, stroke junctions, positive and negative stroke corners. The linear density of each of these geometrical elements is defined as the total number of such elements in the image divided by the total length of the level lines of the coalescence sets. Each linear density is normalized by the image diagonal. Features 9 and 10 are the sum of the strength values of convex and concave stroke corners. These two features are normalized by the image surface and therefore represent the relative image surface corresponding to each signed stroke corners. The last feature is the tool radius normalized by the image diagonal.

Let us notice that linear densities of endpoints and junctions have been previously proposed in Julesz [1986] for perceptive *preattentive* discrimination of textures. We have conducted several experiments on Julesz textons using the methods proposed in this paper. They led to similar perceptive results.

4. EXPERIMENTS AND RESULTS

Experiments are first performed on several synthetic examples in Sections 4.1 to 4.3. These are aimed at illustrating the detection of stroke contours, the precision of the curvature estimation, and the estimation of the tool radius. The setting of k_c , k_e , and k_j is discussed in Section 4.4. Section 4.5 then presents two classified databases of line-drawings, which are used in Section 4.6 to evaluate the proposed analysis method. In all experiments, images are coded on 8 bits and the quantization step used to extract the topographic map is 1.

4.1 Detection of Stroke Contours

Stroke contours are detected using the framework presented in this paper and illustrated on Figure 12. Stroke contours are represented by meaningful level lines, filtered as explained in Section 2.2. Among all the drawings used in this paper, the mean reduction rate of the set of maximal meaningful level lines between Desolneux maximality principle and the MMT-filtering proposed in this paper is 88%.

4.1.1 Stroke Intensity Variations. A level line passes through constant values of gray levels. What happens along strokes where the gray level vary? This situation happens frequently since we do not binarize images. Unless some gray values inside the stroke reach background values, there is actually always at least one level line surrounding the stroke. On Figure 13(a), several synthetic strokes with varying gray values are drawn on a white background, with added Gaussian noise ($\sigma = 5$). On Figure 13(b) is shown the topographic map with a quantization step of 15. There are several level lines surrounding the stroke, with several nested level lines following the linear variation of the gray levels. On Figure 13(c), our detection method selects the most meaningful contour, except for the bottom stroke which might be perceptually visible (using a good continuation principle) although its minimum

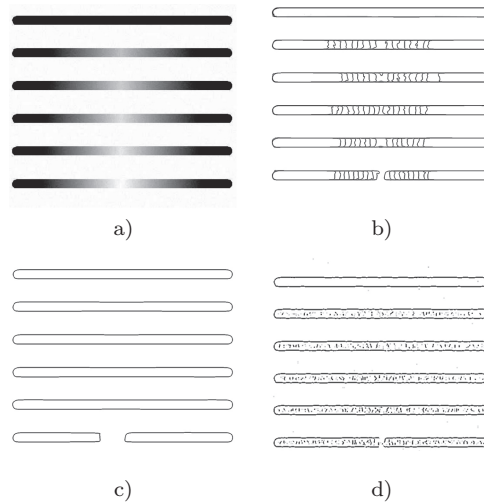


Fig. 13. a) Several synthetic strokes are drawn on a white background, with added Gaussian noise ($\sigma = 5$). The top stroke has a constant gray level value of 0. Next strokes have a gray level value of 0 at their endpoints which linearly increases to respectively 215, 225, 235, 245 and 255 at their middle. b) Topographic map with a quantization step of 15. c) Our detection method selects the most meaningful contour, except for the bottom stroke which might be perceptually visible (thanks to the Gestalt principle of good continuation) although its gray level value reaches the background value. d) Results obtained with a Canny-Deriche detector whose parameters have been set manually to reach a good trade-off between noise and contours detection.

gray level value reaches the background value. On Figure 13(d) is shown the same image analyzed with a Canny-Deriche detector whose parameters are set manually on this image to reach a good trade-off between noise and contours detection.

In conclusion, the stroke contour extraction method presented in this paragraph has several key advantages compared to other approaches. First it inherits the advantages of the topographic map extracted using the FLLT [Monasse and Guichard 2000] and of the meaningful level-lines as defined in Desolneux et al. [2001]. Indeed, the resulting extraction provide closed lines without needing any reconstruction step. It relies on a single parameter (the meaningfulness) that is set to $\epsilon = 1$ for all experiments in this paper. The filtering principle proposed in paragraph 2.2 relies on simple observations on the structure of stroke contours and is parameterless. Lines are invariant to affine contrast changes and gathered in a hierarchical structure.

