Pictorial content analysis of artistic line-drawings using geometrical information

Analyse du contenu pictural de dessins artistiques à partir d'informations géométriques

Thomas Hurtut
Yann Gousseau
Farida Cheriet
Francis Schmitt

2008D008

Juin 2008

Département Traitement du Signal et des Images
Groupe TII : Traitement et Interprétation des Images

Analyse du contenu pictural de dessins artistiques à partir d'informations géométriques. Pictorial content analysis of artistic line-drawings using geometrical information

Thomas Hurtut, Yann Gousseau, Farida Cheriet, and Francis Schmitt

Résumé

Cet article propose un cadre général pour l'analyse du contenu pictural des dessins artistiques au trait. Le terme contenu pictural recouvre à la fois le contenu stylistique et les caractéristiques visuelles des sujets représentés. Tout d'abord, nous proposons une méthode automatique d'extraction des contours des traits dans les dessins. Cette méthode repose sur un filtrage de la structure hiérarchique des lignes de niveau de l'image. Puis le rayon de l'outil graphique est estimé à partir de ces contours. Cette information permet ensuite de paramétrer une étape d'extraction d'attributs géométriques, en particulier de statistiques sur la courbure des traits, de points terminaux, de jonctions et de coins. La pertinence des méthodes proposées est illustrée à l'aide de plusieurs expériences sur des bases de dessins classées. Les résultats sont comparés à ceux obtenus à l'aide d'une méthode reposant sur l'espace-échelle de courbure. Enfin, des expériences de recherche par le contenu suggèrent que l'approche proposée permet de rendre compte de l'effet visuel produit par un dessin au trait sur un observateur.

Abstract

In this paper, a general framework for the analysis of the pictorial content of artistic line-drawings is introduced. The pictorial content is 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 hierarchical sets of level lines. 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 strokes curvature, endpoints, junctions and corners. These geometrical components actively participate to the visual effect delivered by the pictorial content. The efficiency of the proposed methods is illustrated with several experiments on a classified database of artistic line-drawings, and compared with an approach based on the curvature scale space (CSS). Retrieval experiments also suggest that the proposed framework is able to handle the visual effect delivered by a line-drawing to a human observer.

Index Terms

pictorial content; style analysis; line art; line-drawings; image line pattern analysis; image edge analysis; topographic map; level lines.

T. Hurtut is with the Ecole Polytechnique de Montréal and TELECOM ParisTech.
Y. Gousseau and F. Schmitt are with TELECOM ParisTech.
F. Cheriet is with the Ecole Polytechnique de Montréal.

I. Introduction

CULTURAL institutions around the world have, over the past two 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 150000 drawings [1]). Works of visual art 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 to 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" [2]. In this paper, we call *pictorial content* the combination of the artistic style and the visual features formed by the set of strokes composing an artistic line-drawing. The analysis of the pictorial content is useful to retrieve artworks delivering a similar visual effect to the observer [3]–[5]. The visual effect considered here corresponds to a *preattentive* perception [6] prior to any recognition by the visual system of the possible realistic subjects depicted in the artwork. It is also of high interest to enable transfering of some pictorial characteristics from real artworks to non-photorealistic rendered (NPR) drawings in computer graphics [7]–[9].

A. Problem statement and paper scope

The pictorial content is partially related to the artistic style. The style detection problem has been tackled in many references, see e.g. [10]–[12]. 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 Dictionary [13], style is "the combination of distinctive features of artistic expression, execution or performance characterizing a particular person, group, school or era". Unfortunately style depiction is often based on the same visual effects as subject depiction. Style recognition is thus a very difficult task, requiring the knowledge of numerous art historians and experts. An artwork from the blue period of Picasso for instance is recognizable not only because of its blue tonality, but because of many semantical 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 [14] is very large in visual arts images since artworks are often not perfectly realistic. 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.

Line-drawing practice has long been considered as one of the most important requirements in artist works [15], [16]. 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 [17], 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 [18]. 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 (see e.g. [19], [20]).

We propose in this paper the following contributions:

- an approach to detect closed contours using a level lines selection method. This selection method uses the probabilistic background of Desolneux et al. [21] which is recalled in Section II-A. Inspired by this framework, we propose a new algorithm to filter the level lines tree structure specifically suited to line-drawings (Section II-B).
- a set of methods to extract and characterize the following geometrical features: curvature zero-crossings, corners, junctions, endpoints (Section III).

Similarity retrieval experiments with a classified database of line-drawings will be used to demonstrate the efficiency of the features (Section IV). We will compare our method to an approach which uses the curvature scale space (CSS) to detect geometrical points of interest. This article extends a previous conference paper [22], providing more detailed description and quantitative results.

B. Related work

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 [23] or estimated by a multilevel thresholding [24]. These approaches are adapted to very small strokes such as Van Gogh's. Li and Wang studied ancient black and white ink drawings using wavelets and hidden Markov models [11]. Their method is yet close to texture analysis in paintings such as in the approaches developped in [25]. Onkarappa and Guru studied the spatial mutual arrangement of strokes in line-drawing images to achieve similariy retrieval [26].

2) Graphics recognition: Analysis of technical drawings such as maps and engineering drawings has received considerable attention during the last decades [27]. Next open issues in this field are presently vectorization and symbol recognition of handmade sketchy drawings [28]. 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 [28].

3) Cursive and offline writer recognition: The neurobiomecanical process of writing is close to drawing. Cursive alphabets are 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. The line *alphabet* of an artist is however infinite. The cognitive process of drawing and writing are moreover quite different. Writing is a spontaneous task involving no continuous judgment of the written [29] whereas an artist has to continuously analyze his drawing while doing it [30]. Nevertheless, prior to recognize 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 [31] is classically used. Due to the lack of one unique mathematical definition, there exists hundreds of skeletonization methods. Skeletonization usually slightly distorts strokes [32]. These methods are moreover quite sensitive to local geometric variations. We thus choose to rely on stroke contours.

II. 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 [33] 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 [34], [35]. More precisely, we make use of several contributions on the computation [36], filtering [21] and regularization [37] of level lines. The basic elements on which we build our analysis of line drawings are the *meaningful lines* as defined by Desolneux et al. [21]. These are also the building blocks of several recent works on shape recognition [38], [39]. The corresponding notions are recalled in Section II-A. Then, in Section II-B, 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.

A. Background on the topographic map and meaningful level lines

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 $\lambda \in \mathbb{R}$ [34]. Levels 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 in Figure 1.

The whole family of the level lines of an image is called the topographic map [35]. It can be efficiently extracted from an image using the Fast Level Lines Transform (FLLT) [36]. 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 [34]. 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 [36], [40]. Last,

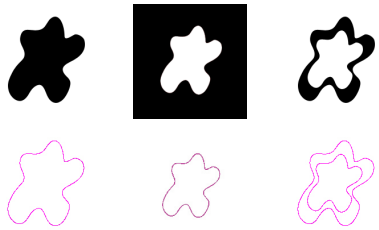


Fig. 1. Negative and positive level lines. Top row shows synthetic images and bottom row shows the level lines corresponding to the central object. Left, the contrast sign when going toward the interior of the level line is negative : this is a negative line. Middle, opposite situation, the level line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

object contours often locally coincide with level lines [34], [39]. 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 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).

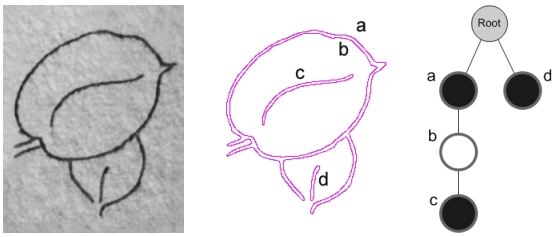


Fig. 2. A simple line-drawing representing two leaves (left). Four level lines are manually chosen to best represent the strokes contours (middle). On the tree structure of these four level lines (right), a positive and a negative level line are represented by a white and a black node respectively. The unique positive level line *b* describes the inner contour of the biggest leaf. The first purpose of this paper is to propose a method for the automatic selection of such lines.

