Multimed Tools Appl
DOI 10.1007/s11042-010-0561-8

Automatic segmentation of digitalized historical manuscripts

Costantino Grana · Daniele Borghesani · Rita Cucchiara

Abstract The artistic content of historical manuscripts provides a lot of challenges in terms of automatic text extraction, picture segmentation and retrieval by similarity. In particular this work addresses the problem of automatic extraction of meaningful pictures, distinguishing them from handwritten text and floral and abstract decorations. The proposed solution firstly employs a circular statistics description of a directional histogram in order to extract text. Then visual descriptors are computed over the pictorial regions of the page: the semantic content is distinguished from the decorative parts using color histograms and a novel texture feature called Gradient Spatial Dependency Matrix. The feature vectors are finally processed using an embedding procedure which allows increased performance in later SVM classification. Results for both feature extraction and embedding based classification are reported, supporting the effectiveness of the proposal on high resolution replicas of artistic manuscripts.

Keywords Manuscript · Image segmentation · Texture analysis

1 Introduction

The automatic analysis of the huge amount of paper documents represents a concrete and attractive possibility in terms of data retrieval (text and images) and data presentation. This activity becomes much more important when dealing with artistic or historical documents that cannot be available to the public, due to their value and delicacy. Computer science can fill the gap between people and all these precious libraries: digital versions of the artistic works can be publicly accessible, both locally at the museum owning the original version and remotely. In this manner, users—

C. Grana (✉) · D. Borghesani · R. Cucchiara
Università degli Studi di Modena e Reggio Emilia, Via Vignolese 905/b, 41100 Modena, Italy
e-mail: costantino.grana@unimore.it

Published online: 09 July 2010

either experts, tourists or people keen on art—can explore more comprehensively the document, choosing their own personal way to browse and enjoy it.

Italy, in particular, has a huge collection of illuminated manuscripts, but many of them are not freely accessible to the public. These masterpieces contain thousands of valuable illustrations: different mythological and real animals, biblical episodes, court life illustrations, and some of them even testify the first attempts in exploring perspective for landscapes. Usually manual segmentation and annotation for all of them is dramatically time consuming. For this reason, the accomplishment of the same task with an automatic procedure is very interesting, and moreover really challenging due to the visual appearance of these pictures (their arrangement over the page, their various framing into the decorative parts and so on).

In this work we propose a solution for the manuscript layout segmentation and the automatic extraction of valuable pictures from the decorated pages by means of visual cues, independently by the layout. This functionality aims at providing the basis for content-based activities such as searches for similarity, comparison, recognition of specific elements (people, life scenes, animals, etc...). The main novelties within the segmentation process is the use of a new texture feature aimed at detecting the correlations between the gradient directions (namely GSDM) and a clustering-based embedding process which allows to reduce the training requirements of learning algorithms both in terms of number of samples and computational time, without impacting on the classification performance.

The paper is structured as follows: in the next Section 2 related work is discussed, Section 3 presents the system architecture, Section 4 details the text segmentation step, while Section 5 describes our proposal for layout segmentation, feature extraction and classification. Finally, results are reported in Section 6, followed by a glimpse to the future work on schedule about this project in Section 7.

2 Related work

Document analysis is one of the most explored fields in image analysis, and a plethora of work has been produced dealing with different aspects of the segmentation of the document. The seminal work of Nagy [26] gives the perfect overview of the techniques proposed until some years ago for text segmentation, OCR and background removal. The most faced problem is clearly text detection and interpretation, either printed text or handwritten characters, but approaches dealing also with pictures segmentation have been studied.

In [4], Chen et al. provide a general partition of the classification approaches proposed so far. In particular, according to their taxonomy, the page can be classified using:

- Image features: global and local descriptors for color, shape, texture, gradients and so on.
- Physical layout features: a hierarchical description of the objects in the page, based on their geometric arrangement.
- Logical structure features: a hierarchy of logical objects, based on the human-perceptible meaning of the document contents. For example, the logical structure of a manuscript is a hierarchy of logical objects, such as title, authors and chapters.

– Textual features: the presence of keywords, computed from OCR applications or directly from document images.

Several works tackle the physical and logical segmentation of the page, exploiting different rules on the page structure, such as geometric constraints over the layout. A different strategy is to compute specific descriptors followed by classification: an example is provided by Diligenti et al. in [6] which exploited Hidden Tree Markov models. The majority of these works employ an XY-tree based representation, and graph or template matching approaches in order to perform classification. Our work belongs to Chen's first class, based on image features.

Among image features, color, shape and texture are generally extracted. In this context, shape features are not so meaningful because of the absence of fixed frames containing pictures. On the other, it's interesting to mention a work by Bigun et al. [2] that aims at overcoming the limitations of the shape-based approach in particular for decoration segmentation: in this work, a decomposition of the original image in its iso-orientation components (called *orientation images*) is performed; each orientation image contains information about the linear structures in the original image which are oriented along the relative orientation, summed and furtherly composed as a feature vector called an *orientation radiogram*.

Color is a useful descriptor successfully exploited in a lot of CBIR systems, but it is not particularly powerful if used alone. Finally, texture features are quite useful since the patterns we are trying to classify have some distinguishing characteristics from the textural point of view. In particular texture features based on frequencies and orientations have been used in [18] to extract and compare elements of high semantic level, without expressing any hypothesis about the physical or logical structure of the analyzed documents, and exploiting a page analysis by blocks. Nicolas et al. in [27] proposed a 2D conditional random field model to perform the same task. Hu et al. [14] use interval encoding features to capture elements of spatial layout, modeled with HMMs. Using grey level images, histogram projection is used in [25] to distinguish text from images, while a more complex approach based on effective thresholding, morphology and connected component analysis has been used in [20]. A multiscale approach has also been proposed in [7] by Fataicha et al.

All these techniques are the basis of complete systems for the management of digital libraries (DL), tools for semantic annotation, classification and retrieval. For the implementation over a large collection of digital documents, the accuracy of the analysis and the computational effort required are both significant. Until now, most of the activities on DL of illuminated manuscripts have been accomplished by manual annotation and indexing, but some interesting systems deserve to be mentioned.

The AGORA [31] software performs a map of the foreground and the background and consequently proposes a user interface to assist in the creation of an XML annotation of the page components. The Madonne system [28] is another French initiative to use document image analysis techniques for the purpose of preserving and exploiting cultural heritage documents.

In [24], Le Bourgeois et al. highlighted some problems with acquisition and compression, then authors gave a brief subdivision of documents classes, and for each of them provided a proposal of analysis. They distinguished between medieval manuscripts, early printed documents of the Renaissance, authors manuscripts from 18th to 19th century and, finally, administrative documents of the 18th–20th century. In this work, the authors performed color depth reduction, then a layout segmenta-

tion that is followed by the main body segmentation using text zones location. The feature analysis step uses some color, shape and geometrical features, and a PCA is performed in order to reduce the dimensionality. Finally the classification stage implements a K-NN approach. Their system has been finalized in the DEBORA project [23], which consists of a complete system specifically designed for the analysis of Renaissance digital libraries. In this paper we are interested in the first class identified by [24], that is composed of illuminated manuscripts.

3 System architecture

The approaches for text and image segmentation and classification presented in this paper are implemented in an integrated system for document analysis and remote access, including basic querying and browsing functionalities. The system elements are reported in Fig. 1. Two different databases have been created in order to store images and annotations. The former stores the high resolution digitized manuscripts, while the latter contains both the automatically extracted knowledge and later in the future it will contain also the historical comments added by experts and automatically linked to the pictures extracted.