4.2 Curvature Estimation

Recall that level lines are regularized thanks to the affine invariant plane curve evolution presented in Section 3.3. We choose the numerical implementation of Moisan [1998] which preserves all theoretical properties (affine and morphological invariances) of the smoothing. A final time of evolution T needs to be fixed. Moisan implementation normalizes this parameter T in such a way that a circle of radius T should vanish exactly at scale T . In all our experiments we use $T = 0.5$ pixel, so that small irregularities having a size of the order of one pixel are removed. Considering the mean spatial resolution of the digitized drawings in our database (4 pixels/mm²), this means that the PDE smoothes details of size less than 0.25mm². This implementation also give control over the sampling step along level lines. This step is fixed to $\Delta s = 0.5$ pixels for all experiments in this paper.

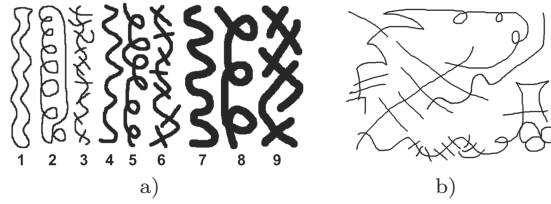


Fig. 14. Two synthetic drawings made with Gimp. The tool radius is estimated using the method proposed at Section 3.2.

4.3 Tool Radius Estimation

The tool radius is an important measure on which depend most of the features proposed in this paper. To evaluate its robustness, tests have been performed on two synthetic drawings, made with the Gimp software in which the tool radius can be fixed. The first one is shown on Figure 14(a) and presents three groups of strokes with different radius: 3, 7 and 15 pixels. The tool radius is estimated using Equation (4). The average error on this example is 12.5%. Line-drawings often include very long strokes and complex clusters. Such a situation happen on Figure 14(b). The error on this example is 1.8%. Let us note that such low error values have little impact on the rest of the methods. The tool radius is used as a coarse information to calibrate some geometrical scales to further analyze drawings. These errors are due to our simple estimation equation (4) which does not take into account possible stroke overlaps at crossings.

4.4 Setting of Parameter k_c , k_e and k_j

Recall that k_c , k_e and k_j are parameters defined in Section 3.4, respectively for the selection of high curvature points, stroke endpoints and stroke junctions.

First, we saw that only points with curvature bigger than $\kappa_t := 1/(k_c R_\phi)$ are considered as high curvature points. Ideally, κ_t is set in such a way that the three first features of Table II do not depend on the presence of endpoints. This is not true when k_c is too low since endpoints regions may be integrated in the distribution. On the opposite, a too high value of k_c raises the risk of not discriminating between a drawing holding strictly straight linear parts and another made of slightly curvy strokes. A good trade-off has been empirically set with $k_c = 5$, using synthetic images made with different tool radius and real line-drawings.

Recall that endpoints are detected using a triangle of scale $k_e R_\phi$ around a maximum candidate (Section 3.4.4). Increasing k_e raises the risk to miss some endpoints that are very close to a junction. Lowering k_e raises the risk to detect some elongated junctions of two strokes as an endpoint.

Last, recall that stroke junctions are detected using a disk whose radius is $k_j R_\phi$. Increasing k_j raises the risk to consider too large regions than what is necessary to cover a stroke junction, and thus provoke some false detection. Lowering k_j raises the risk to miss some stroke junction. Empirical experiments with real and synthetic drawings led us to choose $k_j = k_e = 5$. In the remaining of this paper, we will thus use those two values.

4.5 Two Classified Line Drawings Databases

To our knowledge, there is no publicly available dataset of line drawings classified according to the pictorial content, or more generally according to some style or artistic criteria. We therefore built up two databases in order to test the proposed methodology. The first one gathers 74 digitized line drawings classified by a professional artist into 10 classes. This database is called thereafter the DB1 database. The professional artist was asked to group drawings according to its first pictorial impression. The aim of this step is to obtain classes that are coherent, at least for a human observer (we will confirm this