The first step in order to reach a representation such as the one shown in Figure 2 is to consider lines that are long and contrasted enough, the so-called *meaningful level lines* as defined by Desolneux et al. [21] and briefly presented in the next section.

2) Meaningful level lines: In order to filter the redundant set of level lines of images (the topographic map), Desolneux et al. make use of the general principle of *a contrario* methods. The basic idea is to keep only lines that would be very unlikely in a white noise situation. For each line $\mathcal{L}$ is computed a quantity, called *number of false alarms*, $NFA(\mathcal{L})$, that depends both on the length and on the contrast of the line $\mathcal{L}$. This quantity corresponds to the expected number of lines having better contrast and greater length than $\mathcal{L}$ to be encountered in a white noise image. The smaller this value, the more meaningful the line. Following Desolneux et al. definition, a level line $\mathcal{L}$ is said to be $\varepsilon-$meaningful if $NFA(\mathcal{L}) < \varepsilon$. In practice, the value of ε can safely be fixed to one [21], [41] (that is, the expectation of the number of false detections is fixed to one), in which case we simply call the corresponding lines *meaningful level lines*. In the rest of the paper, we consider such lines, that is, we consider only the value $\varepsilon = 1$. Such 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 lines, the authors of [21] introduced the concept of *maximal* meaningful lines, as recalled in the next section.

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 $\{\mathcal{N}_i\}_{i=1\ldots n}$ of the tree where all nodes are of the same type

Fig. 3. The set of meaningful level lines from the same drawing as in 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 grey levels of the original drawing are coded on 8 bits.

(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 [21] 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, resumed in 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 5c). To cope with this fact, we introduce in the next section a new maximality principle, specifically adapted to strokes in line-drawings.

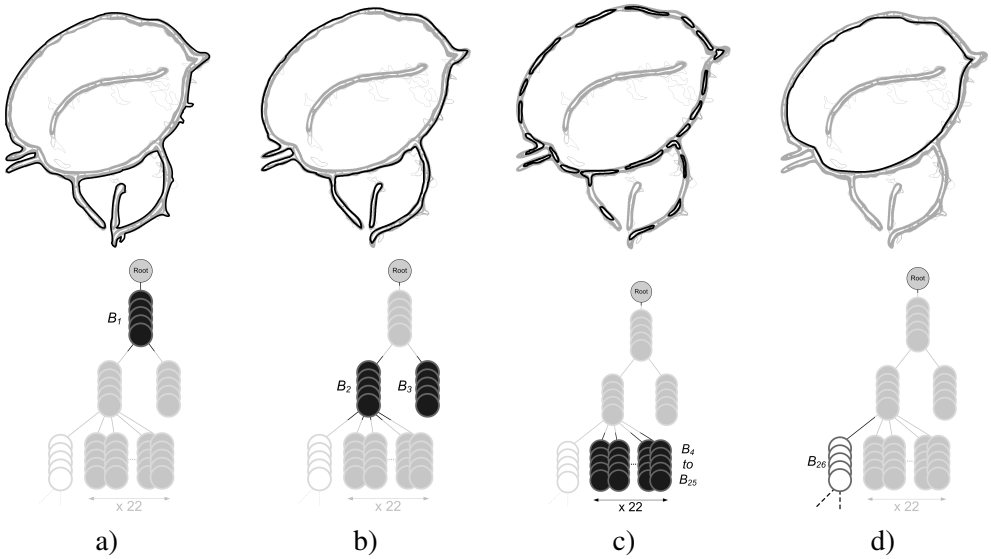


Fig. 4. Tree-structure of the set of meaningful level lines. Each series of nodes represents a maximal monotone branch, which is resumed in the drawing by a thick black line. a) At the first level, a maximal monotone branch encompasses all the strokes. b) The last node of the first branch encompasses 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 $B_2 6$ 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.

B. Stroke cluster extraction

We first propose in Section II-B.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 II-B.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.

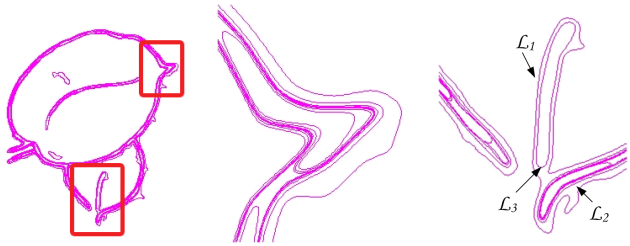


Fig. 5. Same example as on Figure 3 using Desolneux et al. maximality principle (left). Two details are also displayed (center and right). The resulting set contains 54 level lines. The number of lines has been dramatically reduced, but redundancies remain along stroke contours.

1) A new maximality principle: MMT-filtering: The geometrical analysis and description that we will present in Section III rely on stroke contours. It is therefore essential that each contour be described by a unique level line. To eliminate all redundancies we propose a novel maximality principle which takes advantage of the topological structure of line-drawings.

Henceforth we call maximal monotone tree (MMT) a subtree of the set of meaningful level lines than contain strictly positive or negative nodes and which cannot be included in another monotone subtree. In 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 and positive contrast variation occurs inside a stroke. This would corresponds to some artefact. 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* in Figure 2 for instance). This means that redundancies of one contour are grouped inside one MMT. Yet, different contours can be grouped in the same MMT. 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* in Figure 2) and the contour of the small vein of the small leaf (line *d* in Figure 2) are in the branches B_2 and B_3 of the first MMT, respectively. Let $\mathcal{L}$ be a level line that corresponds to a stroke contour. Due to the topologic constraints induced by a line drawing structure, we can prune the children of $\mathcal{L}$ since they should describe the same stroke. We can also prune parents of $\mathcal{L}$ since they either describe the same stroke or embrace several strokes inside the MMT.

Inspired from the previous remarks, we propose the following method to filter the tree of lines, called MMT-filtering. For each MMT of the tree of meaningful level lines, we first look for the most meaningful level line $\mathcal{L}$ (i.e. the level line holding 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 below. Applied to the tree-structure presented on Figure 3, this algorithm returns the level lines which are to be seen in Figure 2b. The maximality principle is detailed in Figure 6 for the first MMT.

Algorithm 1 MMT-filtering

Require: Meaningful level lines tree S
Ensure: Filtered level lines tree
Main function:
for all maximal monotone tree S_i of S **do**
filter(S_i)
end for
Function **filter**(S_i)
while S_i is not empty **do**
- $\mathcal{L} \leftarrow$ 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

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

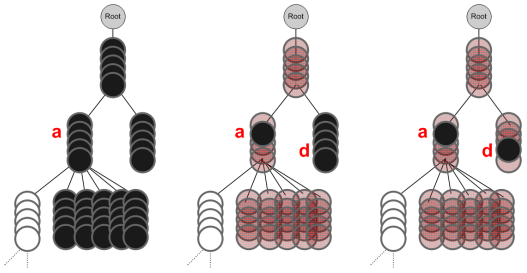


Fig. 6. MMT-filtering applied to the first maximal monotone negative subtree of the tree shown in Figure 4. Left: the most meaningful level line a is found. (a refers to the corresponding label in Figure 2). Middle: level lines that are above or below this level line in the tree are removed from the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

called **coalescence sets**. We make the hypothesis that we deal with drawings made of dark strokes over a bright background. In the (rare) case of drawings made of bright strokes over a dark background, the negative image can 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.

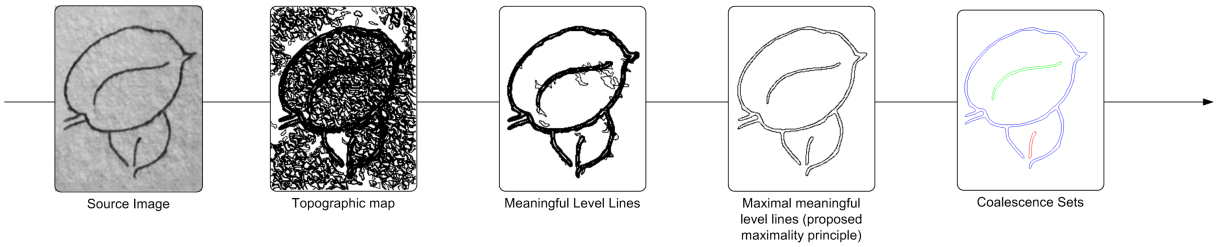


Fig. 7. Overall detection framework of stroke contours. First, the topographic map of the numerical image is computed. The meaningful level lines, as recalled in Section II-A.2, are then extracted. The MMT-filtering method introduced in Section II-B.1 next eliminates redundancies. Cluster of strokes are finally segmented by grouping level lines into coalescence sets. Our contributions to this framework are the last two steps.

