Line Segmentation and Analysis with Special Interest to The Duct of A Line*

Carola Schönlieb
Departement of Mathematics
Hellbrunnerstr. 34
A-5020 Salzburg
Austria
carola.schoenlieb@sbg.ac.at

Kung Chieh Wang
Departement of Mathematics
Hellbrunnerstr. 34
A-5020 Salzburg
Austria
kung.wang@sbg.ac.at

Abstract

The duct is the personal hand writing of an artist, i.e. the application of color, the way of brush or line drawing and the position of the color application. Most art scientists say that the duct of a painter is very (if not most) important for the analysis of his paintings. So, to recover the handwriting of an artist means among other things to analyse lines in his images. There exists a number of different types of lines. A line can be linearly or strongly curved, thick or thin, it can be well separated from other lines, or can correspond to a collection of lines. Also the inner structure, namely the texture of a line varies with respect to the used drawing materials.

Considering an arbitrary painting our first goal is the detection and extraction of lines in an image. This means to follow single lines over line intersections and bifurcations as long as possible and extract resulting lines.

We studied lines of paintings from different artists with various structure and texture analysis methods, which we adapted for the application to lines. We consider lines both in low and fine resolution. In a low resolution we computed, among others, the chaincode and the linelength. In a fine resolution we evaluated statistical values of the graylevel–, edge– and localextrema images.

Our goal is to get a description of the duct, which is said to be like the hand writing of a painter, to get an additional parameter for the classification of paintings with respect to the artist. In the following pages we present our method.

Keywords: arts analysis, line detection, line segmentation, line analysis, feature selection, classification.

1. Introduction

Segmentation and analysis of lines in paintings is a task which is not often discussed in image analysis. The biggest problem in this context is to find characteristics of lines that make them separable from other objects in an image. Especially in paintings, lines have to be detected among other objects, e.g. textured regions. We achieved very good results using the method of Iverson and Zucker that detects lines with logical and linear operators [IZ95].

Another problem are crossings, intersections and bifurcations of lines and how to interpret them. The solution of this problem is the correct extraction of lines. There exist a few approaches on line extraction in the field of roadmap analysis and medical research. One of these methods works with Gabor filters that extract lines in various directions, see [Gra78], [FS98] and [CST00]. The disadvantage of Gabor filters is that lines with large directional changes (for example a spiral) are decomposed in single line segments independently from the local behaviour of the line. Further the above mentioned method using the Gabor filter has high computational costs because 3 dimensional labeling is necessary. Another approach is the computation of the gradient direction at each point of the image and extraction of lines following this direction pixelwise. This method apart from its elegance is unfortunately not very robust against little noise in the image.

For our purpose of line extraction in painted images we consider the chaincode of the thinned lines in an image. Following single lines in the image as long as possible resulting lines are extracted. In each junction pixel of two lineparts a decision with the help of the chaincode in pixels close to the junction is made.

After the extraction of lines we have to find description parameters allowing us to distinguish between lines of different artists. We focus on the structure of a line on the one hand and on the texture of a line on the other.

Our sample set consists of five artists with 6 images from every artist. Finally, results allowing us to distinguish these five artists are presented.

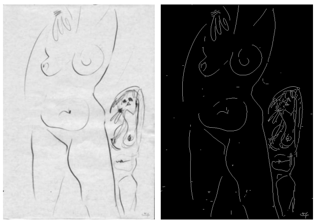


Fig. 1: A sample graylevel image bg02 and its corresponding line image

2. Line Detection and Extraction

We consider a graylevel image of our set with the goal to extract single lines in the image. Lines are continuous objects in the image resulting from the application of drawing and characterized by conditions on it´ s shape. First of all the lines in an image are detected and then separated from each other. This separation means to extract single lines over line intersections and bifurcations as long as the line doesn´ t end. The following subsections explain our line extraction algorithm.