The retrieval subsystem shares the canonical structure of CBIR systems. Retrieval results are visualized by the user interface module, that integrates the visual/keyword-based search engine to propose an innovative browsing experience

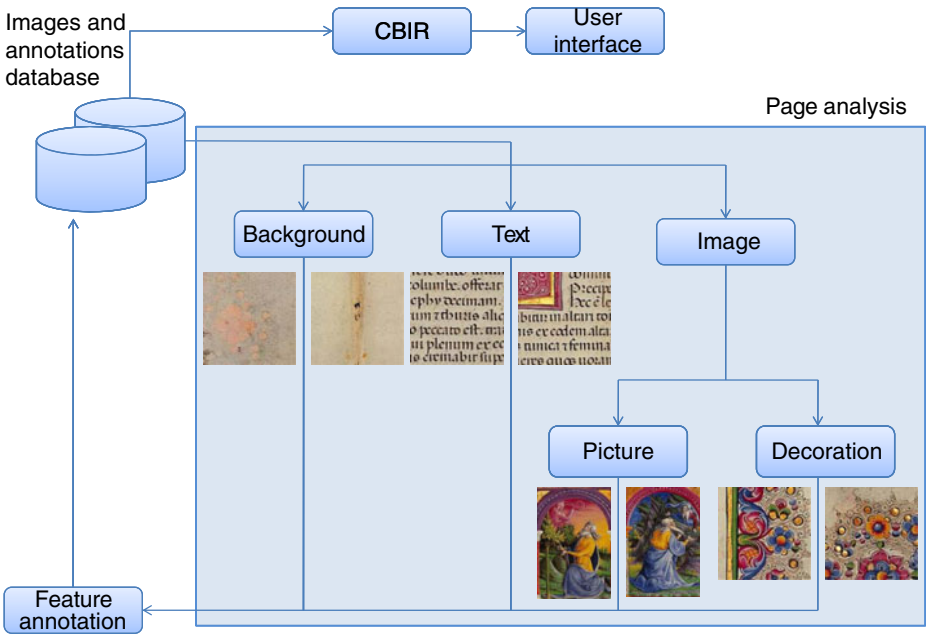


Fig. 1 Overall schema of the system: visual descriptors based on color and texture are composed to distinguish firstly between background, text and images, and later between decorations and meaningful pictures

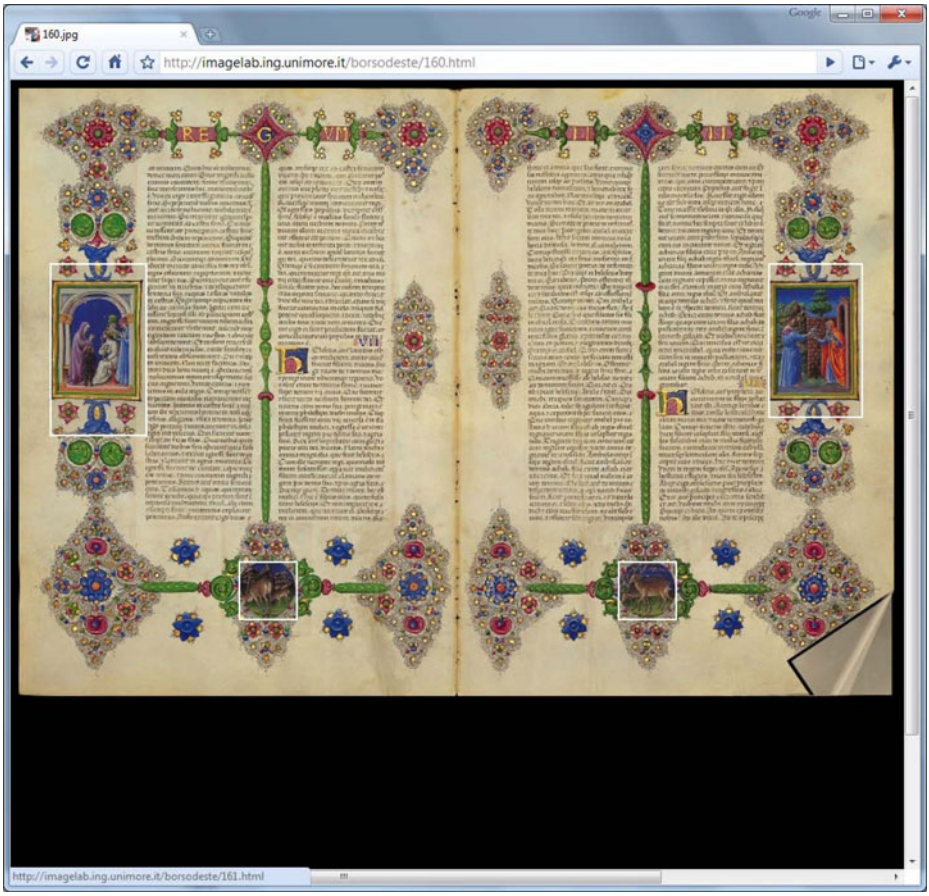


Fig. 2 Screenshot of the web interface, showing an example of automatic segmentation

to the user. In Fig. 2 an example of the final result of the automatic extraction is provided.

An offline page analysis module process the stored images and for each of them detects text and images. The text areas are then stored in the annotation database, in order to potentially allow a later application of OCR functions, or visual keyword spotting [22]. Then, the procedure distinguishes pictures within the decorations and extracts them separately. The details of these steps are fully described in the following sections. After the segmentation, the areas are saved into the annotation database and a feature extraction stage is performed over the picture areas, to allow CBIR functions such as similarity-based retrieval on their visual appearance.

4 Text segmentation

We propose a new method based on texture analysis to distinguish between textual and pictorial regions. Our method is based on an approach proposed in [18],

improved with the adoption of circular statistics, in particular the Von Mises distributions [10]. The basic feature we exploited is the autocorrelation matrix, an effective feature in this case since textual textures have a pronounced orientation that heavily differs both from background and decorations. Formally, the autocorrelation function is the cross correlation of a signal with itself, and it represents a measure of similarity between two signals. Once applied to a grayscale images, it produces a central symmetric matrix, that gives an idea of the degree of regularity of the texture. Our method proceeds subdivinding the original image into square blocks. The size bs of these blocks must be set according to the scale at which the texture should be analyzed. In order to sufficiently catch the horizontal orientation information, a good empirical choice has been determinated as the height of 6 text lines. The definition of the autocorrelation for a block is:

$$C(k,l) = \sum_{y=\max(0,l)}^{bs-1+\min(0,l)} \sum_{x=\max(0,k)}^{bs-1+\min(0,k)} I(x,y) \cdot I(x+k, y+l) \tag{1}$$

where l and k are defined in $[-bs/2, bs/2]$. The result of the autocorrelation can be analyzed extracting an estimate of the relevant directions within the texture. Each angle determines a direction, and the sum of all the pixels along each direction is computed to form a polar representation of the autocorrelation matrix, called *direction histogram*. In this way, each direction will be characterized by a weight, indicating its importance within the block.

$$w(\theta) = \sum_{r \in (0, bs/2]} C(r\cos\theta, r\sin\theta) \tag{2}$$

Since the autocorrelation matrix has a central symmetry by definition, we consider only the first half of the direction histogram in the range $[0°, 180°)$. ω and r are quantized: the step of ω is set to 1°, and the step of r is set to 1 pixel. A text block will be characterized by peaks around 0° and 180° because of the dominant direction is horizontal, and this behavior is different compared to pictorial textures (described by a generic monomodal or multimodal distribution) and also background textures (described by a nearly uniform flat distribution).

The polar distribution obtained by autocorrelation in the previous step is modeled in a statistic framework. The standard Gaussian distributions are inappropriate to model angular datasets: regarding our specific case, there is a discontinuity between 0° and 180°, so these two angles vote for two distinct directions even if they express the same one, producing a bad fitting. Instead, we exploit Von Mises distributions [3] that can correctly represent angular datasets. This statistical formulation has been rarely used for texture analysis, while it has been previously presented for trajectory shape classification [30]. The probability density function is defined as follows:

$$V\left(\theta|\tilde{\theta}, m\right) = \frac{1}{2\pi I_0(m)} e^{m\cos(\theta-\tilde{\theta})} \tag{3}$$

The parameter m denotes how concentrate the distribution is around the mean angle $\tilde{\theta}$. In our context, we used a slightly different formulation (we simply multiply the angles by a factor of 2) with a periodicity of π instead of 2π, considering only angles in $[0, \pi)$ representative for valuable and meaningful directions. I_0 is the modified order 0 Bessel function.