III. 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. This information is crucial to tune several geometrical parameters in the following steps. Another important information to compute is curvature along stroke contours. This allows us to detect and analyze curvature zero-crossings, stroke junctions, corners and endpoints.

A. 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 [42]. Instead, affine plane curve smoothing is chosen [43] which consists in applying the following partial differential equation (PDE) on a level line $s \mapsto \mathcal{L}(s, 0)$:

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

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 IV-B. Experiments are done with the practical implementation of Moisan [37] 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 multi-scale approach such as [44], where the smoothing acts as an analysis tool.

B. Tool radius estimation

In order to estimate the size of the drawing tool, we make the following hypothesis: each stroke cluster Φ has been made with tools having the same radius R_Φ. This may sound restrictive but line-drawings are often made all over with one single tool. This hypothesis let us easily estimate the tool radius. Following 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 given by:

$$R_\Phi \approx \frac{surface}{perimeter} = \frac{-\sum_{i=1}^{n} sign(\mathcal{L}_i) S_i}{\sum_{i=1}^{n} P_i}, \tag{2}$$

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

C. Curvature computation

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

$$\kappa(i) = \theta'(s_i) \approx \frac{\theta_i - \theta_{i-1}}{(s_{i-1} - 2s_i + s_{i+1})/2}, \tag{3}$$

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 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 shifting along a closed curve (clockwise or anticlockwise) and the local geometry (convex or concave). We choose the following convention. Negative level lines will be travelled clockwise. This induces negative curvature values at line convexities and positive values at line concavities. On the opposite, positive level lines will be travelled anticlockwise to invert the curvature sign. This convention first 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 8). Endpoints always correspond to negative curvature values, and regions where strokes create a non-reflex angle (e.g. at junctions) hold positive values.

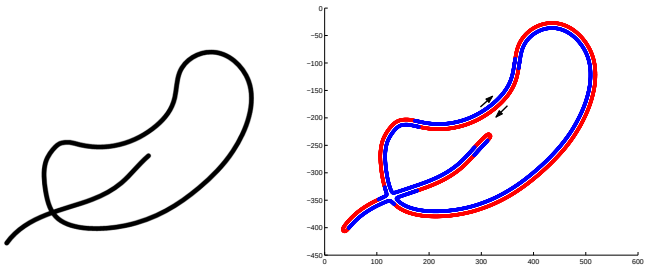


Fig. 8. For computing curvature, a convention is chosen so that negative (resp. positive) level lines are travelled clockwise (resp. anticlockwise). On this example (left), one negative level line delimits the outer contour, and one positive level line delimits the inner contour (right). Arrows notify the travel direction. This convention induces negative curvature values (red lines) and positive values (blue lines) on inner and outer stroke cluster contour respectively. This ensures that pictorial elements such as stroke endpoints or junctions have the same local curvature sign.

Curvature zero-crossings contribute to artistic visual impression and object recognition [45]. They are estimated directly on the curvature signal. An example of a drawing with detected curvature zero-crossing is shown in Figure 9b.

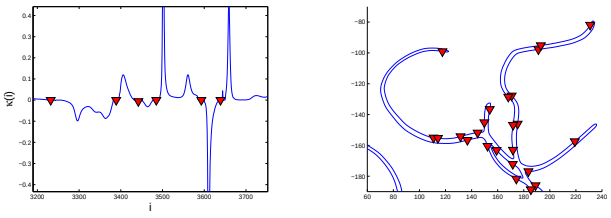


Fig. 9. Left: curvature signal $\kappa(s_i)$ along a stroke contour. Triangles represent estimated zero-crossings. Right: maximal meaningful level line from a drawing, and its superposed zero-crossings.

D. Detection of endpoints, junctions and corners

This section proposes to detect three different strong visual cues: stroke junctions and stroke corners, stroke endpoints, according the taxonomy presented in Figure 10-I. These features all correspond to points with high curvature values. We thus first extract such points before classifying them.

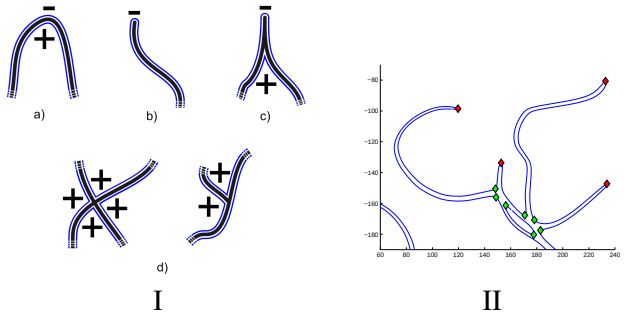


Fig. 10. Left: taxonomy of high curvature values along stroke contours: a) a simple high curvature point along the gesture trajectory, b) an endpoint, c) a stroke corner, d) two types of stroke junctions: X and T junctions. Plus and minus sign refers to the curvature sign on the stroke contour. Strokes are represented by black lines and the stroke contours correspond to the blue surrounding lines. Right: curvature maxima detected with the method presented in Section III-D.1. Red (resp. green) diamonds are positive (resp. negative) maxima.

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 in Figure 10-I. This is related to dominant points detection in pattern recognition. We use the curvature information to extract such points. Similar approaches can be found in [46]–[48]. Other approaches use local measures which are different from curvature [49], [50], and Gaussian scale spaces [44] or wavelets [51].

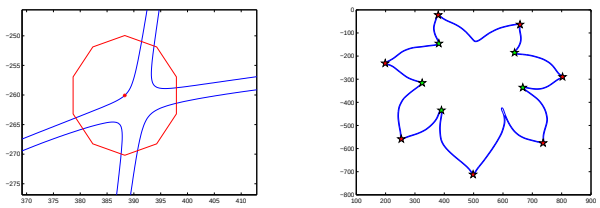
Recall that R_Φ is the tool radius. We iteratively consider every continuous portion of the curvature signal where $|\kappa(i)| > \kappa_t$ with $\kappa_t = 1/(k_c R_\Phi)$. We will discuss the choice of k_c in Section IV-D. Zero-crossings of the derivative curvature signal on each of these portions 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 10-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 III-E.

Depending on the curvature sign, an extremum can be either positive or negative. An example of extrema detection is shown in Figure 10-II. Thanks to the convention on curvature presented in Section III-C, stroke endpoints coincide with negative maxima (Figure 10-Ib), and junctions coincide with several positive maxima (Figure 10-Id). Corners have one positive maximum on the concave side, and one negative maximum on the convex side (Figure 10-Ic). This information will be useful in the next three sections to analyze stroke endpoints, junctions and corners.

2) Stroke junctions: Stroke junctions are important geometrical characteristics of the 1D content of a drawing. Each junction indicate a possible occlusion denoting an actual level of perspective in the depicted scene [52]. 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 11a). 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 IV-E.

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 [53] (see Figure 11b 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 [54]. 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 other 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.



a) b)

Fig. 11. Left: to detect a stroke junction, a disk is centred on the positive curvature maximum. In this example, four different groups of neighbouring points are covered by the disk. Right: stroke corners detected on a drawing. Red stars (resp. green) are convex line corners (resp. concave).

If the underlying stroke describes a form that is perceptually visible such as an object contour, a line corner can be convex or concave, inducing a different pictorial effect. This type of effect has been recently studied in [55], [56]. 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 and objects described by an open contour also create some ambiguities. We propose a well-defined way to precise 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 barycenter 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.