2.1. Line Detection

Line detection is done with the method of logical–linear operators presented by Iverson and Zucker in 1995 [IZ95] and also used by [Ste98]. Image pixels are evaluated by logical linear operators. When the operator is logical–positive, then the pixel belongs to the class of image curves, which is then divided into three subclasses, namely edges, positive and negative lines. The logical linear operators consist of first to fifth derivatives of the profile of the local graylevelmap of the image.
The next step is the separation of the lines from each other. Therefore the image of detected lines is binarized and thinned until there are only objects with one pixel width, Figure 1.

2.2. Connected Components Labeling

We separate the thinned line image in its connected components with a two-dimensional labeling algorithm. To clean the line image from noise and small objects, only those connected components remain, whose sizes lie over a user defined threshold (e.g. = $\frac{1}{10}$ of the length of the diagonal of the image).

2.3. Extraction of Lines

The following computations are done for each of the retained connected components of the previous section. Each connected component should be divided into its corresponding lines. We begin at one endpoint (point with only one linepixel in a neighbourhood or for closed lines an arbitrary point) of the connected component and note this point as the first point of the line that we want to extract. Moving along the connected component our algorithm checks each point of the connected component to be part of the current line or not. The evaluation criterion whether a point should join a line is the chaincode variation, see Fig. 2 below. In addition, the continuation of a line in an intersection point also depends on the directional change following each possible line continuation. If the directional change for the new point to its previous neighbour is greater than a given threshold (e.g. = $\frac{\pi}{2}$) the line ends.

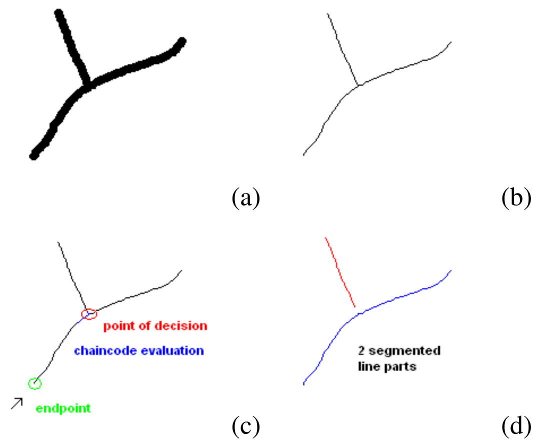
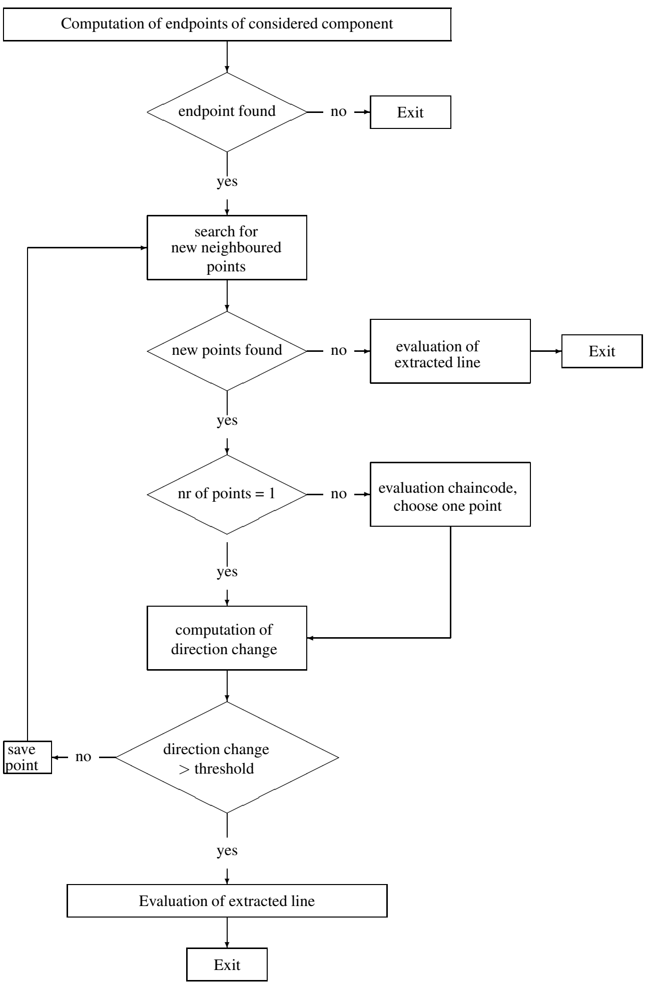


