Pictorial analysis of line-drawings

Thomas Hurtut, Yann Gousseau, Farida Cheriet, Francis Schmitt

Computational Aesthetics in Graphics, Visualization, and Imaging (2008)
P. Brown, D. W. Cunningham, V. Interrante, and J. McCormack (Editors)

Pictorial analysis of line-drawings

T. Hurtut[1,2], Y. Gousseau[1], F. Cheriet[2], and F. Schmitt[1]

[1]Telecom ParisTech, LTCI CNRS, Paris, France
[2]LIV4D, Ecole Polytechnique de Montréal, Canada

Abstract

We present in this paper an approach to the analysis of the pictorial content of artistic line-drawings. The pictorial content is the combination of the stylistic content and of the visual features of the represented subject. This paper focuses on the pictorial content hold by line strokes in line-drawings. To this aim, we propose a parameter-free method to detect the hierarchical set of stroke contours. This structure allows to estimate the radius of the drawing tool that has been used. This information then efficiently tunes several methods to extract strokes curvature information, endpoints, stroke junctions and corners. The efficiency of the proposed methods is illustrated with several experiments.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Vision and Scene Understanding]: Perceptual reasoning, shape

1. Introduction

In this paper, we call *pictorial content* the combination of the artistic style and of the visual features of the represented subject contained in an artwork. This content contributes to the visual impression delivered to a human observer. "In an abstract painting, ideas, emotions, and visual sensations are communicated solely through lines, shapes, colors, and textures that have no representational significance" [Owe07]. The psychological effect delivered by the pictorial content, or *pictorial effect*, is one part of the so-called aesthetic effect [Sil84]. However, since computational methods do not succeed in describing the representational content, the aesthetic effect cannot be fully extracted. The semantic gap [LSDJ06] is still a topical research problem in computer vision. This gap is even wider in the visual arts field since pictures are often not perfectly realistic. We therefore limit our study to the pictorial level of analysis. The analysis of the pictorial content is useful in many applications for identifying and retrieving similar artworks in cultural heritage databases [SGJD06]. It is also useful for non-photorealistic rendering (NPR) of scences in computer graphics that produces the same *pictorial effect* as some real artworks [JEGPO02, BD04, GTDS04].

The pictorial content is partially related to the artistic style. The style detection problem has been tackled in many references, see e.g. [vdHP00, LW04, YJ06]. 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 [P*00], 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, which still cannot be fully carried out with computer vision techniques.

Line-drawings have long been considered as a very important study in artist preliminary works [Kan79, Kle04]. It is also often a standard format for illustration in print publications using black ink on white paper. Line content in artworks have been addressed in several perceptual studies. In its inspiring study [Wil97], Willats proposed several depiction systems that let one person describe the artistic rules chosen by an artist. These depiction systems are well adapted to NPR line-drawings [GTDS04]. Leyton studies the *visual tension* induced by the geometrical shapes in a painting [Ley06]. Leyton theory proposes to see shapes as memory storages. Practically, his theory is based on curvature extrema information and local neighborhood of these extrema.

This approach is closely related to cognitive studies such as the seminal work of Attneave for instance [Att54].

Studying the line content in artworks has been rarely investigated in computer vision. Berezhnoy et al. [BPvdH05] proposed a method using polynomial interpolation of painting brush-strokes on Van Gogh artworks. This approach is 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 [LW04]. Their method is yet close to texture analysis in paintings. Onkarappa and Guru studied the spatial mutual arrangement of strokes in line-drawing images to achieve similarity retrieval [OG07]. This approach aims at describing the artistic composition of an image.

In this paper, we first propose in Section 2 a parameter-free approach to detect closed contours in line-drawings. Next, we introduce in Section 3 several methods for extracting and characterizing some important visual features: curvature, corners, junctions, endpoints. In Section 4, some similarity experiments on a database of artistic line-drawings will be used to demonstrate the efficiency of the features.

2. Detection of closed stroke contours

We thereafter call *stroke cluster* any set of connected strokes. In this section, we explain how to associate closed contours to the stroke clusters present in a line-drawing. This is a crucial step since the geometrical analysis introduced in the next section is based on these closed contours. Among the numerous segmentation approaches proposed in the literature, we choose to rely on level lines and draw our inspiration from the method presented by Desolneux et al. in [DMM01]. Level lines hold several interesting properties. They are represented by closed curves in a hierarchical structure, they do not require any initialization, and the selection method proposed in [DMM01] relies on only one parameter which is fixed for every image.