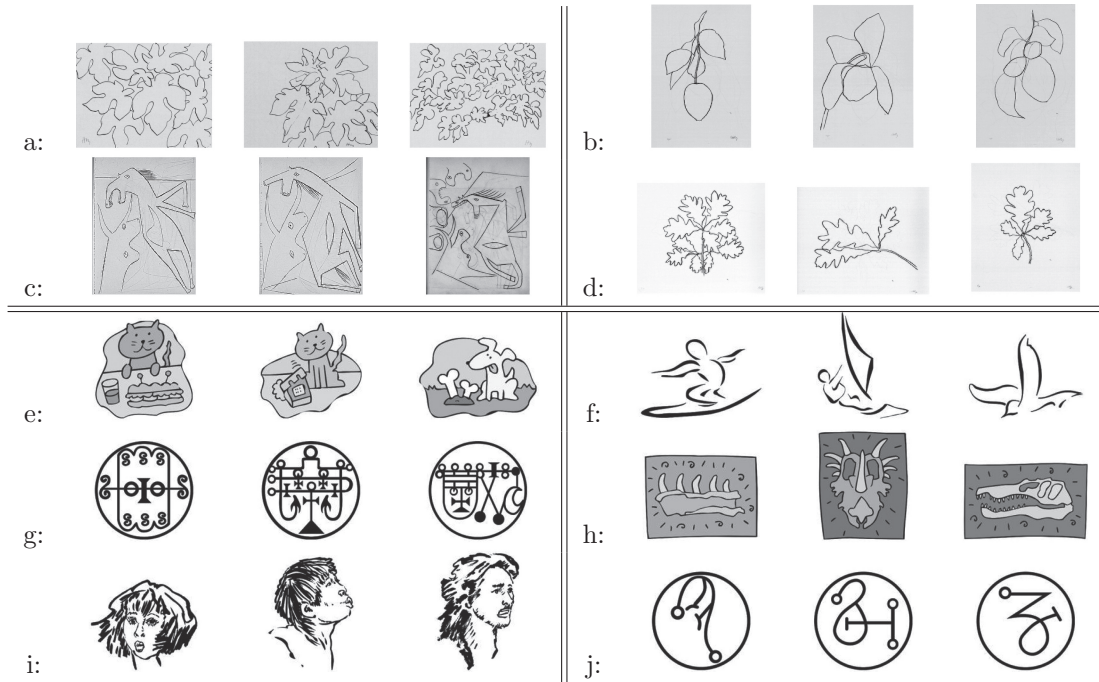


Fig. 15. Samples of some classes from the two classified databases DB1 (classes *a* to *d*) and DB2 (classes *e* to *j*) used in the Section 4.6. ©Succession H. Matisse, Private Collection (*a*); ©Ellsworth Kelly, Grand Rapids Art Museum (*b*, *d*); ©Succession Picasso, Musée Picasso (*c*).

fact further in this section). Two-thirds of this database originates from the catalog of an exhibition of plant drawings made by Henri Matisse and Ellsworth Kelly [Labrusse and de Chassey 2002]. Interestingly, the curators of this exhibition observed that the drawn subjects and the artistic style of both artists are quite similar in this series. They also remark that some subtle differences may appear in the visual impression delivered to the observer that sometimes let the spectator to recognize one of the two artists, but also sometimes mislead him [Labrusse and de Chassey 2002]. These drawings have been scanned with 4800 ppi resolution. The other drawings originate from several Picasso sketch books released on DVD-Rom as a 800×600 digital facsimile edition [Dupuis-Labbé 2006]. Figures 15(a)–(d) show three samples from 4 classes of this database.

A second database, henceforth called the DB2 database, is made of 673 computerized line drawings, all coming from the commercial ArtExplosion clipart repository [ArtExplosion 2002]. These vectorial line drawings have been rasterized into JPG images to let us investigate the behavior of the general approach presented in this paper, including the segmentation step. DB2 gathers images that have been manually classified by an amateur artist into 31 classes. Figures 15(e)–(j) show three samples from 6 classes of this database.

For both databases, images and classes are available on a permanent Website [Hurtut 2010]. In the next section, these databases are used to perform image retrieval. As already mentioned, we are not aware of any standard database classified according to the pictorial content. Such a classification is indeed a hard task and it is legitimate to wonder which confidence one can give to the different classes. In order to address this issue, we built up a sanity check experiment. We asked N different users to classify DB1 and DB2 given the classes found by the expert and a random representative image from

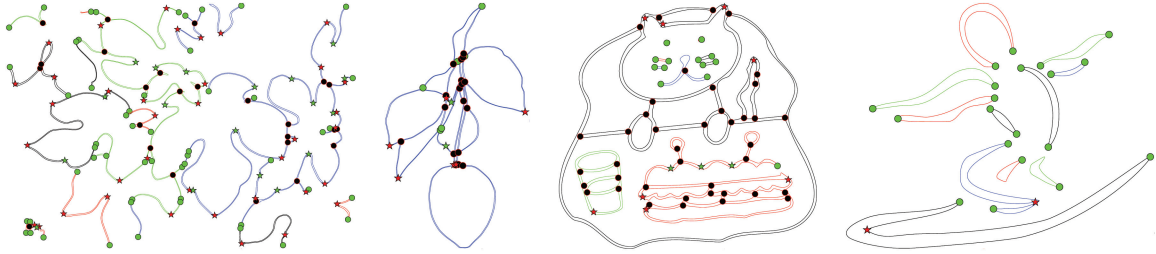


Fig. 16. Four examples of analyzed drawing. Green dots indicate stroke endpoints, black dots indicate stroke junctions, and red (resp. green) stars indicate negative (resp. positive) stroke corners. Coalescence sets are given different colors. From left to right, these examples are based on the first images of classes a , b , e and f on Figure 15.

each class. Users were asked to iteratively assign new images randomly picked in the database to one of the class representative.

