

Hierarchical Indexing using R-trees for Replica Detection

Yannick Maret, David Marimón, Frédéric Dufaux and Touradj Ebrahimi

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Institut de Traitement des Signaux
CH-1015 Lausanne, Switzerland

ABSTRACT

Replica detection is a prerequisite for the discovery of copyright infringement and detection of illicit content. For this purpose, content-based systems can be an efficient alternative to watermarking. Rather than imperceptibly embedding a signal, content-based systems rely on content similarity concepts. Certain content-based systems use adaptive classifiers to detect replicas. In such systems, a suspected content is tested against every original, which can become computationally prohibitive as the number of original contents grows. In this paper, we propose an image detection approach which hierarchically estimates the partition of the image space where the replicas (of an original) lie by means of R-trees. Experimental results show that the proposed system achieves high performance. For instance, a fraction of 0.99975 of the test images are filtered by the system when the test images are unrelated to any of the originals while only a fraction of 0.02 of the test images are rejected when the test image is a replica of one of the originals.

Keywords: Copyright protection, Image Replica Detection, R-trees, Hierarchical Indexing

1. INTRODUCTION

In this paper, we propose a system to detect image replicas. By replica, we refer not only to a bit exact copy of a given reference image, but also to modified versions of the image after minor manipulations, malicious or not, as long as these manipulations do not change the perceptual meaning of the image content. In particular, replicas include all variants of the reference image obtained after common image processing manipulations such as compression, filtering, and adjustments of contrast, saturation or colours. Such a detection system can be applied to *detect copyright infringement* by identifying variations of a given copyrighted image. Another application is to *discover known illicit content* such as child pornography images known to the police.

The problem of image replica detection is a particular subset of the more general problem of content-based search and retrieval of multimedia content. In recent years, multimedia search and retrieval has been the subject of extensive research works and standardisation activities (MPEG-7^{1,2} and more recently JPSearch³). However, the specific problem of image replica detection has so far been the focus of fewer research efforts.

Two existing approaches to detect image replicas are *watermarking*⁴ and *robust fingerprinting*.⁵⁻⁷ Watermarking techniques⁴ consist in embedding a signature in the reference image before dissemination. Replicas of the reference image can subsequently be detected by verifying the presence of the watermark. This class of techniques typically achieves high efficiency for the correct classification of replicas and non-replicas. However, it requires to modify the reference image, namely to embed a signature, prior to its distribution. Unfortunately, this is not always possible. For instance, the method is not applicable to already disseminated copyrighted content nor in the case of illicit content. Robust fingerprinting techniques⁵⁻⁷ analyse the reference image in order to extract a signature associated with the image content. Replicas are then identified whenever their signatures are close to that of the reference. This class of techniques is often based on a single feature, for example characteristic points of the Radon transform,⁵ log-mapping of the Radon transform,⁶ or intra-scale variances of wavelet coefficients.⁷ While it is usually robust, computationally efficient, and suitable for fast database indexing and retrieval, it often performs poorly for the accurate classification of replicas and non-replicas.

Send correspondence to yannick.maret@epfl.ch or frederic.dufaux@epfl.ch

More recently, techniques for image replica detection have been described in.^{8,9} Ke *et al*⁸ propose a method based on the extraction of features, referred to as Key Points (KPs), which are stable in a scale-space representation. An image is typically represented by thousands of KPs. Test images are then classified as replicas or non-replicas using local sensitive hashing to match their KPs to those of the reference image. More specifically, no distance is directly computed, but it is rather the number of matching KPs which quantifies the similarity between two images. While this approach achieves very good performance for replica detection, it requires a computationally complex features extraction step. Qamra *et al*⁹ propose a different method based on the computation of a Perceptual Distance Function (DPF). More precisely, a DPF is generated for each pair of reference and unknown images, to measure the similarity between the two. The main idea of the approach is to activate different features for different image pairs. Hence, only the most similar features are taken into account to compute the distance. While this method achieves good performance, it is inferior to.⁸