To catch the general multimodal behavior of input datasets, we chose a mixture of Von Mises distributions. We used mixtures with two components only, because

they proved to be sufficient in order to recognize the two most meaningful directions (horizontal and vertical) while keeping an affordable computational cost. An example of fitting for the two types of texture analyzed is shown in Fig. 3. The background is also characterized by an almost flat distribution. Generally, a mixture of K Von Mises distributions is defined as follows:

$$M(\theta) = \sum_{k=1}^{K} \alpha_k V\left(\theta|\tilde{\theta}, m_k\right) \tag{4}$$

where α_k represents a weight of the distribution within the mixture. A convenient way to get the maximum likelihood estimates of the mixture parameters is the Expectation–Maximization algorithm [30]. In the E step, the expected values for the likelohood are computed, then a set of parameters to maximize such values are obtained, repeating the process until convergence or maximum number of iterations is reached. To maximize the likelihood, a set of *responsibilities* of the bins for each Von Mises is necessary. Let θ be the index of the bin. The responsibilities are computed as follows:

$$\gamma_\theta = \frac{\alpha_k V\left(\theta|\tilde{\theta}_k, m_k\right)}{\sum_{s=1}^{K} \alpha_s V\left(\theta|\tilde{\theta}_s, m_s\right)} \tag{5}$$

A new set of weights for the Von Mises of the mixture can now be computed using (6).

$$\alpha_k = \frac{\sum_{\theta\in[0,\pi)} w_\theta \gamma_{\theta_k}}{\sum_{\theta\in[0,\pi)} w_\theta} \tag{6}$$

This formulation differs from the one in [30], and the motivation lies on the dataset we used: we do not have a general distribution of angular data to fit, but a sampling of directions and relative weights. For this reason, we consider the weight as a multiplier value for each angle, so formally we have w_θ times the angle θ in our dataset. In the

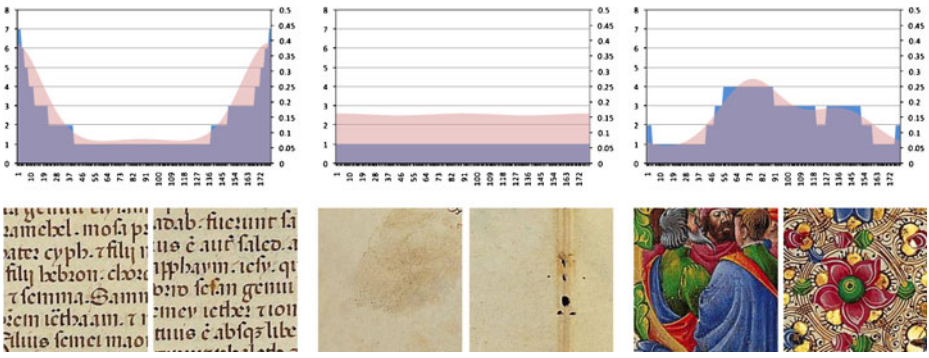


Fig. 3 Example of directional histograms and the corresponding fitting with Von Mises mixtures, respectively for text, background and pictorial textures

M step, we compute the new θ and m values for each Von Mises within the mixture. In particular, θ is computed by maximization of the relative likelihood using (7).

$$\tilde{\theta_k} = \arctan\left(\frac{\sum\limits_{\theta\in[0,\pi)} w_\theta \gamma_{\theta_k} \sin 2\theta}{\sum\limits_{\theta\in[0,\pi)} w_\theta \gamma_{\theta_k} \cos 2\theta}\right) \tag{7}$$

Note the multiplication by a factor of 2 in order to adapt to a π-periodicity. The retrieval of m by maximization is a bit more complicated, due to the presence of the Bessel functions. Given the derivative of the modified Bessel function I_1, the problem could be mathematically solved using this formulation:

$$A(m_k) = \frac{I_1(m_k)}{I_0(m_k)} = \frac{\sum\limits_{\theta\in[0,\pi)} w_\theta \gamma_{\theta_k} \cos\left(\theta - \tilde{\theta_k}\right)}{\sum\limits_{\theta\in[0,\pi)} w_\theta \gamma_{\theta_k}} \tag{8}$$

The value of m_k can be found by the numerical inversion of $A(m_k)$. In particular we use the approximation proposed in [8].

At this point, we have six parameters to play with: α_1, α_2, $\tilde{\theta_1}$, $\tilde{\theta_2}$, m_1 and m_2 of both Von Mises distributions. This represents a very consistent and compact way to describe a whole distribution, making the retrieval faster and effective. An example of modeling is proposed in Fig. 3.

The similarity between two Von Mises distributions can be defined using the Bhattacharyya distance. Given two Von Mises distributions V_1 and V_2, the formulation is shown in (9).

$$B(V_1, V_2) = \sqrt{1 - \sqrt{\frac{1}{I_0(m_1) I_0(m_2)}}\, I_0\left(\frac{\sqrt{m_1^2 + m_2^2 + 2m_1 m_2 \cos 2\left(\tilde{\theta_1} - \tilde{\theta_2}\right)}}{2}\right)} \tag{9}$$

No explicit form is available for mixtures, so we propose a metric that also takes into account the relative weights of the components of the mixture. Given two mixture distributions $M^i(\theta) = \sum\limits_{k=1}^{2} \alpha_k^i V\left(\theta|\tilde{\theta_k^i}, m_k^i\right)$, we computed the Bhattacharyya distance between pairs of distributions and selected the best matching two (calling them b, while the other two o). Then we measure the distance as:

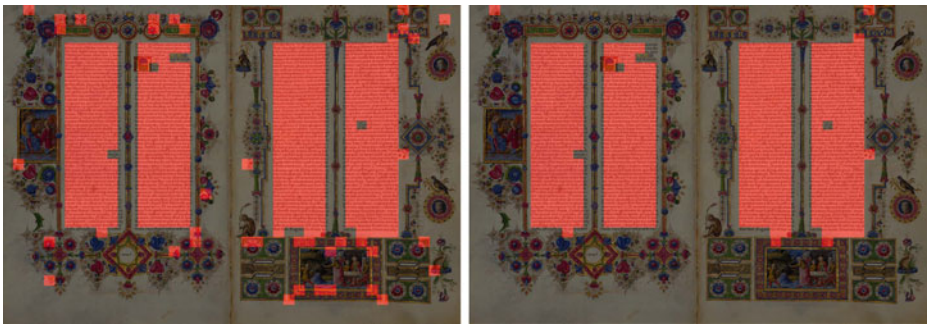
$$d\left(M^1, M^2\right) = \frac{WB_b + WB_o}{\alpha_b^1 \alpha_b^2 + \alpha_o^1 \alpha_o^2} \tag{10}$$

where

$$WB_x\left(M^1, M^2\right) = \alpha_x^1 \alpha_x^2 B\left(V_x^1, V_x^2\right) \tag{11}$$

This metric takes into account the fact that two components can be very similar, but their contribution to the mixtures can be quite low.

Given a manually annotated training dataset, the segmentation by blocks using a SVM classification and the radial basis function as kernel provides a recall of



(a) (b)

Fig. 4 Example of segmentation results for the text regions, before (**a**) and after the postprocessing step (**b**). The algorithm produces a good segmentation that can be later exploited to define the regions to be processed by a OCR algorithm for text extraction and the regions to be processed for picture extraction, as we will see in the next sections

91.14% with a precision of 88.40% for text. Many techniques have been proposed in the literature to improve the classification output by means of neighborhood information. For example, we can see this problem as an image denoising problem, so we can exploit SVM output, representing the classification value of each block, with a relaxation labeling [21], we can model it with Markov Random Fields [1] or Hidden Markov Trees [5], or the segmentation could be obtained with graph cuts. The idea is to extend the neighborhood properties (in terms of labels or classification values) to the current block in order to force a neighborhood consistency, thus eliminating classification outliers. In our case, the simple procedure of filling isolated blocks boosts the precision value up to 95.76%. An example of segmentation results (with and without the postprocessing step) is proposed in Fig. 4.