For the first database, $N = 14$ people participated to this experiment, achieving a precision rate of 93.7% (with respect to the classification of the artist). $N = 3$ people went through the classification of the second database, achieving a precision rate of 92.7%. This indeed shows the strong repeatability of the classes found in both databases and that this classes are non-ambiguous for human observers.

4.6 Drawing Retrieval

For each drawing, the proposed geometrical features are first extracted. Four examples of analyzed drawings are shown on Figure 16 to illustrate the detection of pictorial information. The 11 features presented in Table II are then computed for each image. Computational times for the whole features extraction is around 15 seconds for a 800×600 image using a PC Pentium IV running at 4.3GHz. Most of this time is dedicated to the stroke contours detection. Once the 11 features are computed for every artwork in the database, these are centered by the mean and normalized by the standard deviation of the feature distribution over the considered database. Let $V^a(k)$, $k \in [1, 11]$ be the eleven normalized features of a drawing a . The similarity measure $d(V^a, V^b)$ between two drawings is the following weighted Euclidean distance:

$$d(V^a, V^b) = \left(\sum_{k=1}^{11} w_k (V^a(k) - V^b(k))^2 \right)^{1/2}, \quad (6)$$

where $W_{11D} = (w_1, \dots, w_{11})$ is a 11-dimensional weight vector. We limit our study to this simple dissimilarity measure. Our goal here is to analyze the different proposed features. We do not tend to optimize the classification of DB1 and DB2 (e.g. by using some machine learning procedure).

4.6.1 Comparison with a CSS-Based Approach. In this section, we compare the proposed methodology with an approach based on the well-known curvature scale space (CSS) [Mokhtarian 1995]. To this aim, we use the method presented in [He and Yung 2004, 2008] to detect *dominant points* in an image. According to the terminology used in this paper, stroke corners, junctions, and endpoints are all considered as dominant points. The method described in [He and Yung 2004, 2008] first extracts a binarized edge map from a greyscale image using a Canny edge detector. Neighboring edge pixels are linked to form a set of curves in the image. A two scale CSS is then used to compute an adaptive threshold κ'_T that enables to select dominant points in the image among the curvature maxima. Curvature estimation is given by the CSS using second order finite difference along the curves as in Mokhtarian [1995]. We will henceforth call this method CSS-Canny. We also compare our approach with an hybrid

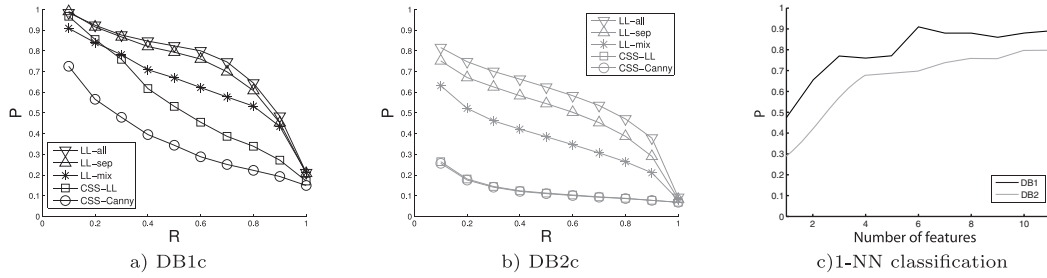


Fig. 17. Mean precision-recall curves over all the queries in database DB1 (a) and in database DB2 (b) using the different approaches presented in Section 4.6.1. c) Precision along the number of features ordered by their Fisher weights, using a 1-NN classification scheme. Using the same numbering as in Table II, DB1 features are here arranged in the following importance order: [1,2,6,9,8,11,5,7,4,3,10]. This means for instance that dealing with line drawings of DB1, the sum of depths of concave stroke corners (10th feature) is the least important feature to classify them. It is worth noticing that DB2 features are arranged along the different following order [11,8,5,4,7,10,1,3,9,6,2].

method, which is the same method as the CSS-Canny, except that Canny edges are replaced by the coalescence sets proposed in this paper. The resulting method is called CSS-LL.

We use CSS-Canny and CSS-LL methods to compute a four dimensional descriptor composed of the same three first features as in Table II using the κ'_T threshold, and the total linear density of all dominant points in the image. Features are normalized, as explained in Section 4.5.

In order to perform a fair comparison between the notion of dominant point introduced in He and Yung [2004] and the characteristic points introduced in this paper, we have to reduce the dimensionality of the feature vector introduced in Section 3.5. We consider a four-dimensional feature vector composed of features 1 to 3 (Table II), and whose fourth feature is computed by summing up features 5,6,7 and 8 (corresponding to characteristic points). These features are extracted on coalescent sets. We will refer to this intermediate method as LL-mix.

The next comparison step is to take into account features 5 to 8 (Table II) individually. This method, henceforth called LL-sep, is equivalent to using $W_{11D} = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]$ in Equation (6). All three methods LL-mix, CSS-Canny and CSS-LL rely on a four dimensional weight vector given by $W_{4D} = [1, 1, 1, 4]$. The last method makes use of all descriptors proposed in Section 3.5, and is called LL-all henceforth. For this complete method, a weight vector given by $W_{11D} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ is used in Equation (6).