In this paper, we introduce a new approach for image replica detection based on our earlier works.²³ The main idea behind our approach is to estimate, and efficiently describe, the partition of the image space that contains the replicas of a particular original image. The system is then able to detect whether a test image is a replicas of one of the original images, or an unrelated image, by simply asserting whether the test image lies inside or outside the partition. The main difference with our previous work²³ is the finer estimation of the replica partition and the added hierarchical retrieval procedure. Note that the partition estimation is relatively coarse, and the system performance can be improved by making use of adaptive classifiers¹² on the potential originals given by the proposed system. However such improvement is not within the scope of this paper.

Simulation results show the effectiveness of the proposed system. For instance, for an average false negative rate of 2%, one achieves a fixed false positive rate of $2.5 \cdot 10^{-4}$. Indeed, the proposed technique significantly outperforms Perceptual Distance Function (DPF),⁹ even if we use fewer features (133 in this work against 268 in DPF). Although the achieved performance is not as good when compared to KPs,⁸ a speed up in terms of computational complexity, in the range of one to two orders of magnitudes, is achieved.

The paper is structured as follows. We describe and motivate the various algorithmic steps in Sec. 2. Experimental results are reported in Sec. 3. Finally, conclusions are drawn in Sec. 4.

2. METHOD

The main idea behind the proposed approach is to (approximately) determine the partition of the image space in which the replicas of a particular image lie. Replicas can then be easily detected by asserting whether a test image lies inside or outside the replica partition. For example, the replica partition determined by all the resized versions (say by a factor going from 0.1 to 5) of an image is, under certain assumptions*, a curve embedded in the image space. The curve starts at factor 0.1, going through the original, and finishing at factor 5. Consequently, transformations of the original by other operations result in as many curves going through the original, and the replicas of a particular image lie in the partition defined by the union of these curves. Finally, replicas of the replicas can also be considered to be replicas of the original, in which case, other curves going through each replica are also included. A possible simplification is then to consider that replicas result from a single function of many variables. One of the variable is the original and the other variables are parameters that control the replica aspect (for example, a resizing factor). In this simplified case, the replica partition is defined by a bounded smooth surface (or manifold) embedded in the image space.

In this paper, we propose to estimate the replica partition of a particular image by generating synthetic replicas from several single-parameterised functions. In other words, the generated replicas correspond to curves going through the original. Then, the replica partition is estimated to be the union of simple volumes (such as boxes or hyper-spheres) that are constructed around each replica example, and images falling within one of these are considered as replicas of the original. The attractiveness of the proposed approach lies in the existence of efficient algorithms to index simple volumes and retrieve them. Consequently, effective methods exist for determining the indexed volumes that contain a given point or intersect with a given simple volume. Not only is this an efficient manner to describe the estimated replica partition, but it also permits to easily determine if a test image falls inside or outside the partition.

*The images are considered to be smooth bi-dimensional signals, and the transformation functions are derivable.

Table 1. *Features overview.* List the types of used features and the number of extracted statistics.

name	# features
Gabor, squared coeff. mean	20
Gabor, squared coeff. std dev.	20
Colour, histogram	10
Colour, channel mean	24
Colour, channel std dev.	24
Colour, spatial distribution	20
Grey-level, histogram	3
Grey-level, bin mean	3
Grey-level, bin std dev.	3
Grey-level, spatial distribution	6
total	133

The rest of this section is organised as follows. We first present the prerequisites of the partitioning and retrieval algorithms. More precisely, Sec. 2.1 details the image representation used throughout this paper, and the training replica examples are introduced in Sec. 2.2. Finally, Sec. 2.3 describes the algorithms that create the replica partitions and retrieve the potential originals of a test image.