Fig. 2: f.l.t.r.(a)-(d): Extraction of line parts from a connected component: The line image (b) is the thinned version of (a). (c): Beginning at one end of the line image the chaincode is evaluated from one pixel to the next one. At the intersection of the two lines, the chaincode for the two possible continuations is computed and compared to the chaincode so far. The line is prolongated in the direction characterized by the smallest chaincode variation. (d): The result are two separate lines.

The following scheme should explain our algorithm for the extraction of lines. The algorithm is run for each connected component and each line end. After the separation of a line from the connected component it is eliminated from the connected component and saved in a separate image. Further line extraction is made for the remaining connected component until all lines are extracted and nothing is left in the connected component. Only those extracted lines are saved for further computations, whose length lies over a threshold dependent on the size of the image (e.g. = $\frac{1}{10}$ of the length of the diagonal of the image).

Line Extraction Algorithm



2.4. Inverse Thinning

The decomposition of the image into single lines is done on the thinned version of the line image. After this process, each extracted line has to be rebuilt to its original form in the image. We call this original form the thick line. The thickness of the original line in each point of the thinned line is computed (compare section 3.1.5). Taking the maximal thickness value of the line, we thicken the thinned line until it has this maximal thickness in each point of the thinned line. This results in a binary mask for the original line.

2.5. Line evaluation

For the task of line classification two properties are computed for each extracted line: variance of the thickness of the thick line along the thinned center line, and length of the thinned center line. If the variance of the thickness is small and the length large, the line is regarded as a perfect line. If the variance of the thickness is large and the length small, then the line is removed and not processed further.

2.6. Line Extraction Results

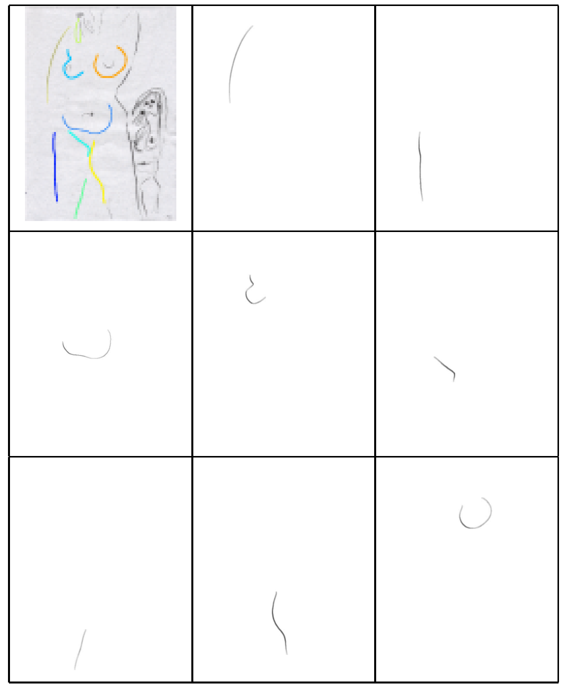


Fig. 3: Sample image of one of our artists with an assortment of extracted lines

3. Line Analysis

In order to describe a line we studied lines in both low and fine resolution. Fine resolution means an image with 600 dpi resolution, low resolution an image with half the size. We use the following computations:

1. Computations on the low resolution of the line (shape analysis):
   - chaincode
   - length of a line in comparison to the length of the direct connection of the two endpoints of the line
   - angle variations along the line
   - local curvature of a line
   - thickness of a line
2. Computations on the fine resolution of the line (texture analysis):
   - fractal dimension along the center line

- cooccurrence matrix and corresponding statistical values [Har79]
- parameter for the abruptness of maxima and minima within 3×3 windows
- edge detection within the line
- number of holes within the line

3.1. Computations on the Coarse Version of the Line:

The following computations are done on the thinned version of a line, called center line.