The weighted Euclidean distance let us order the database by increasing distances from a given query. For each query, we compute the precision-recall curve, classically defined as follows. Consider a query image belonging to a given class of size C . For a given number W of images returned by this query, W_{tp} is defined as the number of returned images belonging to the same class as the query (*true positives*) among the W returned ones. The precision-recall curve is then obtained by plotting the ratio $P = W_{tp}/W$ (precision-rate) as a function of $R = W_{tp}/C$ (the recall rate) [del Bimbo 1999]. A perfectly discriminative method would provide an horizontal line at $P = 1$. These curves are sampled at some fixed recall abscissas and averaged over the whole database. Figure 17(a) and Figure 17(b) show the mean precision-recall curve over all the queries in DB1 and in DB2.

The comparison between CSS-Canny and CSS-LL first supports the claim that level lines and the selection method proposed in this paper are a great benefit to the geometrical analysis of line drawings. Besides, let us recall that the strokes detection method using level-lines uses only one parameter (ε , fixed to 1 for all images), in comparison to the crucial parameters of a Canny-Deriche detector. The comparison between CSS-LL and LL-mix methods also demonstrates that the proposed method is

better able to detect the dominant points than the method relying on CSS. Performances of LL-sep also show that the classification of dominant points (into endpoints, junction, positive and negative corners) is valuable for correct retrieval. Finally, we also observe that the proposed refinements included in the LL-all method to characterize the geometrical elements in a line artwork (namely curvature zero-crossing, tool radius and depth of corners) significantly increase performances on both databases. CSS-Canny and CSS-LL curves have very similar performance on DB2c (see Figure 17(b)). This is due to the fact that DB2 is based on rasterized vector drawings where strokes hold a very high and constant contrast over an unnoisy background. Canny contours performs as well as level-lines on such drawings, in contrast with results obtained on the Kelly-Matisse-Picasso database.

4.6.2 Examples of Similarity Retrieval. Some examples of retrieval results on DB1 and DB2 are shown on Figure 18. Let us underline that these retrieval examples are obtained using the sole geometry of the drawings, expressed through only 11 features. All query results from databases DB1 and DB2 are available online on a permanent Website [Hurtut 2010].

Let us remark that, depending on the class, some features are more discriminative than others. On Figure 15 for instance, class i is characterized by the linear density of endpoints whereas class d is characterized by the linear density of curvature zero-crossings. To study the influence and the importance of the 11 features, we perform a Fisher discriminant analysis to order the features along their correlation scores with respect to the human classification. On Figure 17(c), a 1-NN classification scheme shows the importance of each feature using the n first most important features, n varying from 1 to 11. This experiment shows that most steps increase the precision. Observe also that the order of the features is different for DB1 and DB2, stressing the fact that the respective importance of features is database dependant.

5. CONCLUSION AND FUTURE WORK

In this article we have introduced a method for the analysis of the pictorial content of line drawings using the geometrical information of stroke contours. These contours are extracted using the concept of meaningful level lines. The main contributions of this article are first a parameter-free filtering method applied to the tree of meaningful level lines and dedicated to the segmentation of line drawings. Next we proposed several effective methods to describe and analyze the geometrical structure of line drawings. Curvature, stroke endpoints, junctions, and corners are extracted from images following the literature in art analysis and visual perception. From these, an intuitive and concise description of the pictorial content of images is built. Third, the efficiency of the proposed features is illustrated using a query by example framework on two databases of line-drawings. The cultural heritage community claims for suited methods that allow to retrieve drawings based on pictorial content criteria [Chen et al. 2005; Stanchev et al. 2006]. This article proposes such a framework for line drawings. The primary extraction of geometrical cues, as illustrated in Figure 16, is also of interest for this community to describe or preprocess drawings towards other application such as statistical studies on a specific artist [Berezhnoy et al. 2009]. This step may also be of strong interest for the Computer Graphics community within the context of Non Photorealistic Rendering [Grabli et al. 2004].

This work opens several perspectives. First, more sophisticated similarity measures to compare line drawings could be used, possibly making use of learning procedures or Earth Mover's Distances. Second, the study presented in this article is limited to line drawings, that is, drawings fully based on 1D content. Our goal is to demonstrate the importance of 1D content independently of other types of primitive. Following the framework proposed by Willats [1997] separating 0D, 1D and 2D content, it would be interesting to further investigate some methods to extract the 1D content of any type of artwork or to combine geometric and texture information for the analysis of line drawings.