2.1. Image Analysis

We propose to analyse images at different granularity levels l . At each granularity level, the image is subdivided into patches of the same size. For instance, at the coarsest granularity level there is one patch of the size of the image, at the next level there are 4 patches, then 9 patches and so on. Figure 1 shows these patches for the three first granularity levels. Each patch is then described by the features detailed in Sec. 2.1.1. Finally, the feature vectors are processed and their dimensionality reduced as explained in Sec. 2.1.2.

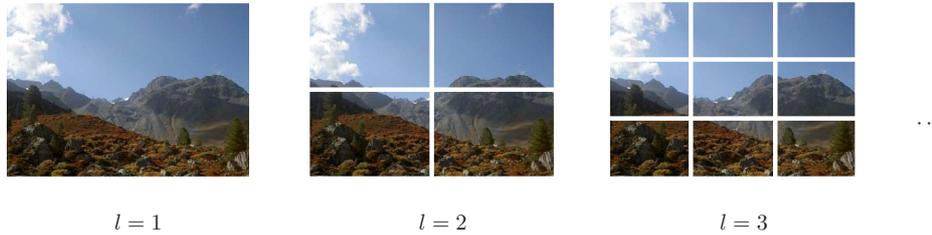


Figure 1. *Image Patches.* This figure represents the image patches at different granularity levels l .

2.1.1. Features Extraction

In order to compare the similarity between two images, visual features are extracted. The goal of feature extraction is twofold. First, it maps images onto a common space where they can be more easily compared. Second, it reduces the space dimensionality by keeping only the relevant information. Many visual features can be envisioned: colour, texture, shape, etc.¹¹ The choice of features generally depends on the image type. In the case of the image replica detection problem, it also depends on the type of replicas that are to be detected. For instance, if rotated images are considered, it would make sense to choose at least one feature that is mostly rotation invariant.

The features used in this work are of three types: texture, colour and grey-level statistics. They are the same as those used in our previous works¹² but are described thereafter for the sake of completeness. As shown in Table 1, we extract 133 features which are detailed in the following subsections.

Texture Features The texture features are composed of the first and second order statistics of each sub-band of the Gabor transform. The latter is performed as in¹³ on the equalised Illumination. More precisely, the used parameters are 0.75 for the upper centre frequency, 0.05 for the lower centre frequency, five scales and six orientations. For more details about these parameters, refer to.¹³ Mean and standard deviation estimates of the squared coefficients are computed for each of the 30 sub-bands.

Image rotation results in a circular shift of the coefficients obtained at a given scale. To provide additional robustness, a transformation invariant to circular shifts is used. More precisely, the estimates for different orientations but for the same scale are grouped in a vector and Fourier transform is applied to it. Finally, the absolute value is applied to each entry of the resulting vector. Since the mean and standard deviation estimates are real valued, the absolute value of the Fourier transform is symmetric, and some of the coefficients can be dropped. This results in a total of 20 mean and 20 variance estimates.

Colour Features The colour features are based on the HSI (Hue Saturation and Intensity) colour representation. Each pixel in the image is classified into one of ten *colour classes* depending on its position in this space. The classes are the achromatic colours ($S = 0$) *black*, *grey* and *white*, and the chromatic colours ($S > 0$) *red*, *orange*, *yellow*, *green*, *cyan*, *blue* and *purple*. The Illumination is used to classify a pixel into one of the three achromatic classes. The Hue is used to classify a pixel in one of the seven chromatic classes.

Colour Classes Histogram. A histogram is computed, giving the proportion of each colour class in the image. It is normalised such that it sums to one, and comprises 10 values.

Channel Statistics. Mean and variance estimates of the equalised Intensity channel are computed for each colour class. Mean and variance estimates of Saturation and logarithmic Hue channels are also computed for each chromatic colour class. This results in a total of 24 mean and 24 variance estimates.

Spatial Distribution Shape. The shape of the spatial distribution of each colour class is computed. This is achieved by computing two shapes characteristics for each colour class: spreadness and elongation.^{14,15} The first characteristic measures the compactness of the spatial distribution of a colour class. The second one gives the ratio between the shape length and width. Note that even if pixels assigned to a colour form totally disconnected components, this feature still captures useful information (namely the spatial distribution of these components). This results in 10 spreadness and 10 elongation measures.