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 $3R_\Phi$, 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 R_Φ estimation. 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 12. We do not consider here the disk previously used in Section III-D.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 IV-E, k_e will be empirically set to the same value as k_j.

E. 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 I. 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

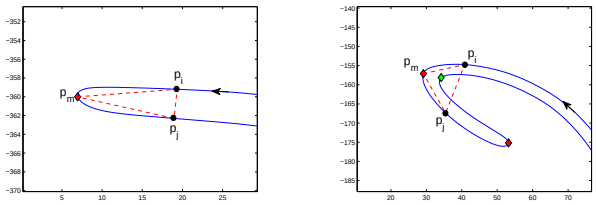


Fig. 12. Left: negative maximum is classified as a stroke endpoint. Right: some points p_q which do not belong to the level line portion $[p_i, p_j]$, are included in the triangle $p_i p_m p_j$. This maximum is thus not classified as a stroke endpoint.

TABLE I
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 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.

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 by Julesz for perceptive *preattentive* discrimination of textures [6]. We have conducted several experiments on Julesz textons using the methods proposed in this paper. They led to similar perceptive results.

IV. EXPERIMENTS AND RESULTS

Experiments are first performed on several synthetic examples in Sections IV-A to IV-E. These sections study the stroke contour detection method, the curvature estimation precision, the tool radius estimation, and the setting of parameters k_c, k_e and k_j. Section IV-F presents a classified database of real drawings which will is used in Section IV-G to evaluate the proposed line-drawing analysis method. In all our experiments, images are coded on 8bits and the quantification step used to extract the topographic map is 1.

A. Stroke contours detection

Stroke contours are detected using the framework presented in this paper and resumed in Figure 7. Stroke contours are represented by meaningful level lines filtered as explained in Section II-B. 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 one proposed in this paper is 88%.

A level line passes through constant values of grey levels. What happens along strokes where the grey 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. In Figure 13a, several synthetic strokes with varying grey values are drawn on a white background, with added Gaussian noise ($\sigma = 5$). In Figure 13b 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 grey levels. In Figure 13c, our detection method selects the most meaningful contour, except for the bottom stroke which might be perceptually visible although its minimum grey level value reaches the background value. In Figure 13d 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.

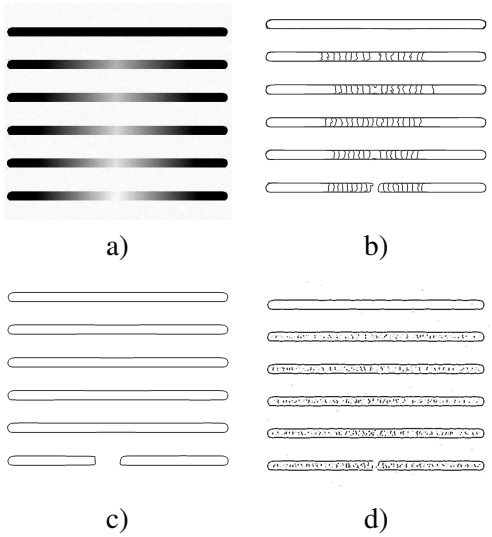


Fig. 13. a) Several synthetic strokes are drawn on a white background, with added Gaussian noise ($\sigma = 5$). The top stroke has a constant grey level value of 0. Next strokes have a grey level value of 0 at their endpoints which linearly increases to respectively 215, 225, 235, 245 and 255 at their 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 grey 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.

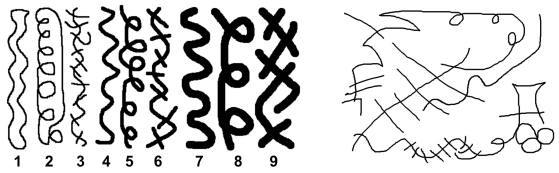
B. Curvature estimation

Level lines are smoothed following the affine invariant curve smoothing recalled in Section III-C. We choose the practical implementation of [37] which preserves all theoretical properties (affine and morphological invariance) 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. Considering the mean spatial resolution of the digitized drawings in our database ($4pixels/mm^2$), this means that the PDE smoothes details of size less than $0.25mm^2$. This implementation also give control over the sampling step along level lines. This quantization step is fixed to $\Delta s = 0.5$ pixels for all experiments in this paper.

C. Tool radius estimation

The tool radius is an important measure on which depend most of the features proposed in this paper to analyze the geometrical content. 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 in Figure 14a and presents three groups of strokes with different radius: 3, 7 and 15 pixels. The tool radius is estimated using Equation 2. The average error on this example is 12.5%. Line drawings often include very long strokes and complex clusters. Such a situation happen in Figure 14b. The error on this example is 1.8%.

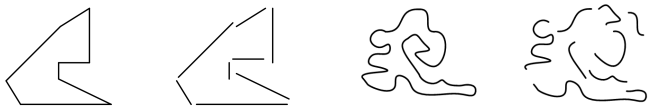


a) b)

Fig. 14. Two synthetic drawings made with the Gimp software. The tool radius is calibrated and estimated using the method proposed in Section III-B.

D. Setting of the curvature threshold κ_t

Recall that the threshold $\kappa_t = 1/(k_c R_\Phi)$ aims at selecting high curvature points (see Section III-D.1), as well as selecting flat parts as explained in Section III-E. To fix this parameter, four groups of 5 synthetic images have been used. Each group is made of one drawing repeated using 5 different tool radius $R_\Phi = 3, 5, 9, 13$ or 17 pixels. These drawings are made with a vector-based drawing program. The four drawings made with $R_\Phi = 5$ are shown in Figure 15. Figure 15a shows a synthetic drawing made of one closed stroke whose *flat parts* holds strictly null curvature values. Figure 15c is more realistic a synthetic drawing with non-zero curvature values. Figure 15b-d, the same drawings have been cut in several parts, introducing many endpoints. All images are rasterized to the same resolution. Ideally, κ_t (or equivalently k_c) is set in such a way that the three first features of Table I are equal for drawings of type *a-b*, and *a-b* respectively, whatever their tool radius. This is not true when k_c is too low since endpoints regions are then integrated in the distribution of points used to characterize flat parts. On the opposite, raising k_c to a too high value raises the risk to not discriminate between a drawing holding strictly straight flat parts (Figure 15a, b), and others (Figure 15c, d). A good tradeoff is found at $k_c = 5$. This value also gave us reasonable results with the real drawings to be presented in Section IV-F. Henceforth in the experimental section, we will thus use the value $\kappa_t = 1/(5R_\Phi)$.



a) b) c) d)

Fig. 15. Synthetic images used to set the value of κ_t. Figures a) and c): synthetic drawings made of one closed stroke using $R_\Phi = 5$. Figures b) and d): the same drawings have been cut in several parts, introducing many endpoints. Each drawing is made with a vector-based drawing program and repeated using 5 different tool radius. All images are then rasterized to the same resolution.

E. Setting of the endpoints and stroke junctions k_e and k_j threshold values

Recall that endpoints are detected using region with radius $k_e R_\Phi$ around a maximum candidate (Section III-D.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 classify some elongated junctions of two strokes as an endpoints. The synthetic drawing shown on Figure 16a illustrates the trade-off between these two opposite effects. This drawing and extensive empirical experiments with real drawings led us to choose $k_e = 5$.

Stroke junctions are detected using a disk with radius $k_j R_\Phi$. Increasing k_j raises the risk to consider too large regions that covers more than one stroke junction, yielding false detections. Lowering k_j raises the risk to miss some stroke junction. The theoretical value of k_j depends on the local geometry of the junction and the relative stroke angles. Using the image shown on Figure 16b and many real drawings, we empirically found a good trade-off

between these two opposite effects using the value $k_j = 5$. It is not surprising that k_j and k_e hold the same value since they are related to the same local scale of extremum neighbourhood. In the remaining part of this paper, we will thus use these two values.

