Non-negative dictionary learning for paper watermark similarity

David Picard, Thomas Henn, Georg Dietz

Non-negative dictionary learning for paper watermark similarity

David Picard and Thomas Henn
ETIS - UMR 8051
ENSEA/Université de Cergy-Pontoise/CNRS
F-95000 Cergy-pontoise, France
Email: picard@ensea.fr, thomas@henn.fr

Georg Dietz
Papierstruktur.de
Dresden, Germany
Email: georg.dietz@papierstruktur.de

***Abstract*—In this paper, we investigate the retrieval of paper watermark by visual similarity. We propose to perform the visual similarity by encoding small regions of the watermark using a non-negative dictionary optimized on a large collection of watermarks. The local codes are then aggregated into a single vector representing the whole watermark. Experiments are carried out on a test of tracings (manual binarization of watermarks).**

I. Introduction

Hand-made papers with paper watermarks were produced in paper-mills by using a wire mesh. From the thirteenth century on nearly all European paper-makers signed their paper production by watermarks. These watermarks were made by wire figures, which were fixed on the wire mesh. During the paper making process the molds were dipped into a liquid pulp. When lifting them out, the liquid drained out through the wire mesh. The fibers became interwoven to a sheet of paper. The pattern of the mesh with the attached watermark wire-figure became as imprinted on the paper.

Paper watermarks in ancient paper are significant clues to historical investigations and can play a major role in the authentication, dating and attribution of art to a certain artist. Furthermore, watermarks and paper structures can provide information about the local usage of a paper (for example, shed light on the traveling of a writer) It can allow the detection of falsification. It has become increasingly important to be able to identify the watermarks so as to retrieve the date and location of the paper producer. Thanks to growing collections of paper watermarks, similar watermarks are more likely to be found. The largest aggregator in this field is the Bernsteinportal (`www.memoryofpaper.eu`). This database consists of more than 227.000 watermarks of 28 different watermark collections. All watermarks of these collections are described by verbal terms in different languages. Such a verbal system has limits due to the many categories used and many more subcategories. There are many ways to describe watermarks, and after a wrong description a watermark might never again be detected. Furthermore, manual browsing of such collections is impossible due to the sheer amount of available data.

In this paper, we investigate a query by example system where a user submits a new watermark and retrieves the most visually similar watermarks in the collection. In order to perform this query by example scheme, we propose to represent watermarks by specially crafted vectors. Our representation aims at matching similar small regions of the watermark. To perform that, we extract many regions from the watermark and compute features on these regions. Inside a region, we consider watermarks to be the union of basic patterns. We thus propose to encode the region as a non-negative combination of positive patterns taken from a dictionary learned on a set of watermarks. We finally aggregate the codes into a single vector such that the dot product emulates matching the regions.

The remaining of this paper is as follows: First we discuss the encoding procedure for the regions. Then, we explain the code matching procedure, before we detail the aggregation of all codes into a single vector. Finally, we present experiments on a set of tracings (manual binarization of watermarks) before we conclude.

II. Sparse non-negative dictionary learning

We consider the image as a collection of regions spanned by a sliding window. Each window is then encoded on a dictionary and the codes are used as local features. Similar regions should be encoded by similar atoms of the dictionary, allowing us to use the codes as a proxy for comparing regions.

We propose to learn the dictionary from a large quantity of images using a reconstruction based criterion, to which we add several constraints to reflect the high specificity of our input images. Let $\mathbf{D}$ be the dictionary, $\{\mathbf{r}\}$ the input regions of all available images and $\mathbf{x_r}$ their corresponding codes, we consider the following optimization problem:

$$\min_{\mathbf{D},\alpha} . \quad \frac{1}{2}\sum_{\mathbf{r}\in\{\mathbf{r}\}} \|\mathbf{r}-\mathbf{D}\mathbf{x_r}\|^2 \tag{1}$$

$$s.t. \quad \forall i, \mathbf{D}_i \geq 0, \|\mathbf{D}_i\|_2^2 = 1 \tag{2}$$

$$\forall \mathbf{r}, \mathbf{x_r} \geq 0, \|\mathbf{x_r}\|_1 \leq k \tag{3}$$

In this problem, $\mathbf{D}\alpha_{\mathbf{x}}$ is the reconstruction of $\mathbf{r}$ using $\mathbf{D}$, which we want as close as possible to the original sample. The first constraint on $\mathbf{D}$ states that all atoms of the dictionary have positive components, which means that atoms can only be constructive and not destructive. In our model, it makes sense