Grey-Level Features The grey-level features are based on the equalised Intensity channel of the HSI model. The dynamic range of the image is linearly partitioned into eight bins corresponding to as many classes. Each pixel of the image falls into one of these bins. The use of grey-level feature is important because the colour features can be unsuited in some cases. For instance, it can happen when the reference or the test images are grey-level, or when conversion to grey-level is one of the considered operations in the replica detection system.

Grey-level Classes Histogram. A grey-level classes histogram is computed, giving the proportion of three intensity ranges in the image. It is normalised such that it sums to one, and comprises 3 values.

Grey-level Statistics. Mean and variance estimates are computed for pixels falling into each bin. This results in a total of 3 mean and 3 variance estimates.

Spatial Distribution Shape. Similarly to colour, the shape of the spatial distribution of each grey-level class is computed. This results in 3 spreadness and 3 elongation measures.

2.1.2. Features Processing

The visual features extracted from each patch are then processed in two steps. In a first steps the feature are normalised and set within the interval $[-1, +1]$. The number of dimensions is then reduced with the help of Principal Component Analysis (PCA). Both processing steps make use of training examples, which are provided by 350 images unrelated to those used in Sec. 3. Note that the feature processing constants are trained for each granularity level.

Table 2. *Training replicas generation.* Image operations and their parameters.

Operations	Parameters
Red, Green, Blue channels colourising	-11% to +11% by steps of 2%
Contrast change	positive and negative enhancements
Downsampling (without antialiasing filtering)	from 5% to 95% by steps of 10%
Scaling (with antialiasing filtering)	from 5% to 95% by steps of 10%
Colour Depth reduction	NA
Framing	width of 6% or 14% of the image size
Rotation	85, 95, 175, 185, 265, 275 degrees
Cropping	by 5% to 35% by steps of 10%
Saturation Modulation	from -22% to +22% with steps of 4%
Value Modulation	from -22% to +22% with steps of 4%

Statistical Normalisation The extracted feature are normalised using a statistical normalisation method.¹⁶ More precisely, let μ_α and σ_α be the mean and standard deviation estimates of the α -th feature over a subset of the training set (training examples for which any feature is an extremum over the training set are consider to be outliers and are ignored). The normalised feature is then given by $(\mathbf{f}_\alpha - \mu_\alpha)/(k \cdot \sigma_\alpha)$, where \mathbf{f}_α is the value of α -th feature. By Tchebychev’s theorem, at least a fraction $1 - 1/k^2$ of the normalised features are within the interval $[-1, +1]$. In the following k is set to 10 so that more than 99% of the features are within $[-1, +1]$. The features outside this interval are clipped to -1 or $+1$.

Dimensionality Reduction Many features are needed in order to have enough information to discriminate between replicas and non-replicas. Nonetheless, 133 features are too many for building an efficient indexing structure. For this reason, the dimensionality of the feature vector is reduced to d by making use of PCA. PCA reduces the dimensionality of the feature space by finding the d directions along which the scatter of the cloud of points is maximised.¹⁷ The PCA algorithm is applied to the training set, and results in a dimensionality reduction matrix \mathbf{W} . Then the reduced features are given by $\mathbf{f} = \mathbf{W} \cdot \mathbf{n}$ where \mathbf{n} is the normalised feature vector extracted from a patch.

2.2. Synthetic Training Examples

Examples of *replica images* can be generated artificially. Indeed, the reference image can be modified using different operations, resulting in several replicas. In this work, the replicas are generated by the operations listed in Table 2 resulting in a total of 77 examples. They are of the same types than those used to test the algorithms (see Sec. 3) but with different parameterisations, except for the non-parameterised operations (such as colour depth reduction). Note that the optimal choice of the training examples is still an open issue (not within the scope of this paper) and is the focus of future research.