Level lines are defined as the connected components of the topological boundary of level sets. There are both positive level lines (corresponding to upper level sets) and negative ones (corresponding to lower level sets). These lines are organized in an inclusion tree, called the topographic map, that can be efficiently computed on digital images, as detailed in [MG00]. Level lines provide a fairly complete but largely redundant representation of the geometry of an image. Desolneux et al. have proposed an image segmentation method relying on a filtering of the topographic. A meaningfulness measure called *number of false alarms* (NFA) is affected to each line. This measure depends on both the length of the line and its contrast. According to the NFA measure, selecting *meaningful level lines* practically yields lines that are well localized along contours but are still largely redundant. In [DMM01], it is suggested to further filter the meaningful lines by using a maximality principle. The goal of this step is to associate exactly one line to each geometrical entity present in the image. The maximality principle relies on the fact that the set of meaningful level lines inherits the hierarchical structure of the topographic map (i.e. two meaningful level lines are either disjoint or linked by inclusion). For each sequence of meaningful level lines that have each exactly one child and are of the same type (positive or negative), only the more meaningful is kept. While giving good results on natural images, this maximality principle is not well adapted to the segmentation of line-drawings. This is mainly due to contrast variations along the strokes.

We therefore propose an alternative way to filter the tree of meaningful level lines that is specifically suited to line-drawings and aims at associating a unique contour to each stroke. The ideal structure we are aiming at is explained in Figure 1 for some manually selected level lines on a line-drawing. In order to achieve such a segmentation, we take advantage of the fact that line-drawings have a constrained structure, and in particular that a stroke is made of a single shape inside which there is no contrast inversion. We call Maximal Monotone Tree (MMT) a subtree of the set of meaningful level lines than contain only positive or only negative nodes and which cannot be included in another monotone subtree. For each MMT of the tree of meaningful level lines, we first look for the most meaningful level line $\mathcal{L}$ according to the NFA measure. We then remove all level lines from the MMT which are children or parents of $\mathcal{L}$. These two steps are repeated until the MMT is empty. This principle allows us to keep only the most meaningful level lines and it insures in most situations that each contour of a stroke will be represented by a unique level line, without having to tune any parameter. The contours obtained using this MMT-principle on the tree of meaningful level lines in Figure 1a is exactly Figure 1b. More examples of stroke contours detection are to be seen in Figure 6.

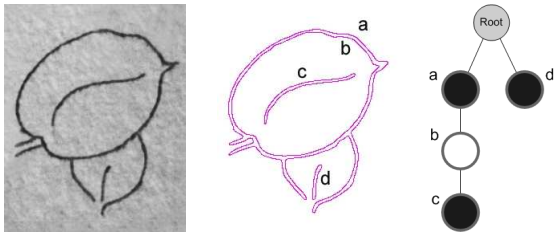


Figure 1: *A simple line drawing of two leaves (left). Four level lines are manually chosen to best represent the strokes contours (middle). On the tree structure of these four level lines (right), a positive or negative level line is represented by a white or black node respectively. The unique positive level line* b *describes the inner contour of the biggest leaf.*

Finally we group these selected meaningful level lines to model the contours of every stroke cluster. These groups of

level lines, thereafter called *coalescence sets*, are obtained by grouping every retrieved negative level line with its positive children. More details can be found in [HGCS08].

3. Pictorial content analysis

This section proposes several methods to analyze the coalescence sets introduced in the previous section. The estimation of the tool radius (Section 3.1) tunes several geometrical parameters in the following steps. The proposed description of line-drawing is based on the curvature distribution (Section 3.2) and on the detection of several visual cues such as stroke junctions, corners and endpoints (Section 3.3). Eleven scalar features are computed based on these cues (Section 3.4).

3.1. Tool radius estimation

We make the following hypothesis. Each stroke cluster Φ is done with tools having the same radius R_Φ. This may sound restrictive since for example calligraphic artworks are discarded, 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 stroke model 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:

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

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

3.2. Curvature computation

Curvature being quite sensible to noise, level lines of the coalescence sets needs to be smoothed. Besides, the hierarchical structure inherited from the topographic map must be preserved. We thus rely on affine plane curve smoothing which is morphologically invariant [Moi98]. This smoothing is applied directly to the polygonal models of level lines. A fixed sampling step at the curvilinear abscissa is used for every image: $\Delta s = 0.5$ pixels. Let us stress that only one smoothing is operated here at a very fine scale. Our approach differs from a multi-scale approach such as [Mok95].