The proposed technique is designed mainly for text segmentation, but the visual characteristics of the pictorial regions and the background are unique insomuch that we can use the results of text segmentation as a preprocessing for the next stages. So all the blocks presenting a generic multimodal distribution are considered as belonging to a potential pictorial region, while all the blocks with a nearly flat distribution are discarded as belonging to the background. We did not employ this algorithm directly for the segmentation of the pictorial data since the classification results for this textures proved to be much less attractive. In fact the classification of pictorial blocks produced a recall of 91.0% (boosted to 97.53% with postprocessing) with a precision of 68.13, while the background classification had a poor recall (around 40%) with a very high precision (more than 99%).

Once the text is removed, the (remaining) potential pictorial regions are then processed for meaningful picture extraction.

5 Automatic picture extraction

Miniature illustrations detection begins with a preprocessing stage on the remaining regions in order to distinguish between background and pictorial data. The result is a binary mask highlighting both pictures and decorations. Since morphological or

pixel level segmentation are not enough to separate them, a block based analysis is performed and a feature vector is extracted for each block. Finally a SVM is used to classify and separate them.

5.1 Image areas extraction

As shown in Fig. 3, the background has a nearly flat distribution and the overall chromatic range is quite distinguishing. For these reasons, we used a more refined and simple binarization with automatic thresholding: a lot of approaches have been proposed in literature (see [32] for a complete survey), but in our context the Otsu algorithm (Fig. 5a) proved to be sufficiently robust to the noise. This algorithm selects the optimal threshold k^* maximizing the between-class variance σ_B^2 of gray levels $k \in [0, \ldots, L-1]$ following a sequential search.

$$\begin{aligned} \sigma_B^2(k) &= \max_{0 \le k \le L-1} \sigma_B^2(k) \\ &= \max_{0 \le k \le L-1} \frac{\left[\mu_T \omega(k) - \mu(k)\right]^2}{\omega(k)\left[1 - \omega(k)\right]} \end{aligned} \tag{12}$$

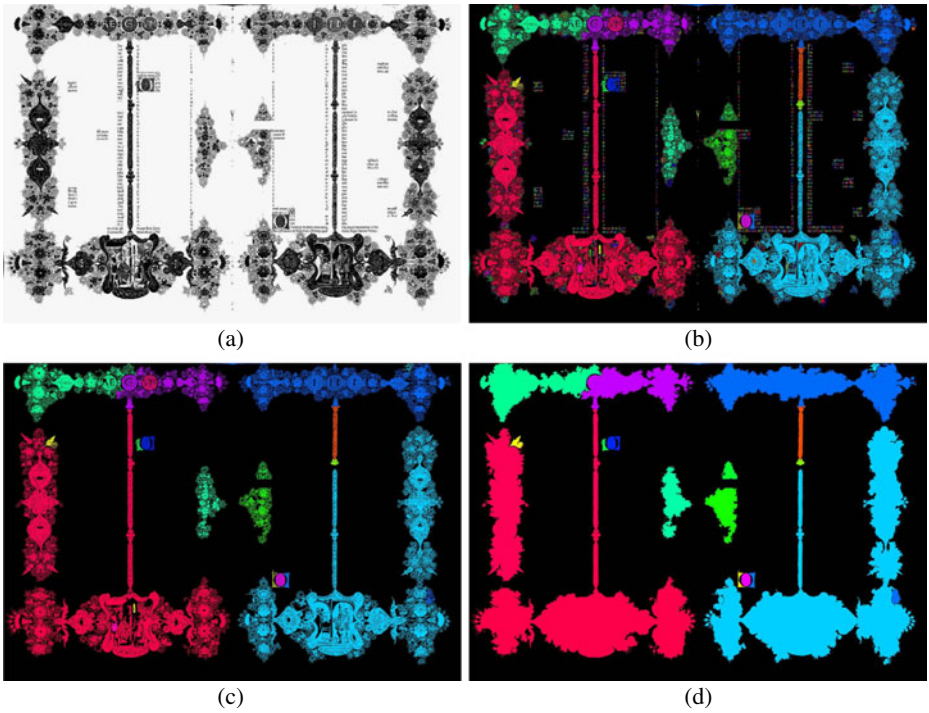


Fig. 5 Steps of the preprocessing stage. After the text removal, the image is passed through the Otsu thresholding (**a**), then a connected components analysis is performed (**b**), the smallest blobs are removed from the result (**c**) and finally a filling procedure is performed (**d**)

given

$$\omega(k) = \sum_{i=0}^{k-1} p_i \mu(k) = \sum_{i=0}^{k-1} i p_i \tag{13}$$

the zeroth- and first-order cumulative moments of the histogram up to the kth gray level respectively and

$$\mu_T = \mu(L) = \sum_{i=0}^{L-1} i p_i \tag{14}$$

the total mean gray level of the original picture.

Other techniques specifically proposed for printed documents, namely Iterative Global Thresholding [19], are often too aggressive on the thin decoration borders, causing too much shrinking on the detected areas.

The connected components of the image are then labeled (Fig. 5b). The labeling in our context is particularly demanding since we work with high resolution images with tenths of thousands of connected components. To improve this stage, we employ a fast connected components analysis technique based on a 2×2 block optimization and an effective array-based data structure for label resolution [11]. Blobs' area is computed, and blobs smaller than a minimum area (empirically defined) are removed in order to focus on the larger ones (Fig. 5c). The contour of each blob is then followed and filled (Fig. 5d). The resulting pixels are used as a mask for the next stages of the processing.

5.2 Block level features

The image areas, as identified by the preprocessing output mask, are analyzed at block level, using a sliding window. The window size has been set depending on the image resolution; in our experiments it was set to 200×200 pixels for images of $3{,}894 \times 2{,}792$ pixels. To ensure an effective coverage of the images, the window is moved so to obtain an overlap of 80% of its area between each step, in order to have multiple blocks for each region. For each block, a set of color and texture features is extracted; in particular we adopt both *RGB Histogram* and *Enhanced HSV Histogram* as color features, and we propose a new texture descriptor named *Gradient Spatial Dependency Matrix (GSDM)*.

RGB histogram This is a 3D color histogram built on the RGB components of the image. Each component is quantized to eight values, resulting in a 512-bin histogram. Each bin of the resulting histogram is then normalized so that they add up to one.

Enhanced HSV histogram The idea of this feature is to separately account the chromatic and achromatic contribution of pixels. To this aim, four bins are added to the standard MPEG-7 HSV histogram, resulting in a 260-bins descriptor that proved to be more robust to bad quality or poorly saturated images [9]. This representation provides an advantage with respect to the standard HSV histogram definition because images have been depicted by hand, so they do not have photographic quality, despite of their high resolution digitalization.

GSDM This feature is inspired to the well known Haralick's grey level co-occurrence matrix (GLCM) [12], which provides a representation of the spatial distribution of grey-scale pixels of the image. Unlike GLCM, this new representation accounts for the spatial distribution of gradients within the image.