2.3. Replica Partition Estimation using R-trees

In this section, we detail the method used to create a replica partition. The basic idea behind the proposed algorithm is to index, for each training example, a box (the simplest volume element). Each box partially describes the estimated replica partition while its entire estimation is given by their union. To index the boxes, we choose R-trees.¹⁸ Indeed, R-trees were originally created to index spatial objects using their Bounding Boxes (BBs). Therefore, an R-tree is constructed so as to efficiently answer the point-based query: *get records whose BBs include the point \mathbf{p}* , and the box-based query: *get records whose BBs intersect the box \mathbf{b}* .

The size of a box is chosen such that a fixed number of the considered training example Nearest Neighbours (NNs) are covered by it. The idea behind using the NNs is twofold. On the one hand, it creates an estimated replica partition composed of as few connected components as possible. Indeed, each box is connected to, at least, as many other boxes as the number of used NNs. On the other hand, since the number of replicas used for training is limited, it is necessary to ensure that novel replicas falls within one of the boxes with high probability. If the sampling of the replica examples generated by a single transformation is dense enough, it is likely that a

novel replica created by the same transformation falls ‘between’ two of the generated replica examples. Thus, the boxes generated around these two replica examples are likely to include the novel replica (assuming that they are part of each other NNs). Clearly, the replica partition defined by the union of these boxes is likely to encompass many of the potential replicas. Conversely, it is also important that unrelated images do not fall within the defined partition. This implies that the content (higher-dimensional volume) of the partition has to be somehow minimised. For this reason, the NNs are determined by making use of the content of the box delimited by each pair of examples rather than by the conventional Euclidian metric. This measure ensures that the determined boxes are those with the minimal contents (for the given algorithm and used parameters). By extension, the determined partition is also the one with the minimal content (again for the given algorithm and used parameters).

We next explain the potential behind the different granularity levels that are used for the image description (Sec. 2.1). Clearly, an image is composed of different regions, each having different characteristics (as visible in Fig. 1). Global features (features describing the image as a whole) give an ‘averaged’ version of the characteristics of every regions and perform well for replica detection.^{9,12} It is however possible for unrelated images to have very similar global features, in which case they will be considered to be replicas of each other. Image descriptions with granularity levels permit to lessen the number of such clashes. While unrelated images might have similar global features, it is less likely for the majority of their patches to have similar features. This is the main idea underlying the following hierarchical approach. A replica partition is constructed for each granularity level. Then, the potential originals of a test image are determined using only the global features granularity level. Every patches of the test image are subsequently tested on the finer granularity partition of each original found earlier, and the corresponding original is kept only if the number of matching patches is sufficient. Finally, the operation can be repeated for the remaining originals until reaching the finest granularity level (we experimented with up to $N_L = 3$).

The above observations lead us to devise the hierarchical indexing and retrieval algorithms presented in the following subsections.

Indexing Algorithm Algorithm 1 describes the constructions of the replica partition for a given original image. First synthetic replicas are generated (line 1.3), and features are extracted from them and from the original (line 1.5). At each granularity level l the image is described by $(l + 1)^2$ feature vectors (each describing one image patch). A replica partition is then estimated for each granularity level (lines 1.8 to 1.24). To achieve this, a box is created around each replica example (lines 1.16 to 1.24). First, the Nearest Neighbours of the replica example are determined, using as measure the content delimited by each pair of examples (line 1.16). Then, the extremal coordinates of the N_{NN} Nearest Neighbours are used to determine the box corners (lines 1.18 and 1.19); the tuning parameter δ permits to increase the box size. Finally, the box is indexed using the INSERT procedure from¹⁸ (line 24); the used key contains the original ID as well as a patch identifier. In the following, we use $N_{NN} = d$ in order to decrease the probability single training examples define corners of the bounding boxes. For example, if $N_{NN} = d - 1$ there is at least one corner defined by a single training example per bounding box.