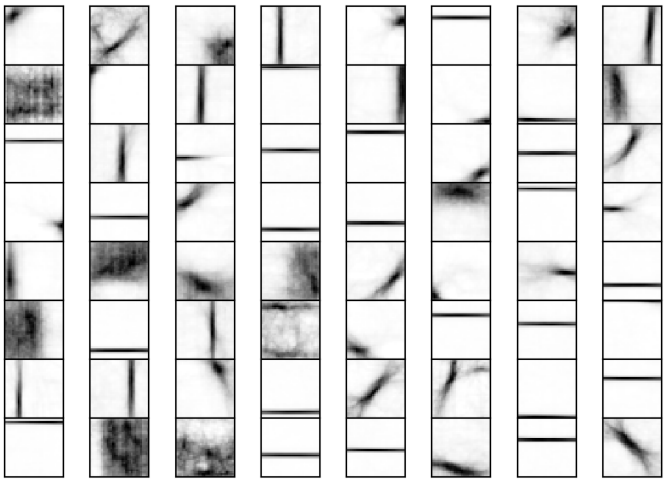

Fig. 1. Example of learned dictionary with 64 atoms of size 32×32 pixels.

to forbid destructive atoms since the watermark pattern are a superposition of strokes. The norm constraint on $\mathbf{D}$ ensures that no atoms diverges to infinite components. The positivity constraint on $\mathbf{x_r}$ follows the same principle as for $\mathbf{D}$: In our model, it makes no sense to use an atom to remove a part of the pattern in the region. Finally, the ℓ_1 norm constraint on $\mathbf{x_r}$ enforces sparsity in the codes, which means that only a few atoms are used to reconstruct the original pattern. The benefits of this sparsity constraint are twofold: First it makes the codes more robust to noise by removing small coefficients, and second it tends to favor more unique decompositions in the case of over-complete dictionaries, which is essential when using the code to compute similarities.

To train the dictionary, we follow a stochastic gradient descent as proposed in [1]. We used mini-batches consisting of all non-empty regions spanned by a sliding window at 5 different scales for each image. To compute the codes of all these region, we used the Frank-Wolfe algorithm [2] which proved to be very efficient for the considered constraints. An example of a learned dictionary at convergence is shown in Figure 1.

III. Local features matching

For a given image, we can now compute a set of codes corresponding to the non-empty regions spanned by the sliding window. In order to compare two images, we could simply count the number of matching codes using the sum of similarities between codes:

$$s_I(I_r, I_s) = \sum_{\mathbf{x}_r \in I_r} \sum_{\mathbf{x}_s \in I_s} s_{\mathbf{x}}(\mathbf{x}_r, \mathbf{x}_s) \tag{4}$$

The similarity between two codes can be defined in different ways, but a popular choice in computer vision tasks [3] is to count 1 if $\mathbf{x}_s$ is the nearest neighbor of $\mathbf{x}_r$ in I_s and the second nearest neighbor is a much less similar code (*i.e.*, $d(\mathbf{x}_r, \mathbf{x}_s) < \eta d(\mathbf{x}_r, \mathbf{x}_2)$, with $\mathbf{x}_2$ the second nearest neighbor of $\mathbf{x}_r$ in I_s and η a contrast parameter, typically 0.6).

To further improve the similarity between images, we can discard proposed matches that do not preserve the global geometry of the image. Indeed, if the tracings are very similar, an affine transform should be able to map the location of all matching regions in I_r to the location of the corresponding regions in I_s. In computer vision tasks, such transform is usually estimated using a consensus algorithm like RANSAC [4]. An example of matching regions filtered by RANSAC is shown on Figure 2. Without the geometric filtering, all regions crossing a chainline would match similar region in the second image. However, these matches have no global geometric coherence and are easily removed, leaving only those that are coherent with the transform of more complex patterns (for example, on top of Figure 2).