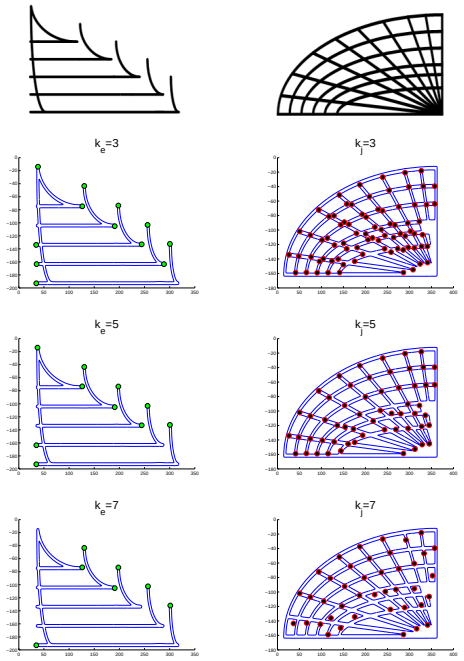


Fig. 16. Top row, synthetic images used to empirically set the values of k_e (left) and k_j (right). Below are detection results with $k_e, k_j \in \{3, 5, 7\}$

F. A classified line-drawings database

In the next section, experiments are performed on a database made of 105 line-drawings. One example of analyzed drawing is to be seen in Figure 18. This database has been manually classified by a professionnal artist into thirteen categories according to the pictorial impact. Half of this database (6 classes) comes from the catalogue of an exhibition of drawings made by Henri Matisse and Ellsworth Kelly [20]. These drawings have been scanned with an HP Scanjet 8200 with 4800 ppi resolution. The real size of drawings can be known using metadata. 21 images (4 classes) come from several Picasso sketch books released on DVD-Rom in 2006 as a 800×600 digital facsimile edition [57]. Finally, 3 classes (31 drawings) are made of computer drawings. Samples of these classified database are to be seen in Figure 17.

The 13 features presented in Table I are extracted from 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 features are computed for every artwork in the database, it is centered by the mean and normalized by the standard deviation of the feature distribution over the database. Let $V^a(k), k \in [1, 11]$ be the eleven normalized features of an artwork a, then the similarity measure $d(V^a, V^b)$ between two drawings is the following weighted Euclidean distance:

$$d(V^a, V^b) = \sqrt{\sum_{k=1}^{11} w_k (V^a(k) - V^b(k))^2}, \tag{4}$$

where $W_{11D} = (w_1, \ldots, w_{11})$ is a 11-dimensionnal weight vector. Thanks to the small size of the feature vector, a query runs almost instantaneously.

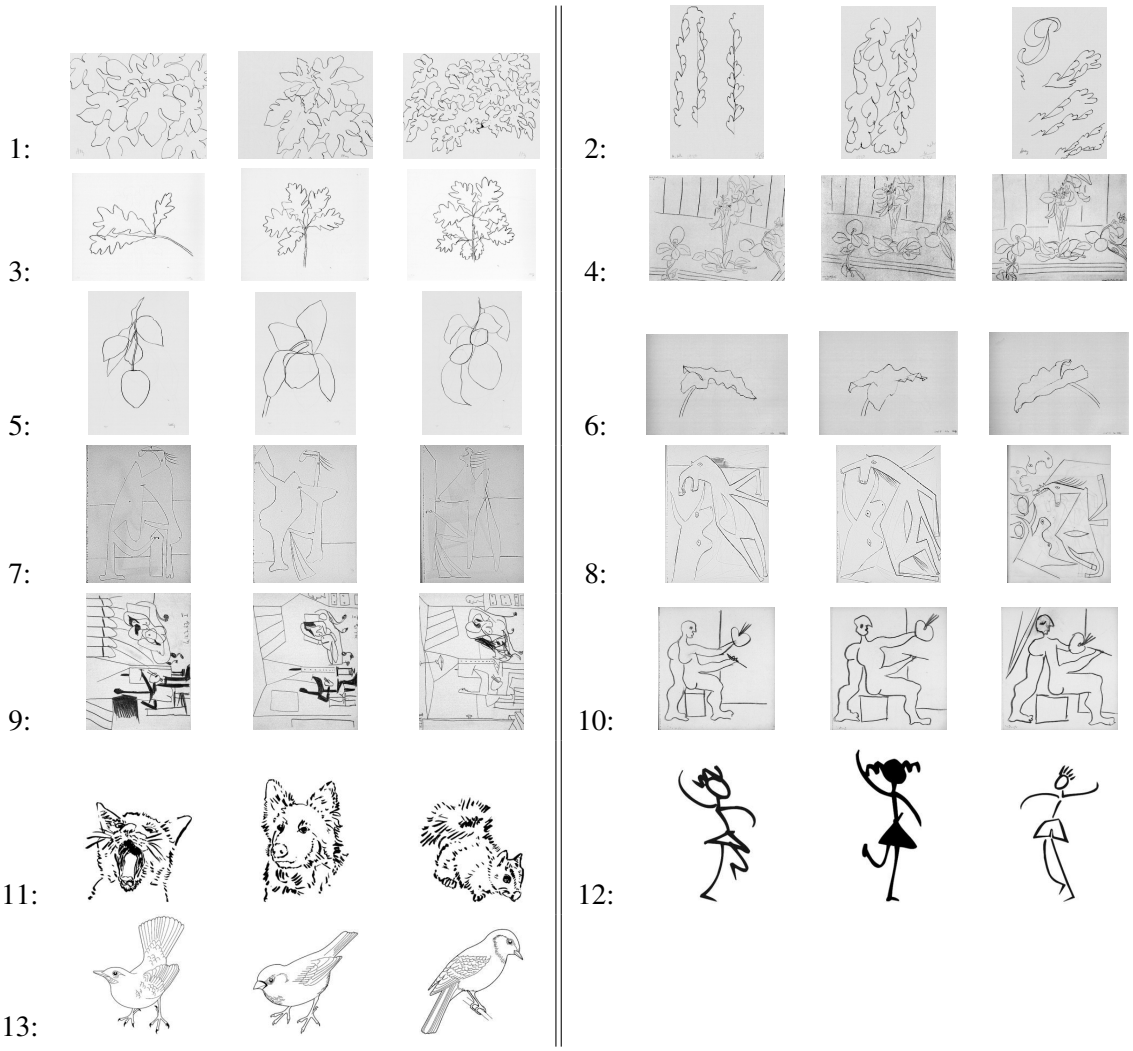


Fig. 17. Samples of the fourteen categories of the classified database that we use in Section IV-G.

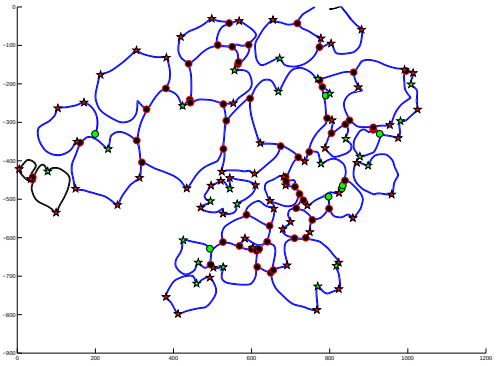


Fig. 18. One example of analyzed drawing. Green dots indicate stroke endpoints, black dots indicate stroke junctions, and red (resp. green) stars indicate negative (resp. positive) stroke corners. Coalescence sets are given different colours.

G. Similarity results

1) Comparison with a CSS based approach: In this section, we compare our method with a recent approach making use of the curvature scale space (CSS) to detect dominant points in an image [58]. Dominant points (DP) are also often called points of interest. According to the terminology used in this paper, stroke corners, junctions, and endpoints are all considered as dominant points. The method described in [58] first extract 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. We will henceforth call this method CSS-Canny.

We also compare our approach with an hybrid method, which is the same method as the CSS, 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 I using the κ'_T threshold, and the total linear density of all dominant points in the image. Features are normalized as explained in Section IV-F.

In order to perform a fair comparison between the notion of dominant point introduced in [58] and the characteristic points introduced in this paper, we have to reduce the dimensionality of the feature vector introduced in Section III-E. We consider a four-dimensional feature vector composed of features 1 to 3 (Table I), 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 I) individually. This method, henceforth called LL-sep, is equivalent to using $W_{11D} = [1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0]$ in Equation 4. 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 III-E, and is called $LL - all$ henceforth. For this complete method, a weight vector given by $W_{11D} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ is used in Equation 4.