The original image I is convolved with a Gaussian filter G with $\sigma = 1$.

$$I_g = I * G \tag{15}$$

The filtered image I_g is then used to compute the horizontal and the vertical gradients images with central differences computation.

$$\begin{aligned} G_x(x, y) &= I_g(x+1, y) - I_g(x-1, y) \\ G_y(x, y) &= I_g(x, y+1) - I_g(x, y-1) \end{aligned} \tag{16}$$

and the module and the direction of the gradient for each pixel $\mathbf{p}$ is straightforward:

$$M(\mathbf{p}) = \sqrt{G_x(\mathbf{p})^2 + G_y(\mathbf{p})^2} \tag{17}$$

$$D(\mathbf{p}) = \begin{cases} \frac{\pi}{2}, & \text{if } G_x(\mathbf{p}) = 0 \\ \left(\tan^{-1} \frac{G_y(\mathbf{p})}{G_x(\mathbf{p})} + \pi\right) \bmod \pi, & \text{otherwise} \end{cases} \tag{18}$$

Finally D is uniformly quantized into Q using eight levels. Given $L = L_x \times L_y$ the set of pixel coordinates of the grayscale image I, with $L_x = \{0, 1, \ldots, N_x - 1\}$ and $L_y = \{0, 1, \ldots, N_y - 1\}$ the x and y spatial domains, in order to summarize the relations between the gradients of neighbor pixels, we define $C_\delta(i, j)$ as the set of all point couples displaced by vector δ, with quantized gradient directions i and j respectively:

$$C_\delta(i, j) = \{\mathbf{r}, \mathbf{s} \in L | Q(\mathbf{r}) = i, Q(\mathbf{s}) = j, \mathbf{r} - \mathbf{s} = \delta\}. \tag{19}$$

Since we are also interested in the strength of the texture, the magnitude of the gradients is considered in the final matrix:

$$P_\delta(i, j) = \sum_{(\mathbf{r}, \mathbf{s}) \in C_\delta(i, j)} M(\mathbf{r}) + M(\mathbf{s}) \tag{20}$$

In our setup, δ was taken in the set $\{(1, -1), (1, 0), (1, 1), (0, 1)\}$, that contains the four main directions $\{45°, 0°, -45°, -90°\}$ at 1 pixel distance. Concluding, the feature used is composed by four square matrices with size 8×8, leading to a 256-dimensional feature vector. In Fig. 6 we provided a view of these four matrices, highlighting the appearance differences between the ornamental and the meaningful picture textures.

5.3 SVM classification in an embedded space

Support Vector Machines are a common technique for data classification [3]. Given a training set of n labeled instances $(\mathbf{x}_i, \mathbf{y}_i)$ for $i = 1 \ldots n$, where $\mathbf{x}_i \in \Re^n$ and $\mathbf{y}_i \in$

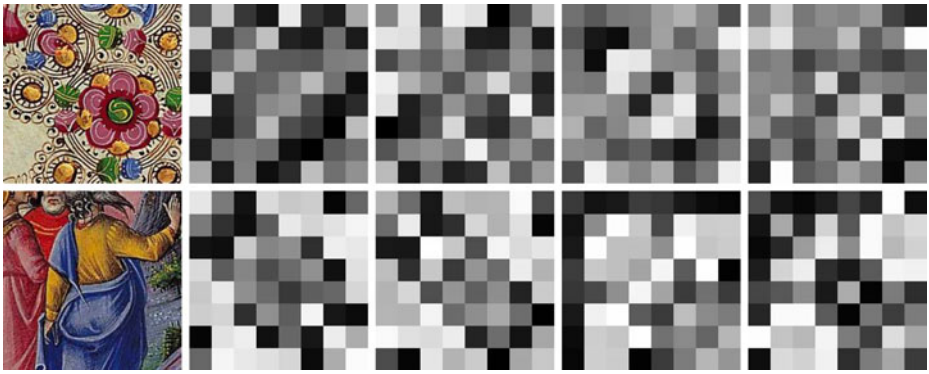


Fig. 6 Visualization of the GSDM output for an ornamental and a picture block.

$\{+1, -1\}$, SVMs find out a linear separating hyperplane (defined by the support vectors) with the maximal margin in a higher dimensional space. SVM can perform a non-linear classification by means of the kernel trick: the dot products are replaced with non-linear kernel functions with the property of distance in feature space and with the positive semidefinite matrix for all elements.

One remarkable property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space [16]. In this particular application this is not totally true. We deal with a 512-dimensional feature vector for the RGB histogram, a 260-dimensional feature vector for the enhanced HSV and a 256-dimensional feature vector for GSDM, so globally we deal with a 1,028-dimensional feature vector. In our tests, in order to obtain acceptable performance we had to use a large number of training samples and these were directly related to the number of features employed. The use of an RBF kernel, usually providing better performance than linear or polynomial ones, required unacceptable training times with our training set. Neither a reduction of the size of the training set with this kernel is acceptable because of lower classification performances and overfitting. Indeed, a reduction of the training set size would be particularly useful, in face of the final application scenario, in which the final user could obtain automatic annotation providing fewer manual samples, thus reducing the necessary work.

The amount of data and the dimensionality of feature vectors are challenging problems. A typical example is the similarity searching, in which we want to find the most similar results to a given query in a CBIR system. When we work with large datasets, the number of distances evaluations necessary to complete the task could become prohibitive. In order to limit this amount of computations and at the same time to maintain an acceptable quality of the results, an *embedding* approach can be exploited.

Embedding spaces have been initially introduced to speed up searches by reducing the dimensionality of the space where the search is performed. The goal is to embed the dataset into a different vector space with a lower dimensionality in such a way that distances in the embedded space approximate distances in the original space. In a more formal way, given a metric space S^n with a defined distance d

$$d : S^n \times S^n \rightarrow S^n \tag{21}$$

an embedding can be defined as a mapping F from (S^n, d) into a new vector space $(\mathbb{R}^k, \delta)$ where k is the new dimension and δ is the new distance.

$$\begin{aligned} F &: S^n \to \mathbb{R}^k \\ \delta &: \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}^k \end{aligned} \tag{22}$$

Given two objects o_1 and o_2, the embedding procedure assures that the distance $\delta(o_1, o_2)$ is as close as possible to $d(o_1, o_2)$ in the original space. In particular, an embedding space $\mathbb{R}^k$ has the contractive property if the distance δ provides a lower-bound for the corresponding distance in the original space S^n. The search in the embedding space should be simpler and faster. Thus, if the provided embedding has the contractive property, we assure that the search preserves the quality of results, because we obtain the same recall as in the original space. There are also different ways to measure the quality, for example distortion, stress, Cluster Preservation Ratio. A more detailed description is provided in [13].

Our implementation is directly derived from the Lipschitz embeddings [13]. The key idea is to extract information about an unknown object x given the distances $d(x, o_i)$ between x and a set of arbitrary reference objects o_i: the coordinates of x in the new space are computed as the distance between x and the reference objects. A similar procedure is described in [29], where it is called "mapping onto a dissimilarity space": they use a Regularized Linear/Quadratic Normal density-based Classifier and compare three criteria to select the representation set, namely random, most-dissimilar and condensed nearest neighbor. In order to exploit the distance metric specifically designed for every single feature (or consequently for every group of features, with simple feature fusion approaches), we used Complete Link clustering [15].

Complete Link is a famous hierarchical clustering approach based on the following criterion:

$$x \in C_i \Leftrightarrow d(x, y) > \eta_i, \forall y \in C_j, j \neq i \tag{23}$$

$$\eta_i = \max_{x,y \in C_i} d(x, y) \tag{24}$$

This criterion implies a strong condition on the similarity of elements in a cluster, because each element must be similar to every other in the cluster. In this case, to each cluster C_i is associated a maximum dissimilarity η_i defined by (24), which measures the maximum dissimilarity between any two elements in the cluster. Any other element outside the cluster must have a dissimilarity greater than η_i from any element in the cluster. Hierarchical clustering methods based on Complete Link generate clusters which satisfy the previous condition. For this clustering method we defined the dissimilarity between two clusters C_i and C_j as

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{25}$$

The algorithm proceeds as follows:

1. Initially we have N clusters $\{x_1\}, \ldots, \{x_N\}$. Let's call E the set of clusters. Each cluster contains a single element x_i.
2. Find the most similar pair of clusters, R and S, according to (25), i.e. find R and S such that $d(S, R) < d(A, B), \forall A, B \in E$.