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

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

where $\theta_{s_i s_{i+1}}$ 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.

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 local curvature sign wherever they are located in a coalescence set (Figure 2). Endpoints have always a negative curvature values, and regions where strokes create a non-reflex angle (e.g. at junctions) hold positive values.

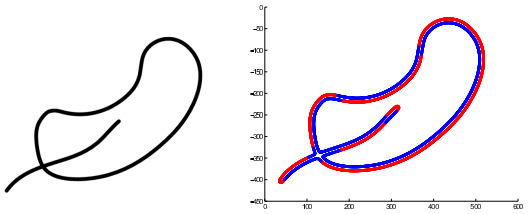
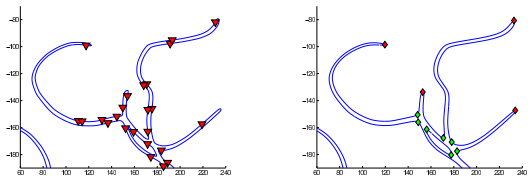


Figure 2: *During curvature computation, convention is chosen so that negative (resp. positive) level lines are travelled clockwise (resp. anticlockwise). On this example (left), one negative level line delimits the outer contour, and one positive level line delimits the inner contour (right). Arrows notify the travel direction. This convention induces negative curvature values (red dots) and positive values (blue dots) 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.*

Curvature zero-crossings contribute to artistic visual impression and object recognition [Low85]. They are estimated directly on the curvature signal. An example of a drawing with detected curvature zero-crossing is to be seen in Figure 3a.



a) b)

Figure 3: *Left, maximal meaningful level lines from a drawing, and its superposed curvature zero-crossings. Right, curvature maxima detected with the method presented in Section 3.3.1. Red (resp. green) diamonds are positive (resp. negative) maxima.*

3.3. Detection of endpoints, junctions and corners

This section proposes to detect three different strong visual cues: stroke endpoints, stroke junctions and stroke corners, following to the taxonomy presented in Figure 4. In Section 3.3.1, we first select a set of candidates to that taxonomy.

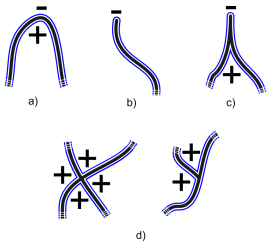


Figure 4: *Taxonomy of high curvature values along stroke contours: a) a simple high curvature point along the gesture trajectory, b) an endpoint, c) a stroke corner, d) two types of stroke junctions: X and T junctions. Plus and minus sign refers to the curvature sign on the stroke contour. Strokes are grey lines, and stroke contours are blue surrounding lines.*

3.3.1. 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 4. 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 4.1. Zero-crossings of the derivative signal of each of these portion are considered as candidates to the taxonomy. Depending on the curvature sign, an extremum can be either positive or negative. An example of extrema detection is shown in Figure 3b. Thanks to the chosen convention on curvature presented in Section 3.2, stroke endpoints coincide with negative maxima (Figure 4b), and junctions coincide with several positive maxima (Figure 4d). Corners have one positive maximum on the concave side, and one negative maximum on the convex side (Figure 4c). This information will be useful in the next three sections to analyze stroke endpoints, junctions and corners.

3.3.2. 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 [Wil97]. A positive curvature maximum indicate possible junction nearby its location point p_m in the drawing. Recall that R_Φ is the tool radius. To state for a junction we center a disk of radius $k_j R_\Phi$ on p_m. 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 Section 3.3.3. 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 $k_j R_\Phi$.

3.3.3. Stroke corners

A positive maximum that has not been classified as stroke junctions is considered as a stroke corner. Such maxima points have a very strong pictorial impact [Ley06]. 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 [WPW02]. 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 belonging to the coalescence set than the points $\{p_{m-j}, \dots, p_{m+i}\}$. 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 form that is perceptually visible such as an object contour, a line corner can be convex or concave and induces a different pictorial effect. This type effect has been recently studied in [FBG05, FS05]. 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 at stake, 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 with 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 $i_\Phi \Delta s \approx R_\Phi$. This works in most of the situations. However 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.