The weighted Euclidean distance let us order the database by increasing distances from a given query. Following this order, let N_x be the number of result images needed to retrieve x images that belong to the same class as the query. This definition of recall rate allows to average curves corresponding to categories with different sizes. Figure 19 shows the precision rate x/N_x as a function of the recall rate, defined by x for all the database. Figure 20 shows the same curves for all classes.

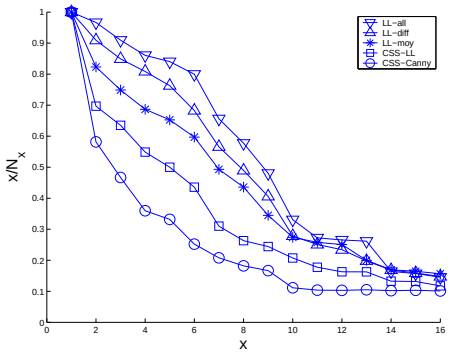


Fig. 19. For each image query, having ordered the database using the similarity measure, N_x is the number of result images needed to retrieve x images that belongs to the same class as the query. This definition of recall rate allows to average curves for queries belonging to classes with different sizes. The first retrieved image ($x = 1$) is the query.

We also use the mean discounted cumulative gain (DCG) to illustrate methods efficiency [59]. This statistical score cumulates the contributions of each retrieved image (same or different class) depending on its rank. The contribution g_r of the r^{th} retrieved image is $g_r = 1/log_2(r)$ if the image is in the same class as the query, and is $g_r = 0$ if it does not belong to the same class [60]. For one query belonging to a class of size C in a database of size N, the DCG is given by:

$$DCG = \frac{1 + \sum_{r=2}^{N} g_r}{1 + \sum_{j=2}^{C} 1/log_2(j)}, \tag{5}$$

where $DCG = 100\%$ means perfect retrieval.

TABLE II
MEAN DCG AND BEP OF THE DIFFERENT COMPARISON METHODS

	CSS-Canny	**CSS-LL**	**LL-mix**	**LL-sep**	**LL-all**
DCG	65.9%	75.6%	83.8%	89.3%	93.8%
BEP	50.9%	64.2%	71.9%	84.0%	88.2%

Finally, we also computed the Bull-Eye Percentage (BEP) which is the macro average over the database of the precision rate of correctly retrieved images over the $2C$ closest images, where C is the size of the class to which the query belongs [59]. Mean DCGs and BEP are resumed in Table II.

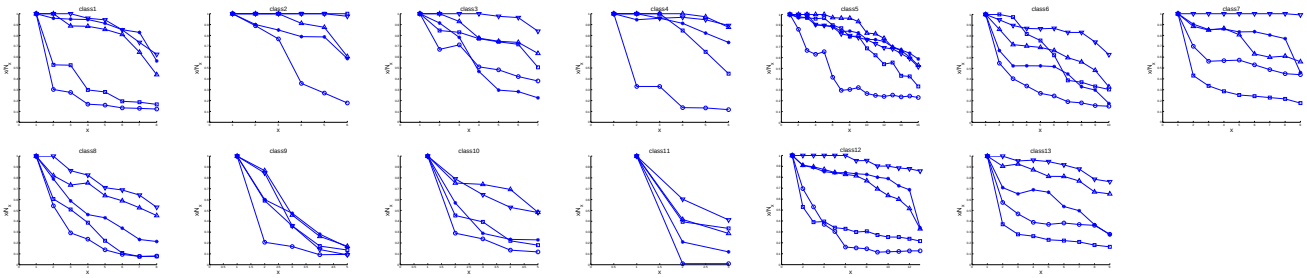


Fig. 20. Precision-recall curves for each class of Table 17. Same legend as in Figure 19.

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 problem. The comparison between CSS-LL and LL-mix methods also demonstrates that the proposed method is better able to detect the pictorial points of interests 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, one 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.

2) Examples of similarity retrieval: A set of retrieval results on the classified database are shown in Figure 21.The complete database and all results are also visible online on a permanent website[1]. In Figure 22, two examples of classification error using the LL-all method are shown. On these figures, the query image is on the top left corner, followed in scanline order by the closest results according to the weighted Euclidean distance. The first example starts from a query from the seventh class. Although all results do not belong to the correct class, matching these images with the query is not shocking. This points out the difficulty to classify a database according to the pictorial effect. In Figure 22b, the semantic gap is so wide and clear with the two last retrieved images that matching these images is indeed quite disrupting.

Finally, we mixed the 105 classified database with 154 unclassified drawings issued from the same sources (Matisse-Kelly exhibition catalogue, and Picasso sketch books). To illustrate the ability of the proposed method to describe the pictorial effect on some serendipitous matching, we show four examples on this bigger database in Figure 23.

V. CONCLUSION AND FUTURE WORK

In this paper we 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 paper 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 and we build an intuitive features vector based on the literature in art analysis. Thirdly, we illustrated features efficiency using a query by example framework on a database of line-drawings.

[1]URL: http://www.hurtut.net/geometry/

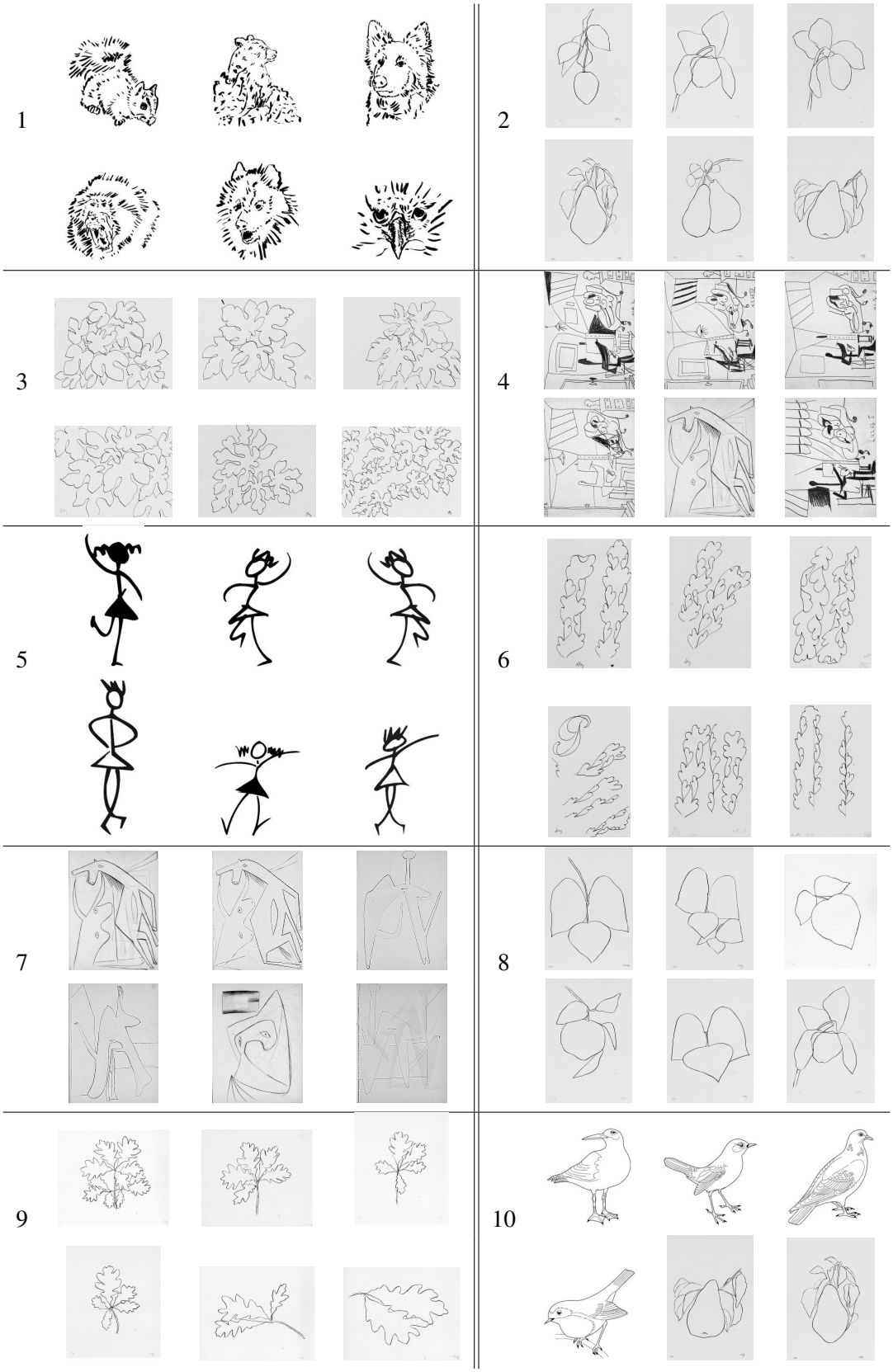


Fig. 21. Similarity retrieval results in a database of 105 line-drawings. For each example, image query is in the upper-left corner followed by its five closest images.