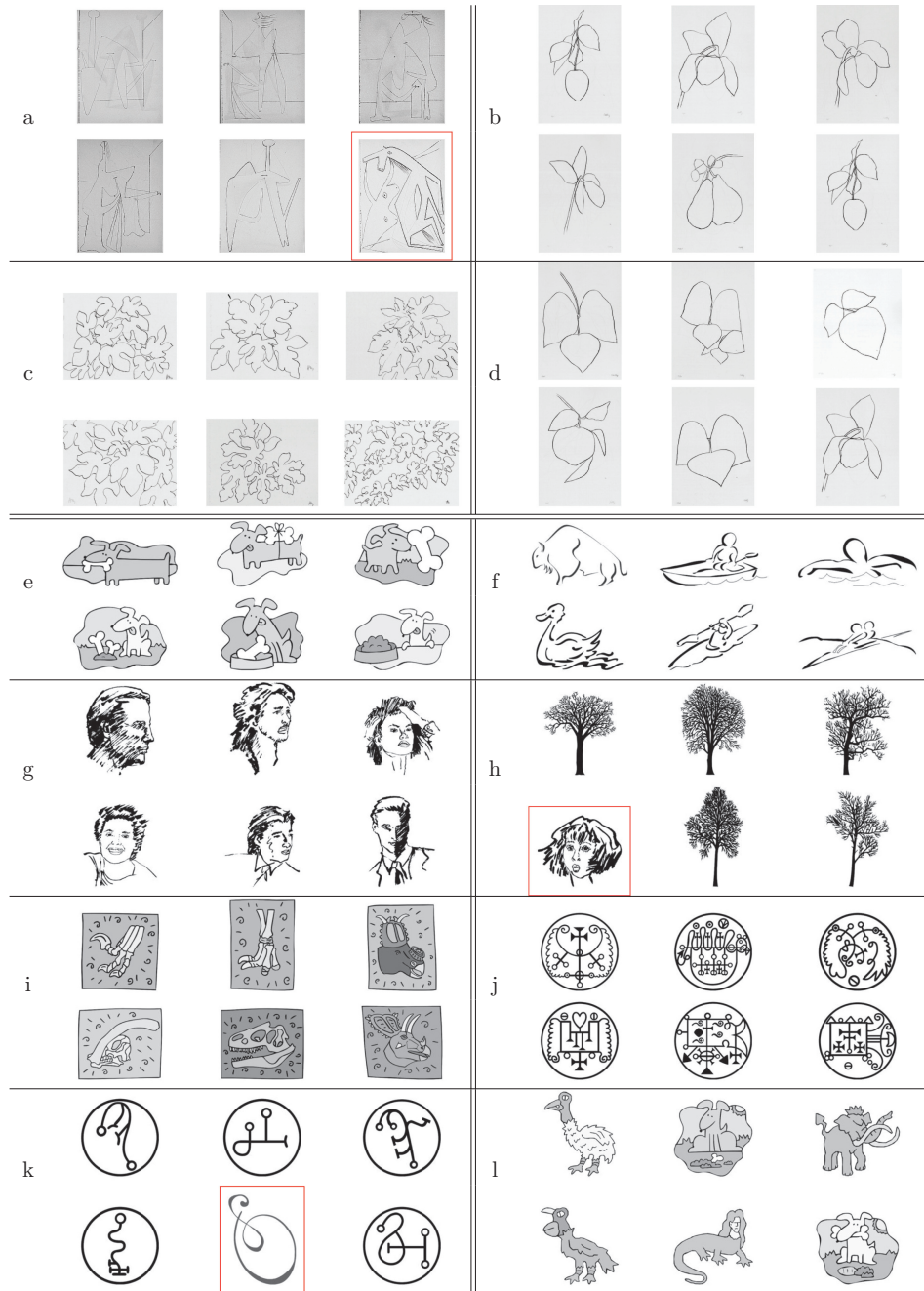


Fig. 18. Examples of drawing retrieval on database DB1 (results *a* to *d*) and on database DB2 (results *e* to *l*). For each example, the image query is in the upper-left corner followed by the five nearest images. These retrieval examples are obtained using the sole geometry of the drawings, expressed through only 11 features. All query results from databases DB1 and DB2 are available online [Hurtut 2010]. Mis-classifications are red-squared. ©Succession Picasso, Musée Picasso Paris (*a*); ©Ellsworth Kelly, Grand Rapids Art Museum (*b, d*); ©Succession H. Matisse, Private Collection (*c*).

ACKNOWLEDGMENTS

We greatly thank the two artists who generously offered their time to classify the databases, as well as the 17 users who participated in the sanity testing.