3.3.4. 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 $k_e R_\Phi$ distance from the maximum. 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 $q < i$ and $q > j$, then p_m is considered as an endpoint. This distance threshold tolerates a 50% precision error on R_Φ. If the maximum is an endpoint, the distance between p_i and p_j should be indeed close to the width $2R_\Phi$. This method is illustrated on Figure 5.

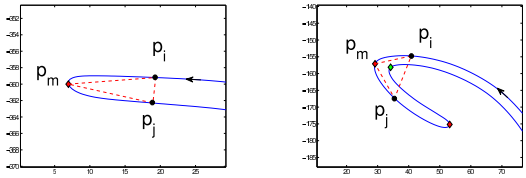


Figure 5: *Left: negative maximum is classified as a stroke endpoint. Right: some points p_q which are not belonging to the level line portion $[p_i, p_j]$, are included in the triangle. This maximum is thus not classified as a stroke endpoint.*

3.4. Indexing features

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 1. The three first features are computed on the distribution of curvature values $\kappa(i)$ satisfying $-\kappa_t \leq \kappa(i) \leq 0$. Negative curvature values let us select one half of the curvature points along the strokes (see Figure 2). We thus indirectly consider points that are related to the underlying drawing stroke. We only consider values that respect $\kappa(i) \geq -\kappa_t$ to discard the strong curvature values that belong to stroke endpoints. These three features 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 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 its total number in the image divided by the total length of the level lines of the coalescence sets in the image. 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 to 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 [Jul86]. We conducted several experiments on Julesz textons using the methods proposed in this paper. They led to the same perceptive results as in Julesz studies.

Table 1: *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 positive stroke corners
8. linear density of negative stroke corners
9. sum of the depths of positive stroke corners
10. sum of the depths of negative stroke corners
11. tool radius normalized by the image diagonal

4. Experiments and results

Section 4.1 studies the setting of the three parameters that have been used to extract the characteristic points: k_c, k_e and k_j. Some similarity retrieval results on a database of artistic line-drawings are discussed in Section 4.3.

4.1. Setting of parameter k_c, k_e and k_j

Ideally, κ_t is set in such a way that the three first features of Table 1 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 tradeoff has been empirically set with $k_c = 5$, using synthetic images made with different tool radius and real line-drawings.

Endpoints are detected using a triangle of scale $k_e R_\Phi$ around a maximum candidate (Section 3.3.4). 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 endpoints. Stroke junctions are detected using a disk region 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 $k_j = k_e = 5$. 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 neighborhood. In the remaining part of this paper, we will thus use those two values.

4.2. Detection of endpoints, junctions and corners

Three examples of analyzed drawings are to be seen in Figure 6. Detected stroke endpoints, junctions and corners are represented. First example shows that corner signs quite often correspond with corner types (Section 3.3.3). Corners strength (i.e. corner relative surface) is not shown for better visibility. Yet, we can see that this feature may have a

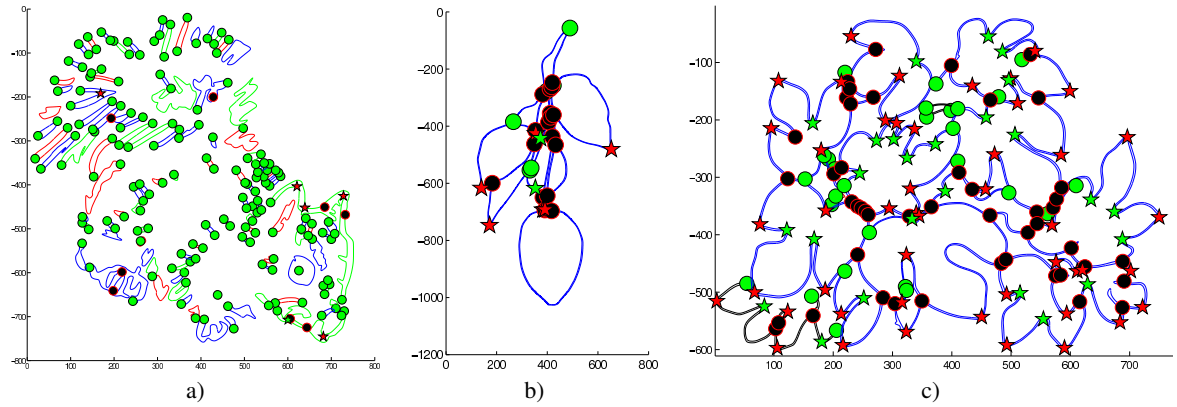


Figure 6: *Examples of analyzed line-drawings. Green dots are detected stroke endpoints, black dots are detected stroke junctions, and green (resp. red) stars are positive (resp. negative) stroke corners. Coalescence sets have different colors. Left, middle and right images are the queries for the first, second and third retrieval example respectively in Figure 7.*