Retrieval Algorithm The proposed retrieval algorithm works in a hierarchical way. It starts at level $l = 1$ and continues at finer granularity levels, possibly up to $l = N_L$. At a given granularity level, $(l + 1)^2$ feature vectors are extracted from the test image patches (line 2.5). A search box is then created around each feature vector and every indexed boxes intersecting with them are retrieved (line 2.9). The boxes retrieved at level $l = 1$ define a set of potential original \mathcal{O} (line 2.12). At finer granularity levels, this set is pruned by removing the originals with not enough matching patches (lines 2.17 to 2.20). The operation is repeated until the finest granularity level is reached.

3. RESULTS

3.1. Methodology

Test Images To simulate the performance of the proposed approach, we use an image collection⁸ that contains 18,785 photographs including (but not limited to) landscapes, animals, constructions, and people. The image

Algorithm 1 Estimates the replica subspace of an original and indexes it

Ensure: the R-trees $\{R_{tree}(l)\}_{l=1}^{N_L}$ contain the replica subspace estimation for the original

```
1: procedure ADDNEWORIGINAL( $\mathbf{I}_0, ID, \delta, N_{NN}$ )
2:   // generate  $N_R$  synthetic replicas (see Sec. 2.2):
3:    $\{\mathbf{I}_1 \dots \mathbf{I}_{N_R}\} = \text{GENERATEREPLICAS}(\mathbf{I}_0)$ 
4:   // extract features from the original image and the replica examples (see Sec. 2.1):
5:   // the  $d \times 1$  vector  $\mathbf{f}_{i,l,b}$  stands for features from image  $i$  at level  $l$  and patch  $b$ 
6:   for  $i = 0$  to  $N_R$  do
7:      $\{\mathbf{f}_{i,1,1} \mathbf{f}_{i,2,1} \mathbf{f}_{i,2,2} \dots \mathbf{f}_{i,N_L,(l+1)^{N_L}}\} = \text{ANALYSEIMAGE}(\mathbf{R}_i, [1 \dots N_L])$ 
8:     // treat each level separately
9:     for  $l = 1$  to  $N_L$  do
10:      // treat each patch separately
11:      for  $b = 1$  to  $(l+1)^2$  do
12:        // add a box per replica example
13:        for  $i = 0$  to  $N_R$  do
14:          // order the  $N_R + 1$  vectors  $\{\mathbf{f}_{j,l,b}\}_{j=0}^{N_R}$  according to  $v_j = (\mathbf{f}_{j,l,b} - \mathbf{f}_{i,l,b}) \bullet (\mathbf{f}_{j,l,b} - \mathbf{f}_{i,l,b})$ :
15:          //  $\sigma$  is a permutation of  $0, 1, \dots, N_R$  such that  $v_{\sigma(j)} \geq v_{\sigma(j-1)}$ 
16:           $\sigma = \text{ORDERBYCONTENT}(\{\mathbf{f}_{j,l,b}\}_{j=1}^{N_R}, \mathbf{f}_{i,l,b})$ 
17:          // compute the bounding box (defined by the two opposed corners  $\mathbf{f}_-$  and  $\mathbf{f}_+$ ):
18:           $\mathbf{f}_- = \left[ \min \{\mathbf{f}_{\sigma(j)}(\alpha)\}_{j=1}^{N_{NN}} \right]_{\alpha=1}^d$ 
19:           $\mathbf{f}_+ = \left[ \max \{\mathbf{f}_{\sigma(j)}(\alpha)\}_{j=1}^{N_{NN}} \right]_{\alpha=1}^d$ 
20:          // compute the side lengths  $\Delta(\alpha)$  of the bounding box :
21:           $\Delta = \mathbf{f}_+ - \mathbf{f}_-$ 
22:          // add to  $R_{tree}(l)$  the box having for opposed corners  $\mathbf{f}_{\pm} \pm \Delta \delta(l)$ :
23:          // assign the key  $(b, ID)$  to the record
24:           $\text{INSERT}(R_{tree}(l), \mathbf{f}_{\pm} \pm \Delta \cdot \delta(l), (b, ID))$ 
```

sizes and aspect ratios are variables, for example 900×600 , 678×435 , or 640×480 . They are mostly colour images, except for about one thousand images that are in grey-levels.