However, this involves a number of comparison quadratic with the number of regions in the images, which quickly becomes intractable. In our case, since we used 5 different scales of the input image, we obtain about 4500 codes per image, which makes the computation of the similarity between two images in the order of one second, which then has to be multiplied by the number of images in the collection. The use of local feature matching is thus not possible at search time.

IV. Local features aggregation

To overcome the computational time of pairwise matching of local features, we propose to use aggregation schemes that reduce the set of codes to a single representation.

We use approaches based on advanced Bag of Word models very popular in the object recognition community. In particular, we focus on the tensor framework developed in [5], [6] which finds an embedding of the set of local features so as to approximate the matching function.

The main idea is to provide an alternative pairwise matching function, as follows:

$$s(I_r, I_s) = \sum_{\mathbf{x}_r \in I_r} \sum_{\mathbf{x}_s \in I_s} s_q(\mathbf{x}_r, \mathbf{x}_s) e^{-\gamma \|\mathbf{x}_r - \mathbf{x}_s\|^2} \tag{5}$$

With s_q defined using a quantized version of $\mathbf{x}$ over a codebook $\{\mu_c\}$:

$$s_q(\mathbf{x}_r, \mathbf{x}_s) = \langle q(\mathbf{x}_r), q(\mathbf{x}_s) \rangle = \sum_c q_c(\mathbf{x}_r) q_c(\mathbf{x}_s), \tag{6}$$

$$q_c(\mathbf{x}) = \begin{cases} 1 \text{ if } c = \arg\min_m \|\mathbf{x} - \mu_m\| \\ 0 \text{ else} \end{cases}$$

Then, this matching function is approximated using a Taylor expansion of the Gaussian kernel:

$$\begin{aligned} s(I_r, I_s) &= \sum_{\mathbf{x}_r \in I_r} \sum_{\mathbf{x}_s \in I_s} \langle q(\mathbf{x}_r), q(\mathbf{x}_s) \rangle \sum_{n=0}^{\infty} A_n \langle \mathbf{x}_r, \mathbf{x}_s \rangle^n \qquad (7) \\ &= \sum_{n=0}^{\infty} A_n \langle \sum_{\mathbf{x}_r \in I_r} q(\mathbf{x}_r) \otimes \mathbf{x}_r^{\otimes n}, \sum_{\mathbf{x}_s \in I_s} q(\mathbf{x}_s) \otimes \mathbf{x}_s^{\otimes n} \rangle \end{aligned}$$

Where $\otimes$ denote the tensor (outer) product between vectors.

By setting different stop to the Taylor expansion, we can recover several methods from the literature, namely Vector of