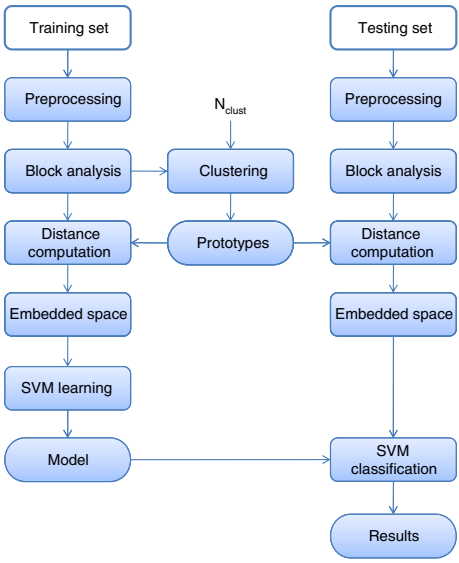


Fig. 7 Diagram that shows the way each image is preprocessed and then analyzed in the learning and the classification procedures

3. Merge R and S into a new cluster.
4. Repeat from Step 2, until the required number of clusters is obtained.

This algorithm produces a hierarchy of elements partitions with at most N levels and i clusters at level i (the initial level is N). To implement the algorithm a proximity matrix D was used. An $N \times N$ proximity matrix $D = \left[d(B_i, B_j)\right]$ contains the dissimilarity between two blocks i and j at element (i, j). At each step, the matrix is updated by deleting rows and columns corresponding to clusters R and S and adding a new row and column corresponding to the newly formed cluster. The values in the new row/column are the maximum of the values in the previous ones. Initial generation of matrix D requires $\frac{1}{2}N(N-1)$ computations of $d(\cdot, \cdot)$.

The final procedure to learn and classify our data blocks, is summarized in Fig. 7. We separately cluster the positive and negative training samples in order to select the most valuable objects which represent the entire sets. These reference examples become the basis of the new embedded space, and the new coordinates of every element in the dataset are computed as their distances with the reference objects, obtaining the feature vectors. Now we can apply the regular SVM learning stage (using the SVMLight library [17]), obtaining our classifier.

For each block, the classification provides a positive value if the block turns out to be mostly (more the 50%) a picture, a negative value otherwise. The overlapping of blocks provides a more refined covering of pictures and decorations, obtaining better results over the borders. A mask is finally provided, highliting all blocks with a positive classification.

6 Experimental results

In this paper, we used the digitalized pages of the Holy Bible of Borso d'Este, duke of Ferrara (Italy) from 1450 to 1471 A.C., which is considered one of the best

Renaissance illuminated manuscript in the world. Tests have been performed among a dataset of 320 high resolution digitalized images (3,894 × 2,792), a total amount of 640 pages. These images have been manually annotated, so half of the pages has been used for training and half for testing. Each page of the dataset is an illuminated manuscript composed by a two-column layered text in Gothic font, spaced out with some decorated drop caps. The entire surrounding is highly decorated. The system aims at extracting the valuable illustrations within the decoration texture (miniature illustrations of scenes, symbols, people and animals), rejecting all the border decorations (ornaments).

Results are reported in terms of recall and precision. The granularity of these results has three levels: pixels, blocks, and regions. Recall and precision at pixel-level are computed comparing the automatic annotation with the raw number of pixels that have been marked by a human operator as valuable pixels. Since we did not yet implement any refinement on the boundaries of the extracted picture (in order to precisely segment it from the decoration), we already expected quite low precision values. Recall and precision at blocks level correspond to the raw recall and precision values outputted by the SVM: based on the ground truth, we labeled each block within the testing set, choosing a positive annotation if the majority of pixels within the block belongs to a valid picture, and a negative annotation otherwise. Finally recall and precision at regions level are computed counting how many blobs have a significant overlap with a corresponding blob in the ground truth.

The first tests were conducted on the features. We computed recall and precision values with different sets of features, in order to verify that a higher number of features could effectively contribute to a better classification. Each feature defines its own way to compute the similarity: in particular, RGB and EHSV histograms exploit a histogram intersection approach, while the GSDM feature performs a sum of point-to-point Euclidean distances between the matrices. These values are standardized, and then arithmetic mean is computed to fuse their results. The tests were conducted applying the previously described embedding procedure firstly to the single features, then to their combination.

Table 1 shows that the addition of different features helps improving the classification performance. In particular, simple information about colors in the HSV space proved to be discriminant enough to distinguish the images from the decorations, since decorations have a limited palette and a major amount of background pixels. Texture information help to significantly increase the precision, and a further improvement on recall values is highlighted. Finally the addition of the RGB histogram seems to propose a good compromise between recall and precision: it boosts precision values with a minimum loss in recall values.

Table 1 Comparison using different feature sets

	RGB (%)	eHSV (%)	GSDM (%)	All %
Re_{pixels}	82.36	80.35	82.42	**83.49**
Pr_{pixels}	53.60	**57.61**	43.32	52.97
Re_{regions}	84.21	81.50	84.21	**85.69**
Pr_{regions}	70.33	**74.91**	57.27	73.36
Re_{blocks}	68.58	62.85	74.60	**75.87**
Pr_{blocks}	84.23	**87.31**	74.23	85.80

Best results are bold faced

Table 2 Comparison with and without the embedding procedure

Samples	10,000	1,000	1,000	1,000
Embedding	No	No	Yes	Yes
Kernel type	Linear	Linear	Linear	RBF
Re_{regions} (%)	84.91	84.03	**85.75**	85.69
Pr_{regions} (%)	73.28	69.57	72.46	**73.36**
Re_{blocks} (%)	74.44	72.81	74.92	**75.87**
Pr_{blocks} (%)	85.90	84.82	85.09	**85.80**
Support vectors	1,075	802	445	**377**

Best results are bold faced

The subsequent test was focused on the effective usefulness of the embedding. Using the combined feature set, we compared the performance of the system with and without the embedding procedure. Table 2 shows that by using an embedding approach with only 1,000 positive samples and 1,000 negative samples we can obtain similar performances to those obtained by using ten times more samples. This is a great advantage because it implies that, given a new manuscript to be analyzed, the human operator has to manually annotate only a few pages. This procedure can be also included into a relevance feedback context: using a limited amount of correction on the results proposed with a standardly trained system, in a small amount of time good results can be easily achieved.

We would like to highlight that these results are not postprocessed, and that they were obtained using general assumptions on the appearance of blocks, without any prior inference. The set of features used reflects this approach. We used color features (RGB and HSV histograms) because generally decorations blocks have a

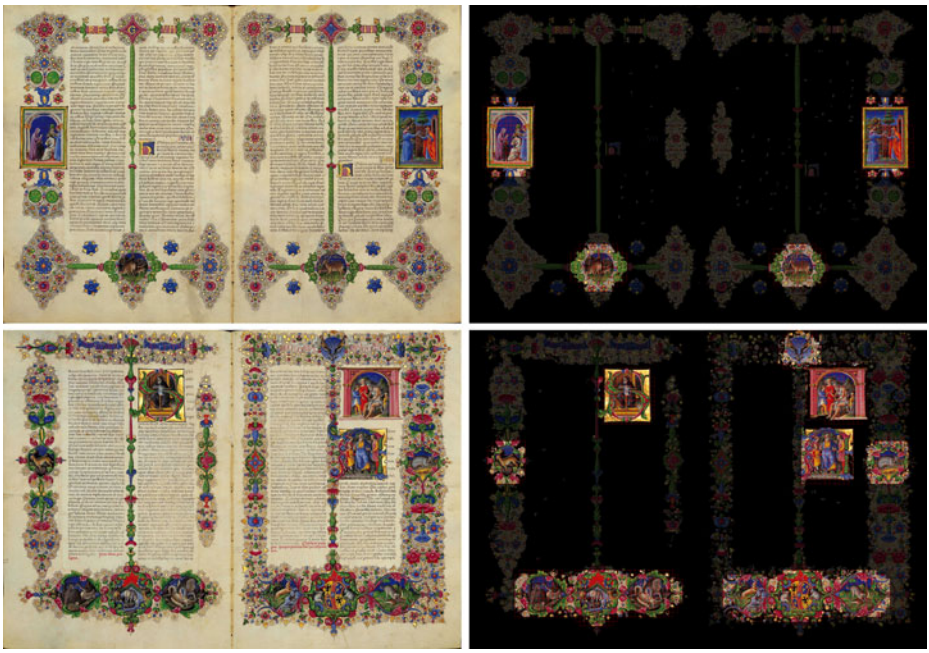


Fig. 8 Example of picture detection results

different (and quite limited) palette of colors with a lot of background, and we used texture features because generally decorations blocks are quite more regular and repetitive with a lot of symmetry. Some example results are shown in Fig. 8.