We first randomly select 500 images from the collection. Among the selected pictures, 150 are randomly chosen to be the original images, and the remaining are used to train the features preprocessing phase (Sec. 2.1.2). The 18,385 images remaining in the collection are used as negative test images. The replica test images are generated by the transforms listed below. These operations are the same than those used in.^{8,9} There are twelve categories, as shown thereafter. Moreover, an example for each is depicted in Fig. 2. The whole test set hence consists of 6000 replica images and 18,285 negative test images[†].

- *Colourising.* Tints the Red, Green, or Blue channel by 10%;
- *Changing contrast.* Increases or decreases the contrast using ImageMagick's default parameter;
- *Cropping.* Crops by 5%, 10%, 20%, or 30%;
- *Despeckling.* Applies ImageMagick's despeckling operation;
- *Downsampling.* Downsamples by 10%, 20%,30%,40%, 50%, 70%, or 90% (without antialiasing filtering);
- *Flipping.* Flips along the horizontal axis.
- *Colour Depth.* Reduces the colour palette to 256 colours;
- *Framing.* Adds an outer frame of 10% the image size. Four images are produced with different frame colour.
- *Rotating.* Rotates by 90° , 180° or 270° .
- *Scaling.* Scales up by 2, 4, 8 times, and down by 2, 4, 8 times (with antialiasing filter).

[†] Actually there are $18,285 + (150 - 1) \cdot 40$ negative test images since, for a given original, the replicas of other originals images corresponds to negative examples.

Algorithm 2 Finds the potential originals of a test image

Require: Originals to be indexed in the R-trees $\{R_{tree}(l)\}_{l=1}^{N_L}$ with Algorithm 1

```
1: procedure FINDPOTENTIALORIGINALS( $\mathbf{I}_T, \delta_s, \mathbf{m}_{min}$ )
2:   for  $l = 1$  to  $N_L$  do
3:     // extract features, at level  $l$ , from the test image:
4:     // the  $d \times 1$  vector  $\mathbf{f}_b$  stands for features from the test image  $\mathbf{I}_T$  at level  $l$  and patch  $b$ 
5:      $\{\mathbf{f}_1 \cdots \mathbf{f}_{(l+1)^2}\} = \text{ANALYSEIMAGE}(\mathbf{I}_T, l)$ 
6:     // get the set of keys  $\mathcal{K} = \{(b_i, ID_i)\}_i$  from  $R_{tree}(l)$ :
7:     // the keys in  $\mathcal{K}$  correspond to indexed boxes that intersect with the box defined
8:     // by the opposing corners  $\mathbf{f}_i - \delta_s$  and  $\mathbf{f}_i + \delta_s(l)$ 
9:      $\mathcal{R} = \text{SEARCH}(R_{tree}(l), \{\mathbf{f}_1 \cdots \mathbf{f}_{(l+1)^2}\} \pm \delta_s(l))$ 
10:    if  $l = 0$  then
11:      // determine the potential originals:
12:       $\mathcal{O} = \{ID \text{ such that } (ID, 1) \in \mathcal{R}\}$ 
13:    else
14:      // prune the potential originals:
15:      for  $ID \in \mathcal{O}$  do
16:        // determine the patches that match for this level and potential original:
17:         $\mathcal{P} = \{b \text{ such that } (ID, b) \in \mathcal{R}\}$ 
18:        if  $|\mathcal{P}| < \mathbf{m}_{min}(l)$  then
19:          // not enough patches match, remove this potential original:
20:           $\mathcal{O} = \mathcal{O} \setminus ID$ 
21:    return  $\mathcal{O}$ 
```