discriminative impact between these three examples. Left example is made of many short strokes. It is thus strongly characterized by a particularly high linear density of endpoints.

4.3. Similarity results on a line-drawings database

In this section, we rely on a database made of 105 line-drawings. Most of this database comes from an exhibition catalogue of drawings made by Henri Matisse and Ellsworth Kelly [RL02]. Images have been scanned with an HP Scanjet 8200 with 4800 ppi resolution. 21 images come from several Picasso sketch books released on DVD-Rom in 2006 as a 800×600 digital facsimile edition [dMNR06]. Finally, 31 computer drawings from [Exp02] have also been included. Computational times for the complete features extraction procedure takes 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. Thanks to the Euclidean distance and the size of the feature vector, a query runs almost instantaneously.

Each drawing is described by its eleven scalar features (Table 1). Once each feature is computed for every drawing in a database, it is centered by the mean and normalized by the standard deviation of the feature distribution over the database. Let $V^i(k), j \in [1,11]$ be the eleven normalized features of a drawing, then the similarity measure $d(V_1,V_2)$ between two drawings is the following Euclidean distance: $d(V_1,V_2) = (\sum_{k=1}^{11}(V_1^i(k) - V_2^i(k))^2)^{1/2}$.

Retrieval results on this database are shown in Figure 7. On these figures, the query image is on the top left corner, followed by the closest results according to the Euclidean distance. First example is based on the query image of Figure 6a. All results hold a high linear density of endpoints. This example gives a good insight of what we call pictorial effect, results being semantically different, yet delivering a very similar pictorial expression. Same remark can be done for examples 2, 4, 7 and 8. Examples 3, 6, 5 and 9 are very similar both from the pictorial and semantical point of view. The visual features of the represented subject are indeed very similar and discriminative compared to the rest of the database content. The two last results of the 10^{th} example for instance are yet so different from the query and the three closest images that they look unsatisfying, knowing that more similar rendered birds are in the database. This points out a limit of the proposed description: in some cases, the semantic impact delivered by the represented subject might overwhelm the pictorial effect. In the image retrieval context, this limit could be moved away by integrating some subject metadata when available, if someone seeks images having the same semantical content and being rendered with the same style as the query. The whole database and all retrieval results are to be seen online on a website[†]. This database has also been classified by a professional artist into 13 classes. Discounted cumulative gain (DCG) and Bull-eye percentage (BEP) [SMKF04] are 93.8% and 88.2% respectively using these classes.

5. Conclusion and future work

In this paper we introduced a method for the pictorial content analysis of line drawings using the geometrical information of stroke contours. Computer Graphics and Non-Photorealistic Rendering (NPR) could benefit from methods that are able to analyze the artistic content. This would enable style transferring for the rendering of drawings or line textures. We believe that the proposed methods are a promising step toward this objective. This work opens several perspectives. First we are presently collaborating with several