REFERENCES

- ALVAREZ, L., GUICHARD, F., LIONS, P., AND MOREL, J. 1993. Axioms and fundamental equations of image processing. *Archive Rational Mechanics Anal.* 123, 3, 199–257.
- ARTEXPLOSION. 2002. Hallogram. <http://www.hallogram.com/artexplosion/>.
- BAE, S. AND DURAND, F. 2004. Statistical analysis and transfer of pictorial styles. In *Proceedings of the MIT Workshop, Oxygene*.
- BEREZHNOY, I., POSTMA, E., AND VAN DEN HERIK, H. 2005. Authentic: Computerized Brushstroke Analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'05)*. 1586–1588.
- BEREZHNOY, I., POSTMA, E., AND VAN DEN HERIK, J. 2009. Automatic extraction of brushstroke orientation from paintings. *Mach. Visi. Appl.* 20, 1–9.
- CAO, F. 2003. *Geometric Curve Evolution and Image Processing*. Lecture Notes in Mathematics, vol. 1805, Springer.
- CAO, F., LISANI, J., MOREL, J.-M., MUSÉ, P., AND SUR, F. 2008. *A Theory of Shape Identification*. Lecture Notes in Mathematics. Springer.
- CASELLES, V., COLL, B., AND MOREL, J.-M. 1999. Topographic maps and local contrast changes in natural images. *Int. J. Comput. Vis.* 33, 1, 5–27.
- CHEN, C., WACTLAR, H., WANG, J., AND KIERNAN, K. 2005. Digital imagery for significant cultural and historical materials. *Int. J. Digit. Libr.* 5, 4, 275–286.
- CLOUZOT, H. G. 1954. The mystery of Picasso. DVD.
- DEL BIMBO, A. 1999. *Visual Information Retrieval*. Morgan Kaufman Publishers.
- DERICHE, R. 1987. Using Canny's criteria to derive recursively implemented optimal edge detector. *Int. J. Comput. Vis.* 1, 167–187.
- DESOLNEUX, A., MOISAN, L., AND MOREL, J.-M. 2001. Edge detection by helmholtz principle. *J. Math. Imag. Vis.* 14, 271–284.
- DUPUIS-LABBÉ, D. 2006. Les carnets de picasso. DVD-Rom, Réunion des Musées Nationaux (RMN).
- EAKINS, J., BRIGGS, P., AND BURFORD, B. 2004. Image retrieval interfaces: A user perspective. In *Proceedings of the 3rd International Conference on Image and Video Retrieval*. Springer, 628–637.
- FANTONI, C., BERTAMINI, M., AND GERBINO, W. 2005. Contour curvature polarity and surface interpolation. *Vis. Res.* 45, 1047–1062, 7.
- FARINE, M. 2005. Klimt, dessins érotiques. *Oeil-Lausanne Then Paris*, 567, 64.
- FAYOLLE, J., RIOU, L., AND DUCOTTER, C. 2000. Robustness of a multiscale scheme of feature points detection. *Patt. Recog.* 33, 9, 1437–1453.
- FELDMAN, J. AND SINGH, M. 2005. Information along contours and object boundaries. *Psych. Rev.* 112, 1, 243–252.
- GHOSH, A. AND PETKOV, N. 2006. Effect of high curvature point deletion on the performance of two contour based shape recognition algorithms. *Int. J. Patt. Recog. Artif. Intell.* 20, 6, 913–924.
- GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. X. 2004. Programmable Style for NPR Line Drawing. In *Proceedings of the Eurographic Symposium on Rendering (EGSR'04)*. 33–44.
- GU, Y. H. AND TJAHHADI, T. 2000. Coarse-to-fine planar object identification using invariant curve features and B-spline modeling. *Patt. Recog.* 33, 9, 1411–1422.
- HE, X. AND YUNG, N. 2004. Curvature scale space corner detector with adaptive threshold and dynamic region of support. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*. Vol. 2.
- HE, X. AND YUNG, N. 2008. Corner detector based on global and local curvature properties. *Optic. Engin.* 47, 5, 057008.
- HURTUT, T. 2010. Last updated <http://www.math-info.univ-paris5.fr/hurtut/linedrawings/>.
- HURTUT, T., GOUSSEAU, Y., CHERIET, F., AND SCHMITT, F. 2008. Pictorial analysis of line-drawings. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization and Imaging*.
- JODOIN, P.-M., EPSTEIN, E., GRANGER-PICHÉ, M., AND OSTROMOUKHOV, V. 2002. Hatching by example: a statistical approach. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering (NPAR'02)*. 29–36.
- JOHNSON JR, C., HENDRIKS, E., BERZHNOY, I., BREVDO, E., HUGHES, S., DAUBECHIES, I., LI, J., POSTMA, E., AND WANG, J. 2008. Image processing for artist identification. *IEEE Signal Process. Mag.* 25, 4, 37–48.
- JULESZ, B. 1986. Texton gradients: The texton theory revisited. *Biol. Cybernet.* 54, 4, 245–251.
- ACM Journal on Computing and Cultural Heritage, Vol. 4, No. 1, Article 3, Publication date: August 2011.