3.1.1 Chaincode: The chaincode of a line is a sequence of direction specifications, which ranges from 0 to 7 (8 direction possibilities from one pixel to the next one). Therefore the chaincode gives information about the direction or the curvature of a line.

3.1.2 Length of a Line: The length of a line is computed by adding 1 in all directions that are even multiples of 45 degree and $\sqrt{2}$ for all odd multiples of 45 degree. Although the computation of the length of a line is trivial (it is delivered with the chaincode), its importance is significant. For example, the comparison of the actual linelength and the distance of the two lineends tells us something about the straightness, i.e. the curvature of the line. If the difference between the two lengths is small, the line is nearly straight. In contrast, if the difference is large, the line should be highly curved.
A finer analysis is described in the next section, where the angle variations along the line are considered.

3.1.3 Angle Variations Along the Line: Approximating the line with a polygon with a fixed distance of the polygon points along the line, we are able to measure angle variations along the line. For every pair of neighboured polygon segments, we calculate their angle using the first polygon segment as base line. Variating the fixed distance between two polygon points a number of polygons results. For each approximating polygon we get a list of angles. Computing the variance of these angles for each polygon we get a list of values representing the angle variations along the line. First order statistical parameters of the angle variations are evaluated.

3.1.4 Local Curvature of a Line: For each pixel of the thinned line the pixel nearest to the barycenter of the actual pixel and its two neighbours is computed. The reciprocal value of the radius of the circle passing through this barycenter pixel and the next but one neighboured pixels is used as approximation of the local curvature of the line, Figure 4.

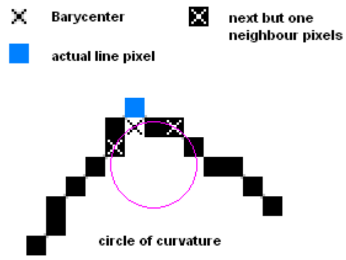


Fig. 4: A sample circle for the approximation of the local curvature of a line

3.1.5 Thickness of a Line: In order to compute the local thickness of a line, the following steps are required:

- Computation of the thinned version of the line
- Evaluation of the gradient direction at each pixel of the thinned line.

The thickness of a line is twice the distance from the thinned version of the line (= center line) to the boundary of the binary version of the line. To compute this distance we evaluate the length of the direct connection of each center line pixel to the boundary of the thick line in gradient direction with the same method as in 3.1.2. Let v be the vector built with the two next but one neighbour line pixels of a center line pixel. The gradient direction in our center line pixel is taken as the normal vector of the vector v. Therefore small directional changes of the center line do not affect the direction of the gradient vector.

3.2. Computations on the Fine Resolution of a Line

For analysing the texture of a thick line, computations within small masks over the line are made. Taking each pixel from the center line (thinned version of the line) as the center of the mask with sidelength = (mean thickness of the line · 2) features like fractal dimension or entropy are evaluated. For better adaptation of the masks to the form of the line, the line is cut in smaller almost linear pieces which are rotated in vertical position before feature evaluation.

3.2.1 Fractal Dimension: The fractal dimension is a measure for the coarseness of objects and textures. Evaluating the fractal dimension of a graylevel image we get a measure for the coarseness of the texture of a digital image.
There are many methods to compute the fractal dimension for graylevel images. We used two different types: capacity dimension (or box dimension) and information dimension, q.v. [Reu01].