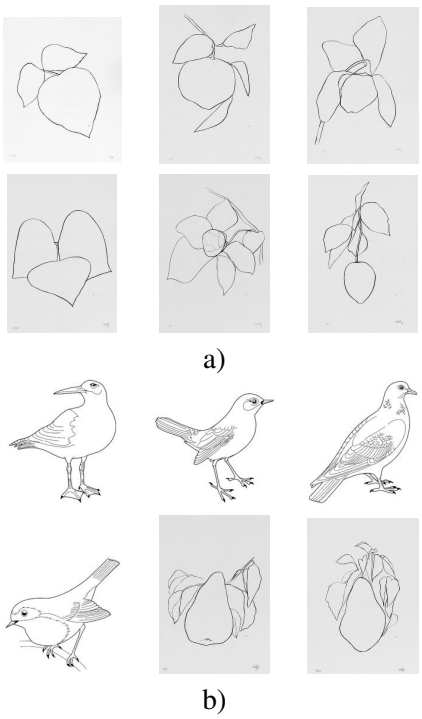


Fig. 22. a) Example of error from class 7. The query is in the top left corner, followed by the closest results. The 2^{nd} and 3^{rd} results do not belong to the 7^{th} class. Yet, matching these images with the query is not shocking. b) Example of error from class 12. Same Layout as in Figure 22. The semantic gap is so wide and clear on this example that matching the last two retrieved images with the query is indeed disrupting.

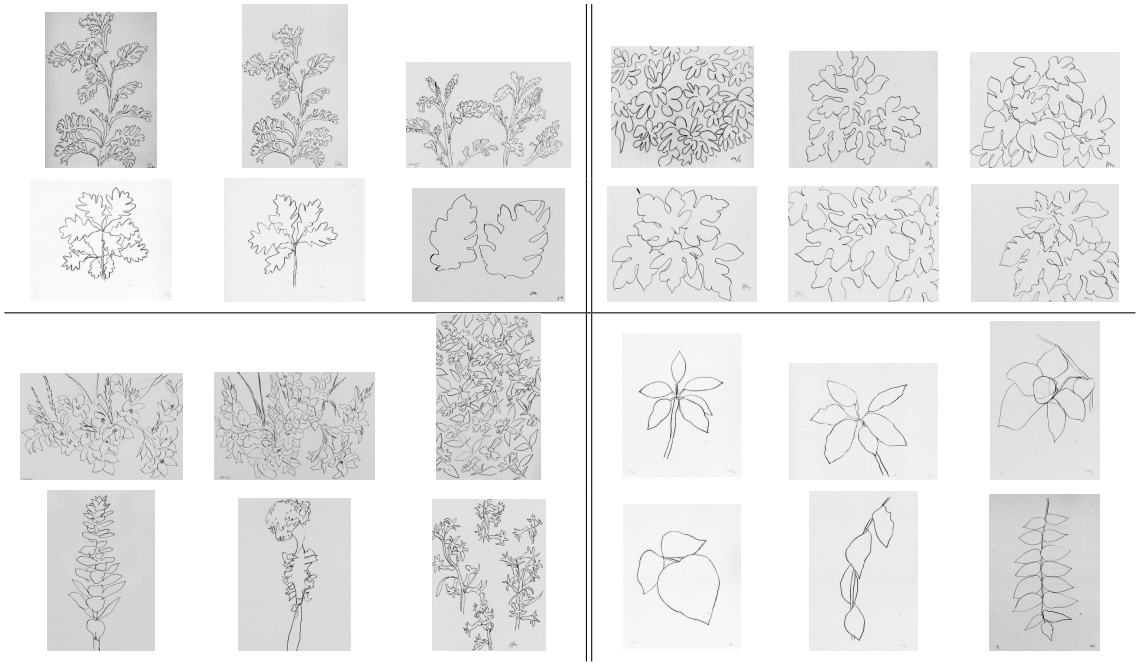


Fig. 23. Query example on a database of 259 drawings.

Computer Graphics and Non-Photorealistic Rendering (NPR) explicitly claim for methods that are able to analyze the artistic content for the sake of style analysis, style transfer rendering and aesthetic effect reproduction. We believe that the proposed methods provide a first step toward this objective concerning line art rendering. Besides, cultural heritage community also claim for suited methods that allow to retrieve drawings based on pictorial content criteria. This paper proposes such a framework for line-drawings.

This work opens several perspectives. First, we are currently working on more sophisticated similarity measures to compare line-drawings. The distance used is this paper is rather simple and in all experiments we use a uniform weighting. Future works include some kernel optimization and the possibility to describe separately every stroke cluster in a line artwork. We are also investigating the comparison of two sets of stroke clusters through a global optimization procedure relying on the Earth Mover Distance [61]. Second, the study presented in this paper is limited to line-drawings, that is drawings fully based on 1D content. The goal behind this limitation is to demonstrate the importance of 1D content independently of other types of primitive. Following the Willats framework (0D, 1D, 2D content), it would be interesting to further investigate some methods to extract 1D content of any type of artwork.

REFERENCES

[1] "Arts graphiques dpt. louvre." http://arts-graphiques.louvre.fr/.

[2] P. Owen, "Painting," *Encyclopaedia Britannica*, 2007.

[3] J. Eakins, P. Briggs, and B. Burford, "Image retrieval interfaces: A user perspective," *Proc. Third Int. Conf. on Image and Video Retrieval, Springer-Verlag Heidelberg, Dublin, Ireland*, pp. 628–637, 2004.

[4] C. Chen, H. Wactlar, J. Wang, and K. Kiernan, "Digital imagery for significant cultural and historical materials," *International Journal on Digital Libraries*, vol. 5, no. 4, pp. 275–286, 2005.

[5] P. Stanchev, D. Green Jr, and B. Dimitrov, "Some Issues in the Art Image Database Systems," *Journal of Digital Information Management*, vol. 4, no. 4, p. 227, 2006.

[6] B. Julesz, "Texton gradients: The texton theory revisited," *Biological Cybernetics*, vol. 54, no. 4, pp. 245–251, 1986.

[7] P.-M. Jodoin, E. Epstein, M. Granger-Piché, and V. Ostromoukhov, "Hatching by example: a statistical approach.," in *NPAR*, pp. 29–36, 2002.

[8] S. Bae and F. Durand, "Statistical analysis and transfer of pictorial styles," *MIT Workshop, Oxygene*, Sept. 2004.

[9] S. Grabli, E. Turquin, F. Durand, and F. X. Sillion, "Programmable style for NPR line drawing," *Eurographic Symposium on Rendering*, 2004.

[10] H. J. van den Herik and E. O. Postma, "Discovering the visual signature of painters," *Future Directions for Intelligent Systems and Information Sciences, Physica-Verlag*, pp. 129–147, 2000.

[11] J. Li and Z. Wang, "Studying digital imagery of ancient paintings by mixtures of stochastic models," *IEEE Trans. on Image Processing*, vol. 13, pp. 340–353, March 2004.