- KAMMERER, P., ZOLDA, E., AND SABLATNIG, R. 2003. Computer aided analysis of underdrawings in infrared reflectograms. In *Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. 19–27.
- KANDINSKY, W. 1979. *Point and Line to Plane*. Dover Publications.
- KLEE, P. 2004. *Paul Klee, cours du Bauhaus*. Hazan.
- LABRUSSE, R. AND DE CHASSEY, E. 2002. *Henri Matisse and Ellsworth Kelly—Dessins de plantes*. Gallimard, Centre Pompidou.
- LAM, L., LEE, S., AND SUEN, C. 1992. Thinning methodologies-A comprehensive survey. *IEEE Trans. Patt. Anal. Mach. Intell.* 14, 9, 869–885.
- LEYTON, M. 2006. *The Structure of Paintings*. SpringerWienNewYork.
- LI, J. AND WANG, Z. 2004. Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Trans. Image Process.* 13, 340–353.
- LOUVRE. 2010. Last updated, Arts graphiques dpt. louvre. <http://arts-graphiques.louvre.fr/>.
- MARJI, M. AND SIY, P. 2003. A new algorithm for dominant points detection and polygonization of digital curves. *Patt. Recogn.* 36, 10, 2239–2251.
- MOISAN, L. 1998. Affine plane curve evolution: a fully consistent scheme. *IEEE Trans. Image Process.* 7, 3, 411–420.
- MOKHTARIAN, F. 1995. Silhouette-based isolated object recognition through curvature scale space. *IEEE Trans. Pattern Anal. Mach. Intell.* 17, 5, 539–544.
- MONASSE, P. AND GUICHARD, F. 2000. Fast computation of a contrast-invariant image representation. *IEEE Trans. Image Process.* 9, 5, 860–872.
- MUSÉ, P., SUR, F., CAO, F., GOUSSEAU, Y., AND MOREL, J.-M. 2006. An *a contrario* decision method for shape element recognition. *Int. J. Comput. Visi.* 69, 3, 295–315.
- ONKARAPPA, N. AND GURU, D. 2007 Modified 9DLT Matrix for Similarity Retrieval of Line-Drawing Images. In *Proceedings of the Conference on Pattern Recognition and Machine Intelligence (PRMI'07)*. 136–143.
- OWEN, P. 2007. Painting. In *Encyclopaedia Britannica*.
- PAVLIDIS, T. 1993. Recognition of printed text under realistic conditions. *Patt. Recogn. Lett.* 14, 4, 317–326.
- PICKETT, J., ET AL. 2000. *The American Heritage Dictionary of the English Language*. Houghton Mifflin.
- PLAMONDON, R. AND PRIVITERA, C. M. 1999. The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information. *IEEE Trans. Image Process.* 8, 1(Jan), 80–91.
- POYATO, Á. C., GARCÍA, N. L. F., CARNICER, R. M., AND MADRID-CUEVAS, F. J. 2004. A method for dominant points detection and matching 2D object identification. In *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR)*. 424–431.
- SCHOMAKER, L. 1991. *Simulation and Recognition of Handwriting Movements: A Vertical Approach to Modeling Human Motor Behavior*. Nijmeegs Instituut voor Cognitie-onderzoek en Informatietechnologie.
- SERRA, J. 1982. *Image Analysis and Mathematical Morphology*. Academic Press.
- STANCHEV, P., GREEN JR, D., AND DIMITROV, B. 2006. Some Issues in the Art Image Database Systems. *J. Digit. Inform. Manage.* 4, 4, 227.
- TOMBRE, K. 2006. Graphics recognition: The last ten years and the next ten years. In *Lecture Notes In Computer Science*, vol. 3926, 422.
- TOMBRE, K. AND CHHABRA, A. EDS. 1998. *Proceedings of the International Workshop Graphics Recognition Algorithms and Systems*. Vol. 1389. Springer.
- VAN DEN HERIK, H. J. AND POSTMA, E. O. 2000. Discovering the visual signature of painters. In *Future Directions for Intelligent Systems and Information Sciences*, Physica-Verlag, 129–147.
- VAN SOMMERS, P. 1984. *Drawing and Cognition: Descriptive and Experimental Studies of Graphic Production Processes*. Cambridge University Press.
- WILLATS, J. 1997. *Art and Representation*. Princeton University Press.
- WINTER, J., PANIS, S., AND WAGEMANS, J. 2002. Perceptual saliency of points along the contour of everyday objects: A large-scale study. *J. Vision* 2, 7, 487.
- WU, W. 2003. An adaptive method for detecting dominant points. *Patt. Recogn.* 36, 10, 2231–2237.
- YAN, Y. AND JIN, J. 2006. Indexing and retrieving oil paintings using style information. In *Proceedings of the 8th International Conference on Visual Information and Information Systems*, Lecture Notes In Computer Science, Vol. 3736, 143.

Received June 2010; revised December 2010; accepted February 2011