3.2.2 Cooccurrence Matrix: In order to create a cooccurrence matrix of a graylevel image, for each pair of pixels having distance δ in the image independent of their directional context, the number of the occurrence of the pixels' graylevels is computed. An element (i,j) of the cooccurrence matrix is the number of graylevel pairs (i,j) with distance δ from each other.

Haralick, q.v. [Har79] defined 13 features that extract statistical attributes from the cooccurrence matrix data. We computed only a subset, i.e. five of these 13 values, namely the trace of the cooccurrence matrix, the energy, entropy, contrast, correlation and the inverse difference moment.

In addition to those Haralick values, we also evaluated 2 features from Conners, Trivedi and Harlow, q.v. [CTH84]: Clustershadow and clusterprominence.

3.2.3 Abruptness of Maxima and Minima: A 2 dimensional graylevel image can be interpreted as a 3 dimensional surface, i.e. a grayscale $f(x, y)$ (height) relates to a pixel with the coordinates (x, y). So there are local extrema in an image. Fine textures have a large number of small–sized local extrema, coarse textures are represented by a smaller number of large–sized local extrema. We define the abruptness of a local extremum by the sum of height–differences between the local extremum and its 8–connectivity neighbourhood pixels:

f = graylevel image of size $M \times N$.

The abruptness of maximum or minimum in the pixel $(x, y) \in \{0, 1, \ldots, M-1\} \times \{0, 1, \ldots, N-1\}$ is defined as:

$$abruptness(x, y) = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} |f(x, y) - f(i, j)|$$

From the resulting graylevel distribution of the abruptness image parameters like mean, variance, skewness and curtosis are computed.

To count the number of maxima and minima in a line, we count all those line pixels, whose graylevel is bigger than that of its neighbours or smaller than that of its neighbours respectively.

3.2.4 Edge Detection Within a Line

The edge image of a line also serves as a texture descriptor. Thereby we compute the gradient image of a line in fine resolution. Statistical parameters of the resulting graylevel distribution are evaluated.

3.2.5 Number of Holes in a Line

The number of holes in a line says something about the inner structure of the line, i.e. the painting material. A line with many holes was maybe painted with wax crayon, contrary to a line with a small number of holes, which was maybe painted with water colors.

In short the number of holes relative to the whole area of the line is related to the writing material.

4. Feature Selection and Interpretation

Considering the set of all evaluated feature values of lines the features have to be selected concerning their classification rate. This means to find those features, that relate the greatest amount of lines to the right artists. At first we used a neural network containing only one feature of the set. With a common feed forward feature selection algorithm we select the best feature and construct a new network with the selected best feature combined with all remaining features. Iterating this and stopping, when the classification rate can't be improved anymore, we get one of the best feature combinations.

In the following example 2 lines from 2 different artists are analyzed and the results are compared and interpreted.

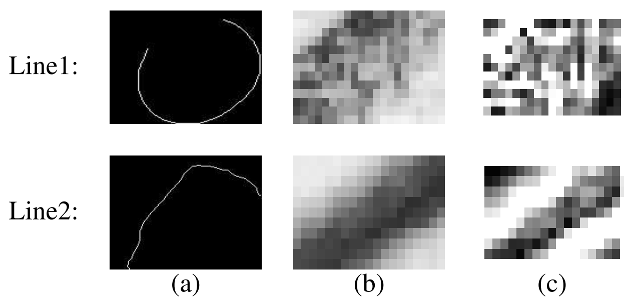


Fig. 5: Two lines, from left to right: (a) extacted line; (b) part of graylevel distribution of (a); (c) edge image of (b).

features	Line1	Line2
line length	1805.7	1751
distance of the endpoints of the line	486.8860	1170.2461
mean thickness	54.18	41.41
mean of local curvature of the line	0.08677	0.13176
chaincode standard deviation	0.3205	0.4714
mean fractal information dimension	2.1559	2.0999
mean of abruptness image	2.185	0.662
variance of abruptness image	719.47	275.76
cooccurrence entropy	1.96	0.73
cooccurrence correlation	2086676.23	544874.72

Interpretation

The detailed interpretation and analysis of the above feature values of the two lines in Figure 5 results in the following conclusions:

- Considering line1 the rate between the actual length of the line and the length of the direct connection of the endpoints of the line is nearly 4; therefore the line is not straight but probably curved. In comparison this rate for line2 is still high but smaller than for line1.