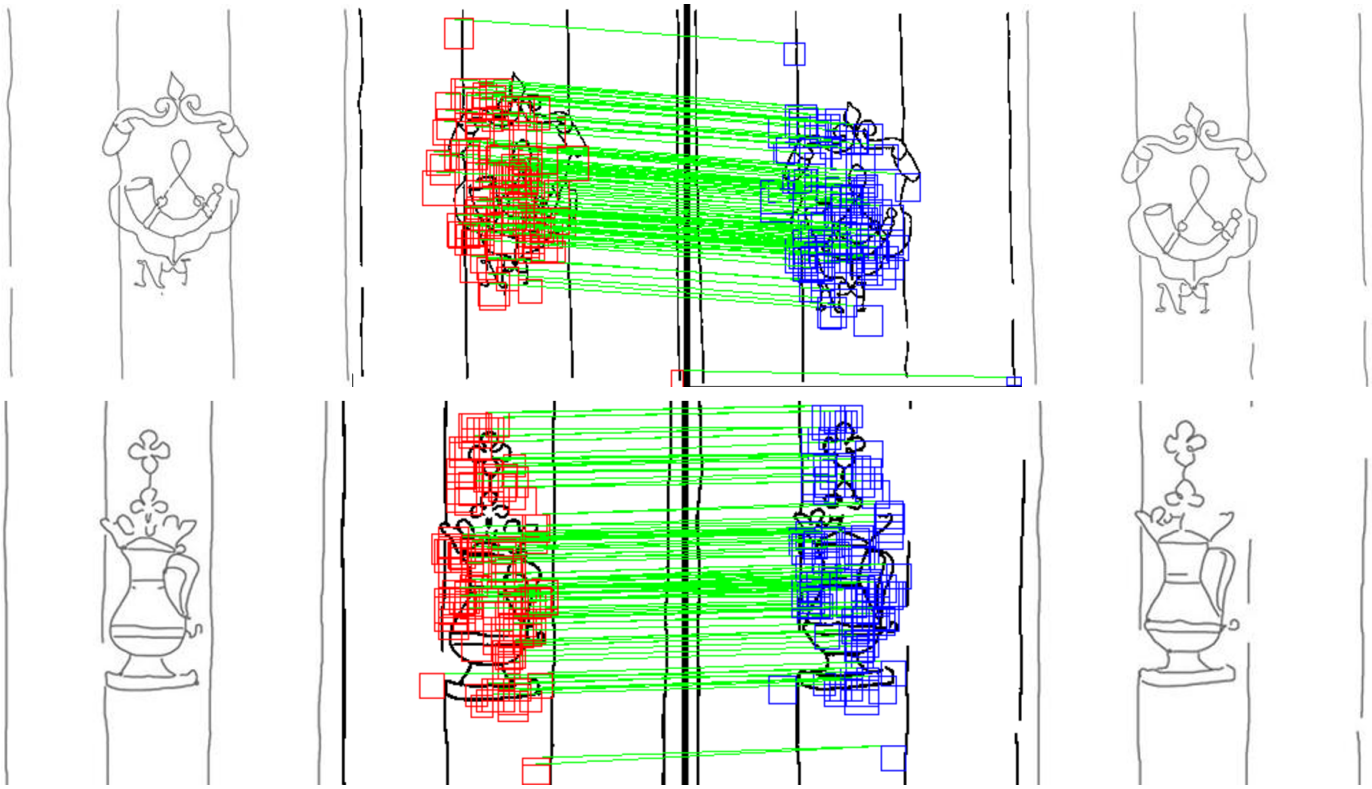

Fig. 2. Examples of matching regions after applying outliers removal following the RANSAC estimation of the affine transform between the pairs of regions. First and last column correspond to the original images, while column 2 and 3 correspond to the processed tracings with the matching regions highlighted. Most of the matching regions are centered on the tracings, which is a desirable property in the case of similarity search. Note that regions of different scales can be matched.

Locally Aggregated Descriptors (VLAD) [7] for the first order, and Vector of Locally aggregated Tensors (VLAT) [5] for the second order.

The main advantage of such methods, is that all local features are processed ahead of time to produce the $n+1$ order tensor. At search time, only the dot product (inner product) between the obtained tensors is required to compute the similarity. However, high order tensors are difficult to use because of their increased size.

To further improve the results, we use a principal component analysis inside each cluster $\{\mu_c\}$ followed by ℓ_2 normalization for each code [8]. This is known to obtain better results [9]. The final representation are then processed by a component-wise signed square root followed by a global ℓ_2 norm, which is a standard procedure when using such methods.

V. Experimental results

The dataset on which we performed experiments consists in 658 tracings of which about 10% are labeled. Examples of such tracings are shown in Figure 3. The labels consist in groups of tracings that should be the first to be retrieve when taking one image of the group as query. The performances are measured as the average rank of the first relevant result (*i.e.*, given a query tracing, we sort all other images by decreasing similarity and return the index of the first relevant image). A random sorting of the 657 remaining images would lead to a mean average rank of about 168.

The results for different parameters are shown in Table I. We varied the order of the Taylor approximation and the step size at which regions are extracted. Codes are computed using the same dictionary trained on regions extracted from the whole dataset at 5 different scales. Only the clusters are recomputed for each setup of different step size and approximation order.