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

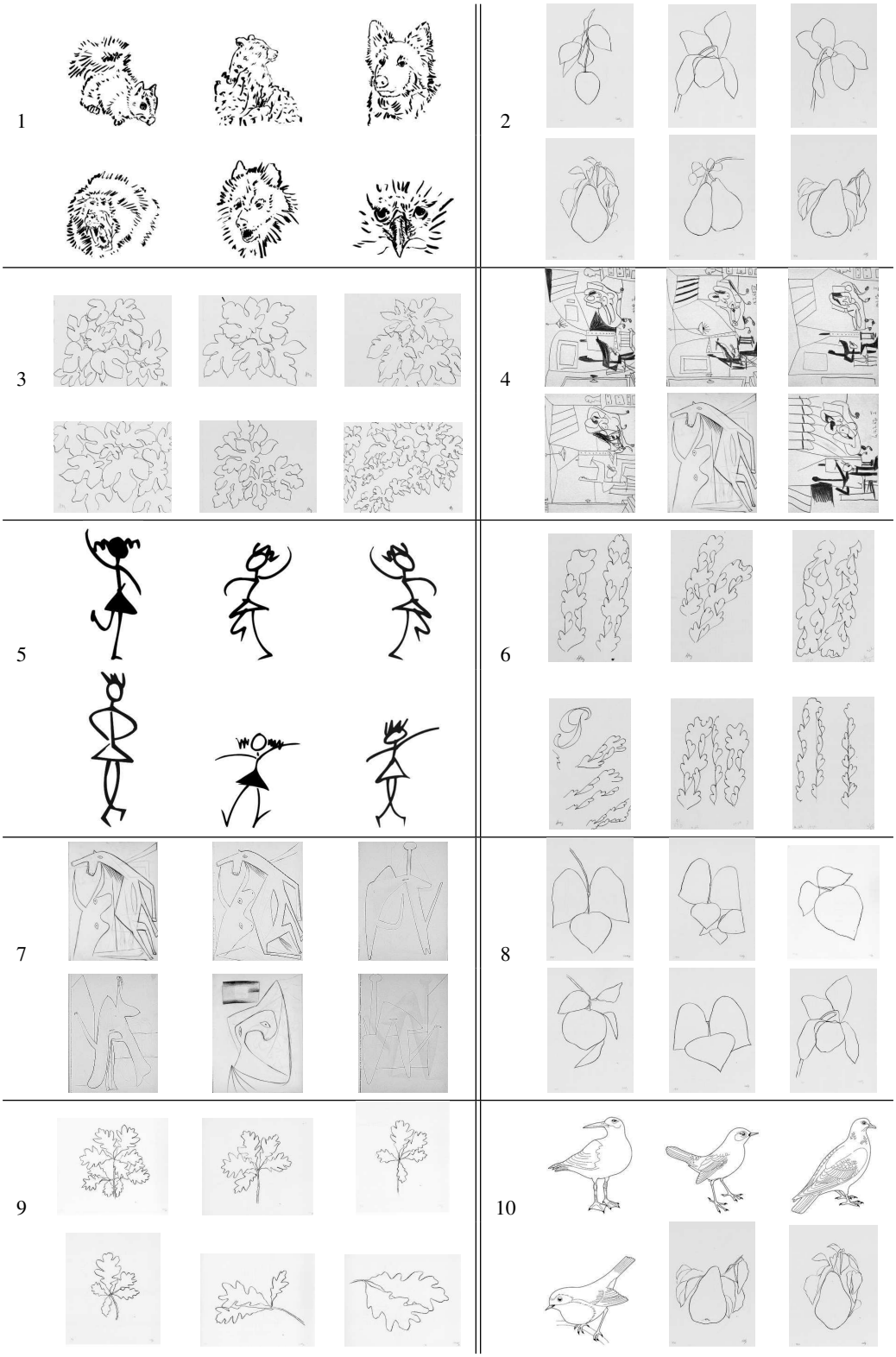


Figure 7: *Similarity retrieval results in* 105 *line-drawings database. For each example, image query is in the upper-left corner followed by its five closest images.*

artists to evaluate the efficiency of the proposed features. These artists will be asked to classify a database on which we will rely to quantify the performance of our approach. Another perspective concerns more sophisticated similarity measures using some kernel optimization for non-uniform weighting. We are also investigating the possibility to describe separately every coalescence set in a line-drawing and to model the comparison of two groups of coalescence sets through a global optimization procedure, e.g. by relying on the Earth Mover Distance [RTG00]. This paper is limited to line-drawings, that is drawings fully based on 1D content. Our goal was to observe the importance of 1D content independently of other types of primitive. It would be interesting to further investigate some methods to extract 1D content in any type of artwork as for example in paintings.

References

[Att54] ATTNEAVE F.: Some informational aspects of visual perception. *Psychol. Rev 61*, 3 (1954), 183–93.

[BD04] BAE S., DURAND F.: Statistical analysis and transfer of pictorial styles. *MIT Workshop, Oxygene* (Sept. 2004).

[BPvdH05] BEREZHNOY I., POSTMA E., VAN DEN HERIK H.: Authentic: Computerized Brushstroke Analysis. *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* (2005), 1586–1588.