This conclusion proves true when looking at the images of the two example lines.

- The mean of local line curvatures and the chaincode variation along the line are both measures for the curvature of a line. Though line2 is approximately straighter than line1 (compare also the point above), the local curvature and chaincode variation along the line are greater for line2. Therefore line1 is smoother than line2.
- The value of the fractal information dimension of the graylevel distribution of the thick line for line1 is greater than for line2. Because of that the inner structure of line1 has a greater degree of randomness than line2.
- The mean and variance of the abruptness image of line1 are both higher than these values for line2. That confirms the previous observation for the fractal dimension.
- Also the values of the entropy and the correlation of the cooccurrence matrix reflect the interpretation above.

5. Classification Results

By analyzing lines with the above methods, we obtain 52 line descriptors for further processing. As mentioned at the beginning our actual goal is the classification of paintings with respect to the artist. With this basis we searched for qualified measures for the classification of images to the respective artist using neural networks (compare section 4). Using the feed forward method eight features provided one of the best classification results, namely the linelength, number of holes in a line, median and mean of the graylevel distribution of the line, mean of the abruptness of maxima and minima, median and standard deviation of the edge image of a line. Here are the classification results for our five artists:

artist	total number of lines	number of correct classified lines	percentage of correct classified lines
bg	128	75	58.59%
he	100	85	85.00%
hf	115	58	50.43%
rn	83	80	89.89%
th	89	72	86.75%
total	515	370	71.84%

As we only consider lines of the images for classification with respect to the painter the results are very promising. The reason for the difference in the number of lines lies in the fact that we extract lines from six images per artist and each image contains a different number of lines.

6. Conclusions and Outlook

We presented a method for line extraction from a graylevel image and several methods for line analysis with the goal of representing the duct of a line and for further classification of paintings. In considering lines in both coarse and fine resolution, we get on the one hand shape and on the other hand material descriptors.

By now only single lines are extracted and analyzed. We improved the common chaincode algorithm, so that it can handle closed or open lines, single objects or systems of lines. Especially in arts analysis, this is very important, because there are often cases, in which a single line shouldn't be separated from others (i.e. hatching). If we consider larger systems of lines of an image additional structural features could be evaluated. Interpreting this system as a graph, features like number of vertices, weights and number or length of edges could be computed.

Another possible appoach could also be the analysis of lines with respect to their colour distribution.

References

[CST00] Jian Chen, Yoshinobu Sato, and Shinichi Tamura. Orientation space filtering for multiple orientation line segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(5):417–429, 2000.

[CTH84] R. W. Conners, M. M. Trivedi, and C. A. Harlow. Segmentation of high-resolution urban scene using texture operators. *Computer Vision, Graphics and Image Processing*, 25:273–310, 1984.

[FS98] Hans G. Feichtinger and Thomas Strohmer, editors. *Gabor analysis and algorithms*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston Inc., Boston, MA, 1998.

[Gra78] G. H. Granlund. In Search of a General Picture Processing Operator. *Computer Graphics and Image Processing*, 8(2):155–173, 1978.

[Har79] R. M. Haralick. Statistical and structural approaches to texture. *Proc. IEEE*, 67:786–804, 1979.

[IZ95] Lee A. Iverson and Steven W. Zucker. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):982–996, 1995.

[Reu01] M. Reuter. Fraktaldimension von Grauwertbildern. 2001.

[Ste98] Carsten Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, February 1998.