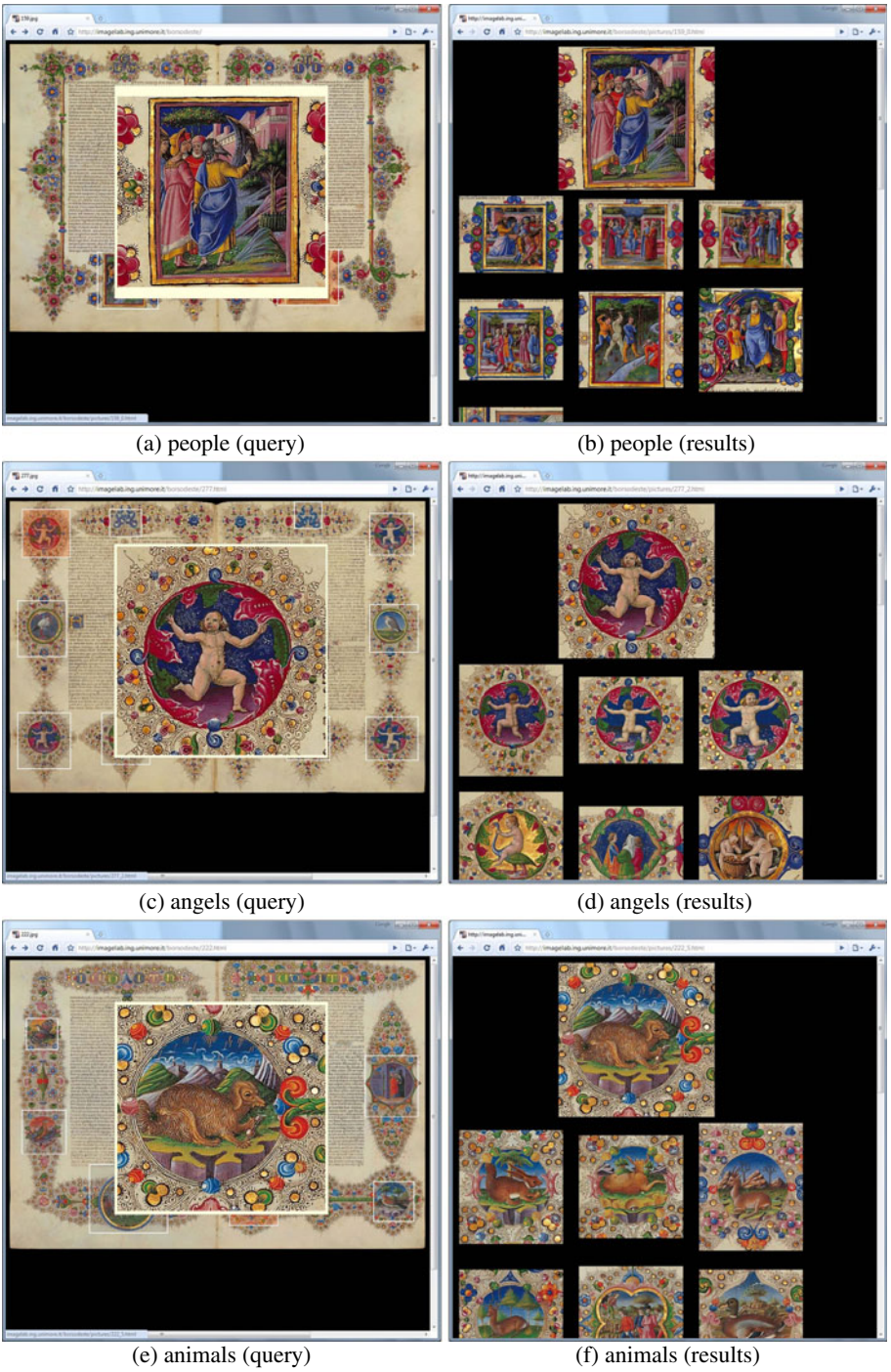
7 A glimpse to the future works

7.1 Cleaning out results

Since we analyzed Bible images by blocks, the extracted pictures contain a lot of noise. In fact, the segmentation we accomplished is the result of a classification procedure, so it is not very refined at boundary level. With a view of a future employment of these pictures for object extraction and recognition tasks (animals, symbols, people, scene, etc...), we tested some basic segmentation algorithms in order to find out boundaries of depicted objects. In Fig. 9 we show some segmentation results using the method proposed by Shih *et al.* for color images [33]. This method relies on an automatic seeded region growing starting from automatically selected initial seeds, with a final region-merging procedure to merge similar or small regions.

The proposed technique provided nice segmentation results, even because the chromatic range of these pictures is often limited. For this reason, this method is also very accurate for the separation between background and pictorial data. Nevertheless a deeper study on this topic should be necessary.

Fig. 9 Segmentation of extracted pictures by boundary detection



(a) people (query) (b) people (results)

(c) angels (query) (d) angels (results)

(e) animals (query) (f) animals (results)

Fig. 10 A content-based retrieval example of people, animals and angels. Navigating through the interface, the user can select the query image, and the system proposes the retrieval results ordered by descending appearance similarity

7.2 Content-based image retrieval and browsing

The final application devised for this work is the design of a system to present all the details of illuminated manuscripts (and in particular the Borso d'Este Holy Bible) in a user friendly way. Retrieval of images by content similarity will be improved with a joint use of both pictorial and textual information. An example of CBIR results is provided in Fig. 10.

The real implementation, aimed at a museum interactive system, should include an interface to enhance the user experience, providing new methodologies to interact with the work. A set of gestures for browsing, selection, zooming and manipulating contents will provide the user the possibility to experience a personal way to explore the book. We produced some initial application samples using popular web 2.0 applications or browser plugins.