[DMM01] DESOLNEUX A., MOISAN L., MOREL J.: Edge detection by helmohltz principle. *International Journal of Computer Vision 14* (2001), 271–284.

[dMNR06] DES MUSÉES NATIONAUX (RMN) R.: Les carnets de picasso. DVD-Rom, 2006.

[Exp02] EXPLOSION A.: Hallogram. http://www.hallogram.com/artexplosion/, 2002.

[FBG05] FANTONI C., BERTAMINI M., GERBINO W.: Contour curvature polarity and surface interpolation. *Vision Research 45*, 1047-1062 (2005), 7.

[FS05] FELDMAN J., SINGH M.: Information along contours and object boundaries. *Psychological Review 112*, 1 (2005), 243–252.

[GTDS04] GRABLI S., TURQUIN E., DURAND F., SILLION F. X.: Programmable style for NPR line drawing. *Eurographic Symposium on Rendering* (2004).

[HGCS08] HURTUT T., GOUSSEAU Y., CHERIET F., SCHMITT F.: *Pictorial content analysis of line-drawings using geometrical shape information*. Tech. Rep. 2008 E 001, TELECOMParisTech, 2008.

[JEGPO02] JODOIN P.-M., EPSTEIN E., GRANGER-PICHÉ M., OSTROMOUKHOV V.: Hatching by example: a statistical approach. In *NPAR* (2002), pp. 29–36.

[Jul86] JULESZ B.: Texton gradients: The texton theory revisited. *Biological Cybernetics 54*, 4 (1986), 245–251.

[Kan79] KANDINSKY W.: *Point and Line to Plane*. Dover Publications, 1979.

[Kle04] KLEE P.: *Paul Klee, cours du Bauhaus*. Hazan, 2004.

[Ley06] LEYTON M.: *The structure of paintings*. SpringerWienNewYork, 2006.

[Low85] LOWE D.: *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers Norwell, MA, USA, 1985.

[LSDJ06] LEW M., SEBE N., DJERABA C., JAIN R.: Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 2*, 1 (2006), 1–19.

[LW04] LI J., WANG Z.: Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Trans. on Image Processing 13* (March 2004), 340–353.

[MG00] MONASSE P., GUICHARD F.: Fast computation of a contrast-invariant image representation. *Image Processing, IEEE Transactions on 9*, 5 (2000), 860–872.

[Moi98] MOISAN L.: Affine plane curve evolution: a fully consistent scheme. *IEEE Trans. Image Processing 7*, 3 (March 1998), 411–420.

[Mok95] MOKHTARIAN F.: Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17*, 5 (1995), 539–544.

[OG07] ONKARAPPA N., GURU D.: Modified 9DLT Matrix for Similarity Retrieval of Line-Drawing Images. *Proceedings PReMI* (2007), 136–143.

[Owe07] OWEN P.: Painting. *Encyclopaedia Britannica* (2007).

[P*00] PICKETT J., ET AL.: *The American Heritage Dictionary of the English Language*. Houghton Mifflin, 2000.

[RL02] REMI LABRUSSE E. D. C.: *Henri Matisse and Ellsworth Kelly - Dessins de plantes*. Gallimard, Centre Pompidou, 2002.

[RTG00] RUBNER Y., TOMASI C., GUIBAS L. J.: The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision 40*, 2 (2000), 99–121.

[SGJD06] STANCHEV P., GREEN JR D., DIMITROV B.: Some Issues in the Art Image Database Systems. *Journal of Digital Information Management 4*, 4 (2006), 227.

[Sil84] SILVERMAN R.: *Learning about Art: A Practical Approach*. Romar Arts, 1984.

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton Shape Benchmark. *Shape Modeling Applications, 2004. Proceedings* (2004), 167–178.

[vdHP00] VAN DEN HERIK H. J., POSTMA E. O.: Discovering the visual signature of painters. *Future Directions for Intelligent Systems and Information Sciences, Physica-Verlag* (2000), 129–147.

[Wil97] WILLATS J.: *Art and representation*. Princeton University Press, 1997.

[WPW02] WINTER J., PANIS S., WAGEMANS J.: Perceptual saliency of points along the contour of everyday objects: A large-scale study. *Journal of Vision 2*, 7 (2002), 487.

[YJ06] YAN Y., JIN J.: Indexing and Retrieving Oil Paintings Using Style Information. *Lecture Notes In Computer Science 3736* (2006), 143.