[12] Y. Yan and J. Jin, "Indexing and Retrieving Oil Paintings Using Style Information," *Lecture Notes In Computer Science*, vol. 3736, p. 143, 2006.

[13] J. Pickett *et al.*, *The American Heritage Dictionary of the English Language*. Houghton Mifflin, 2000.

[14] M. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 2, no. 1, pp. 1–19, 2006.

[15] W. Kandinsky, *Point and Line to Plane*. Dover Publications, 1979.

[16] P. Klee, *Paul Klee, cours du Bauhaus*. Hazan, 2004.

[17] H. G. Clouzot, "The mystery of picasso." DVD, 1954.

[18] P. Kammerer, E. Zolda, and R. Sablatnig, "Computer aided analysis of underdrawings in infrared reflectograms," in *Proc. of 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage* (D. Arnold, A. Chalmers, and F. Niccolucci, eds.), (Brighton, United Kingdom), pp. 19–27, 2003.

[19] M. Farine, "Klimt, dessins erotiques: Les dessins de nus de Klimt offrent des corps voluptueux et temoignent d'une parfaite maitrise de l'art graphique," *Oeil-Lausanne Then Paris-*, vol. 567, p. 64, 2005.

[20] E. d. C. Remi Labrusse, *Henri Matisse and Ellsworth Kelly - Dessins de plantes*. Gallimard, Centre Pompidou, 2002.

[21] A. Desolneux, L. Moisan, and J. Morel, "Edge detection by helmohltz principle," *International Journal of Computer Vision*, vol. 14, pp. 271–284, 2001.

[22] T. Hurtut, Y. Gousseau, F. Cheriet, and F. Schmitt, "Pictorial analysis of line-drawings," in *International Symposium on Computational Aesthetics in Graphics, Visualization and Imaging*, 2008.

[23] I. Berezhnoy, E. Postma, and H. van den Herik, "Authentic: Computerized Brushstroke Analysis," *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pp. 1586–1588, 2005.

[24] I. Berezhnoy, E. Postma, and J. van den Herik, "Automatic extraction of brushstroke orientation from paintings," *to appear in Machine Vision and Application Journal*, 2008.

[25] C. Richard Johnson, E. Hendriks, I. Berezhnoy, E. Brevdo, S. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang, "Image processing for artist identification - computerized analysis of vincent van gogh's painting brushstrokes," *to appear in IEEE Signal Processing Magazine*, 2008.

[26] N. Onkarappa and D. Guru, "Modified 9DLT Matrix for Similarity Retrieval of Line-Drawing Images," *Proceedings PReMI*, pp. 136–143, 2007.

[27] K. Tombre and A. Chhabra, *Graphics Recognition: Algorithms and Systems: Second International Workshop, GREC'97, Nancy, France, August 1997: Selected Papers*. Springer, 1998.

[28] K. Tombre, "Graphics Recognition: The Last Ten Years and the Next Ten Years," *Lecture Notes In Computer Science*, vol. 3926, p. 422, 2006.

[29] L. Schomaker, *Simulation and recognition of handwriting movements: a vertical approach to modeling human motor behavior*. Nijmeegs Instituut voor Cognitie-onderzoek en Informatietechnologie, 1991.

[30] P. Van Sommers, *Drawing and Cognition: Descriptive and Experimental Studies of Graphic Production Processes*. Cambridge University Press, 1984.

[31] L. Lam, S. Lee, and C. Suen, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[32] T. Pavlidis, "Recognition of printed text under realistic conditions," *Pattern Recognition Letters*, vol. 14, no. 4, pp. 317–326, 1993.

[33] R. Deriche, "Using canny's criteria to derive recursively implemented optimal edge detector," *Int. J. Computer Vision*, vol. 1, pp. 167–187, 1987.

[34] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[35] V. Caselles, B. Coll, and J.-M. Morel, "Topographic maps and local contrast changes in natural images," *International Journal of Computer Vision*, vol. 33, pp. 5–27, 1999.

[36] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 860–872, 2000.

[37] L. Moisan, "Affine plane curve evolution: a fully consistent scheme," *IEEE Trans. Image Processing*, vol. 7, pp. 411–420, March 1998.

[38] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel, "An *a contrario* decision method for shape element recognition," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 295–315, 2006.

[39] F. Cao, J. Lisani, J.-M. Morel, P. Musé, and F. Sur, *A theory of shape identification*. Lecture Notes in Mahematics, Springer, 2008.

[40] V. Caselles, B. Coll, and J.-M. Morel, "Topographic maps and local contrast changes in natural images," *Int. J. Comp. Vision*, vol. 33, no. 1, pp. 5–27, 1999.

[41] F. Cao, P. Musé, and F. Sur, "Extracting Meaningful Curves from Images," *Journal of Mathematical Imaging and Vision*, vol. 22, no. 2, pp. 159–181, 2005.

[42] F. Cao, *Geometric Curve Evolution and Image Processing*. Springer, Lecture Notes in Mathematics, 2003.

[43] L. Alvarez, F. Guichard, P. Lions, and J. Morel, "Axioms and fundamental equations of image processing," *Archive for Rational Mechanics and Analysis*, vol. 123, no. 3, pp. 199–257, 1993.

[44] F. Mokhtarian, "Silhouette-based isolated object recognition through curvature scale space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 539–544, 1995.

[45] D. Lowe, *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers Norwell, MA, USA, 1985.

[46] A. Ghosh and N. Petkov, "Effect of high curvature point deletion on the performance of two contour based shape recognition algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 6, pp. 913–924, 2006.

[47] W. Wu, "An adaptive method for detecting dominant points," *Pattern Recognition*, vol. 36, no. 10, pp. 2231–2237, 2003.

[48] Y. H. Gu and T. Tjahjadi, "Coarse-to-fine planar object identification using invariant curve features and B-spline modeling," *Pattern Recognition*, vol. 33, no. 9, pp. 1411–1422, 2000.

[49] Á. C. Poyato, N. L. F. García, R. M. Carnicer, and F. J. Madrid-Cuevas, "A method for dominant points detection and matching 2d object identification.," in *International Conference on Image Analysis and Recognition (ICIAR)*, pp. 424–431, 2004.

[50] M. Marji and P. Siy, "A new algorithm for dominant points detection and polygonization of digital curves," *Pattern Recognition*, vol. 36, no. 10, pp. 2239–2251, 2003.

[51] J. Fayolle, L. Riou, and C. Ducotter, "Robustness of a multiscale scheme of feature points detection," *Pattern Recognition*, vol. 33, no. 9, pp. 1437–1453, 2000.

[52] J. Willats, *Art and representation*. Princeton University Press, 1997.

[53] M. Leyton, *The structure of paintings*. SpringerWienNewYork, 2006.

[54] J. Winter, S. Panis, and J. Wagemans, "Perceptual saliency of points along the contour of everyday objects: A large-scale study," *Journal of Vision*, vol. 2, no. 7, p. 487, 2002.

[55] C. Fantoni, M. Bertamini, and W. Gerbino, "Contour curvature polarity and surface interpolation," *Vision Research*, vol. 45, no. 1047-1062, p. 7, 2005.

[56] J. Feldman and M. Singh, "Information along contours and object boundaries," *Psychological Review*, vol. 112, no. 1, pp. 243–252, 2005.

[57] R. des Musées Nationaux (RMN), "Les carnets de picasso." DVD-Rom, 2006.

[58] X. He and N. Yung, "Curvature scale space corner detector with adaptive threshold and dynamic region of support," *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, 2004.

[59] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," *Shape Modeling Applications, 2004. Proceedings*, pp. 167–178, 2004.

[60] K. Järvelin and J. Kekäläinen, "IR evaluation methods for retrieving highly relevant documents," *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 41–48, 2000.

[61] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

Dépôt légal : 2008 – 2ème trimestre
Imprimé à l'Ecole Nationale Supérieure des Télécommunications – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

TELECOM ParisTech

Institut TELECOM - membre de ParisTech

46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.telecom-paristech.frfr

Département TSI