Our best result is an average rank of 30, which is significantly better than the random sorting (average rank of 168). As we can see, using a smaller step size of 8 instead of 16 pixels leads to better performances. This can be explained by the increase of the number of selected regions which automatically increases the number of correct matches for complex patterns (*i.e.*, not translation invariant like chainlines). Similarly, using a second order expansion significantly improves the results. This can be explained by the added precision which tends to also favor complex patterns over simpler ones like the chainlines.

VI. Conclusion

In this paper, we proposed a method for the retrieval of paper watermarks based on sparse non-negative matrix factorization. We train a dictionary on non-negative patterns on many images using a stochastic gradient descent algorithm. Using a sliding window, we compute non-negative sparse

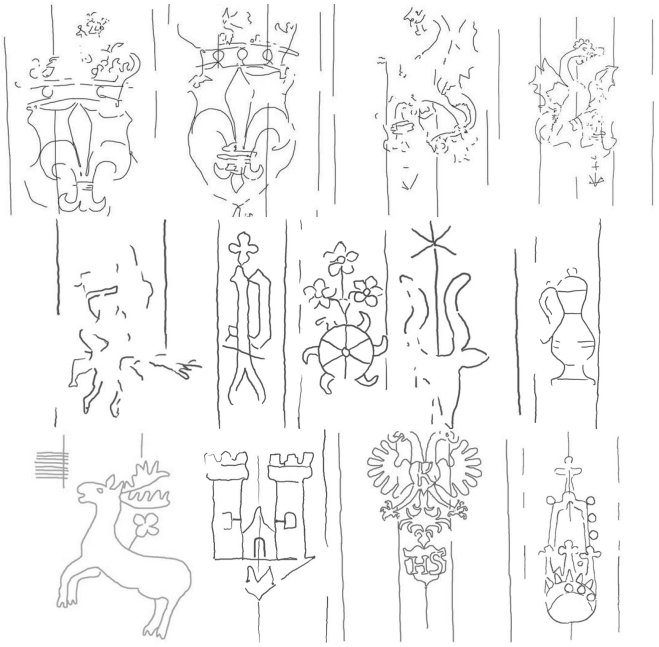

Fig. 3. Examples of paper watermark tracings after preprocessing, showing the diversity of encountered patterns.

dictionary	support	step	clusters	order	average 1st
64	32	8	64	1	39.8
64	32	8	64	2	**30.6**
64	32	16	64	1	61.7
64	32	16	64	2	36.5

TABLE I
1ST RELEVANT IMAGE AVERAGE RANK FOR VARYING PARAMETERS.

codes on many regions of the image based on the dictionary. These codes are then aggregated in a single image representation using either first order approximation (VLAD) or second order approximation (VLAT). We perform experiments on a dataset of watermarks that show our method is able to retrieve relevant watermarks with an average rank of the first relevant result of about 30/657. Further investigations involve considering invariance to specific transforms (scale, flip, rotation) when optimizing the dictionary, as proposed in [10]. Also, the aggregation scheme could be improved by preserving the spatial layout of the codes as proposed in [11].

REFERENCES

[1] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[2] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956. [Online]. Available: http://dx.doi.org/10.1002/nav.3800030109

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 20, 2003, pp. 91–110.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981. [Online]. Available: http://dx.doi.org/10.1145/358669.358692

[5] D. Picard and P.-H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. pages–669.

[6] ——, "Efficient image signatures and similarities using tensor products of local descriptors," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 680–687, 2013.

[7] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[8] R. Negrel, D. Picard, and P.-H. Gosselin, "Web-scale image retrieval using compact tensor aggregation of visual descriptors," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 24–33, 2013.

[9] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the vlad image representation," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 653–656.

[10] T. Guthier, V. Willert, and J. Eggert, "Topological sparse learning of dynamic form patterns," *Neural computation*, vol. 27, no. 1, pp. 42–73, 2015.

[11] D. Picard, "Preserving local spatial information in image similarity using tensor aggregation of local features," in *2016 IEEE International Conference on Image Processing (ICIP)*, Unknown, Unknown or Invalid Region, 2016, pp. 201–205. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01359109