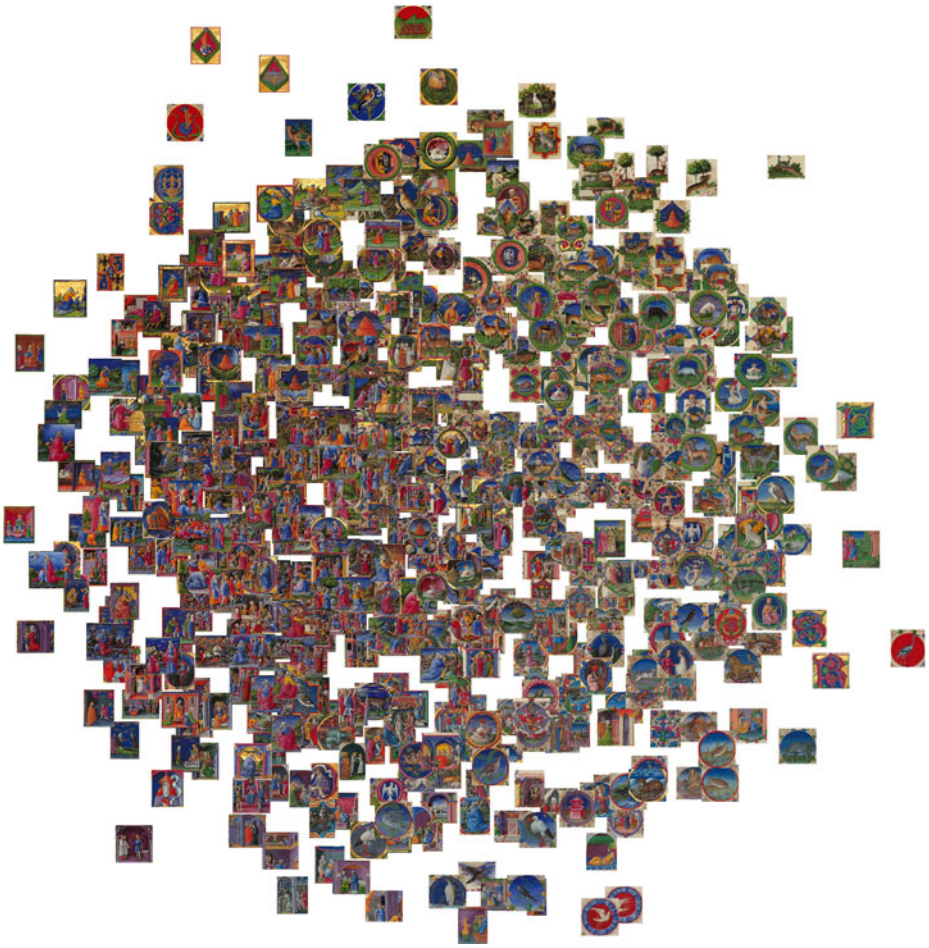


Fig. 11 The Sammon Mapping visualization of the similarity results

The presentation of miniature illustrations segmented from digitalized images of Bible's page is performed using a Sammon Mapping. It is a very well known approach with the power to visually enhance correlation between data, similarities and clusters. Given n images to present to the user, the goal of the Sammon mapping in this application is to find n points in a 2D space in such a way that the corresponding distances between each pair of points in this new two-dimensional space approximate the original ones as close as possible. The result is shown in Fig. 11. The user can browse through the images simply clicking on them on the map and for each image a set of information will be available to the user, for example explanations about the content of the image from a religious and historical point of view. It will also be possible to navigate to the relative page in the Bible: this freedom of exploration would let the user discover new interesting links across the work.

8 Conclusions

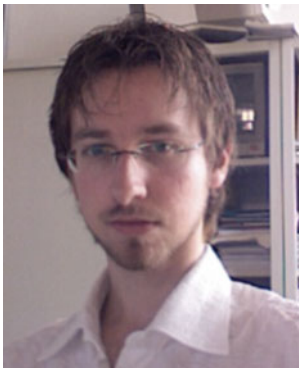
This paper describes a system for the automatic segmentation of text and decorations from illuminated manuscripts. Starting from the high resolution replicas of the Bible pages, text is detected and removed, a preprocessing stage focuses the analysis on the most valuable pixels of the image, then a sliding window analysis extracts low level color and texture features of each block. By the application of the described embedding procedure, SVM classification provides good results with less training samples and allows the use of RBF kernels. The goal of the final system is to provide an enhanced experience of high quality historical books.

References

1. Barbu A Learning real-time MRF inference for image denoising. In: Computer vision and pattern recognition
2. Bigun J, Bhattacharjee SK, Michel S (1996) In: Orientation radiograms for image retrieval: an alternative to segmentation, vol 3, pp 346–350
3. Bishop C (2006) Pattern recognition and machine learning. Springer
4. Chen N, Blostein D (2007) A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int J Doc Anal Recog 10:1–16
5. Crouse MS, Nowak RD, Baraniuk RG (1998) Wavelet-based statistical signal processing using hidden Markov models. IEEE Trans Signal Process 46:886–902
6. Diligenti M, Frasconi P, Gori M (2003) Hidden tree Markov models for document image classification. IEEE Trans Pattern Anal Mach Intell 25:519–523
7. Fataicha Y, Cheriet M, Nie J, et al (2002) Content analysis in document images: a scale Space approach. In: International conference on pattern recognition, vol 3. IEEE Computer Society, pp 335–338
8. Gill G (1981) Evaluation and inversion of the ratios of modified Bessel functions, $I_0(x)/I_1(x)$ and $I_{1.5}(x)/I_{0.5}(x)$. ACM Trans Math Softw 7:199–208
9. Grana C, Vezzani R, Cucchiara R (2007) Enhancing HSV histograms with achromatic points detection for video retrieval. In: International conference on image and video retrieval, pp 302–308
10. Grana C, Borghesani D, Cucchiara R (2008) Describing texture directions with Von Mises distributions. In: International conference on pattern recognition
11. Grana C, Borghesani D, Cucchiara R (2009) Fast block based connected components labeling. In: Proceedings of the IEEE international conference on image processing. Cairo, Egypt

12. Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. IEEE Trans Syst Man Cybern 3:610–621
13. Hjaltason G, Samet H (2003) Properties of Embedding Methods for Similarity Searching in Metric Spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 25:530–549.
14. Hu J, Kashi R, Wilfong R (1999) Document classification using layout analysis. In: International workshop on database and expert systems applications. IEEE Computer Society, pp 556–560
15. Jain A, Dubes R (1988) Algorithms for clustering data. Prentice-Hall, Inc
16. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: European conference on machine learning. Springer, pp 137–142
17. Joachims T (2002) Learning to classify text using support vector machines: methods, theory, and algorithms. Kluwer Academic Publishers/Springer
18. Journet N, Ramel J, Mullot R et al (2008) Document image characterization using a multiresolution analysis of the texture: application to old documents. Int J Doc Anal Recog 11:9–18
19. Kavallieratou E (2005) A binarization algorithm specialized on document images and photos. In: International conference on document analysis and recognition. IEEE Computer Society, pp 463–467
20. Kitamoto A, Onishi M, Ikezaki T, et al (2006) Digital bleaching and content extraction for the digital archive of rare books. In: International conference on document image analysis for libraries. IEEE Computer Society, pp 133–144
21. Kittler J, Illingworth J (1985) Relaxation labelling algorithms—a review. Image Vis Comput 3:206–216
22. Konidaris T, Gatos B, Ntzios K, et al (2007) Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. Int J Doc Anal Recog 9:167–177
23. Le Bourgeois F, Emptoz H (2007) DEBORA: Digital accEss to BOoks of the RenAissance. Int J Doc Anal Recog 9:193–221
24. Le Bourgeois F, Trinh E, Allier B, et al (2004) Document images analysis solutions for digital libraries. In: International conference on document image analysis for libraries. IEEE Computer Society, pp 2–24
25. Meng G, Zheng N, Song Y, et al (2007) Document images retrieval based on multiple features combination. In: International conference on document analysis and recognition, vol 1. IEEE Computer Society, pp 143–147
26. Nagy G (2000) Twenty years of document image analysis in PAMI. IEEE Trans Pattern Anal Mach Intell 22:38–62
27. Nicolas S, Dardenne J, Paquet T, et al (2007) Document image segmentation using a 2D conditional random field model. In: International conference on document analysis and recognition, vol 1, pp 407–411
28. Ogier J, Tombre K (2006) Madonne: document image analysis techniques for cultural heritage documents. In: Digital cultural heritage. Proceedings of 1st EVA conference, Oesterreichische Computer Gesellschaft, pp 107–114
29. Pekalska E, Duin RPW (2002) Dissimilarity representations allow for building good classifiers. Pattern Recogn Lett 23:943–956
30. Prati A, Calderara S, Cucchiara R (2008) Using circular statistics for trajectory analysis. In: International conference on image and video retrieval, pp 1–8
31. Ramel J, Busson S, Demonet M (2006) AGORA: the interactive document image analysis tool of the BVH project. In: International conference on document image analysis for libraries, pp 145–155
32. Sezgin M, Sankur B (2004) Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging 13:146–168
33. Shih FY, Cheng S (2005) Automatic seeded region growing for color image segmentation. Image Vis Comput 23:877–886

Costantino Grana received a PhD in Information Engineering in 2004 and is currently Assistant Professor at the University of Modena, Italy. He has worked in Medical Imaging on dermoscopic images for automatic melanoma identification, and in Video Surveillance. His present interests comprise multimedia information analysis, focusing on image and video concept detection, and historical document analysis.



Daniele Borghesani is currently pursuing the Ph.D. at the University of Modena and Reggio Emilia (Italy), where he took his master degree in Computer Science in 2006. During his master thesis and his research, he carried out different image processing and pattern recognition studies. In particular, his current research regards document analysis and content-based image retrieval, focused on Renaissance illuminated manuscripts.

Rita Cucchiara (1989, Laurea in Electronic Engineering and 1993, PhD in Computer Engineering at the University of Bologna, Italy) is full Professor at the University of Modena and Reggio Emilia where she heads the ImageLab laboratory (http://imagelab.ing.unimore.it). Her current interests include Pattern Recognition and Computer Vision for video surveillance and multimedia. She is a Fellow of IAPR.