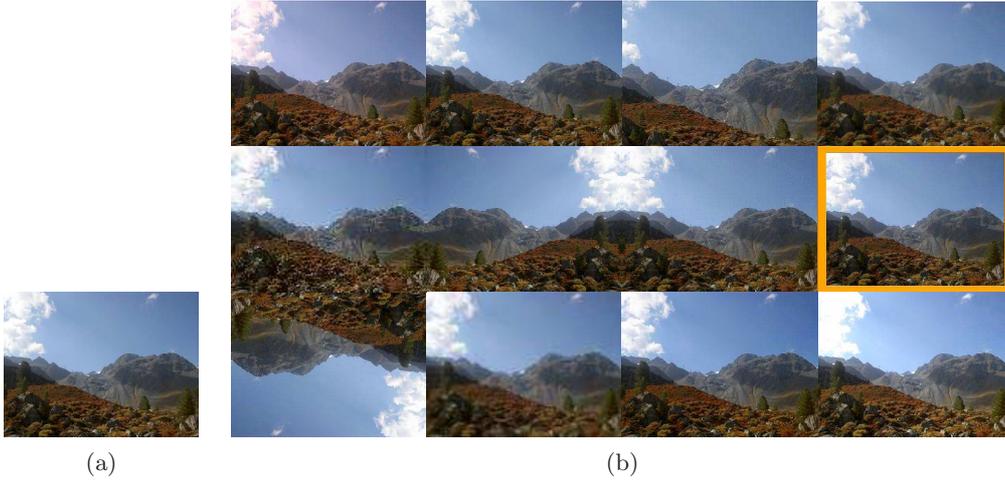


Figure 2. *Examples of Test Replicas.* (a) is the original while (b) represents examples of test replicas. There is one replica example per category, the order used is the same than in the text (left-right, top-down).

- *Changing Saturation.* Changes the values of the saturation channel by 70%, 80%, 90%, 110%, or 120%.
- *Changing Intensity.* Changes the intensity channel by 80%, 90%, 110%, or 120%.

Evaluation Metric In order to evaluate the performance of the proposed system, we measure the tradeoff between the *false positive* and *false negative* rates. The Receiver Operating Characteristic (ROC) curve²⁰ is often used to represent the tradeoff between error types; in this representation the true positive rate (one minus false negative rate) is plotted as a function of the false positive rate. In this study, we use a variant of the ROC curve called Detection Error Tradeoff (DET) curve.²¹ Contrary to ROC curves, the DET curves represent the

false negative rate as a function of the false positive rate. Since both axes correspond to error measurements, they can both make use of a logarithmic scale. The interpretation of DET curves is analogous to that of ROC curves: a classifier X is more accurate than a classifier Y when its DET curve is ‘below’ that of Y.

The performance is assessed as follows. For every original images selected in the previous section, replica partitions are constructed using Algorithm 1. Then, each image in the test set is used with Algorithm 2. The system’s tradeoff between false positive and false negative rates is assessed by making use of a single averaged DET curve. A false negative occurs if a replica is not detected as such (that is, \mathcal{C} does not contain the correct original) while a false positive happens if a test image is wrongly detected to be a replica (that is, \mathcal{C} contain originals that are not correct). Note that a test replica detected as the replica of some original and that \mathcal{C} contains three potential originals but none is actually correct, it results in two false positives and one false negative. A DET curve is then computed using the threshold averaging Algorithm in,²⁰ where the false negative rates are given by one minus the true positive rates and search boxes of different sizes (by varying the half-side lengths δ in Algorithm 2) are used instead of different thresholds. In the following, we use half-side lengths going from 0 to $7 \cdot 10^{-3}$. Finally, the precision of the DET curve is limited since the number of test examples is relatively low. Taking into account the number of test replicas ($150 \cdot 40$) and the number of negative examples ($N = 18,285 + (150 - 1) \cdot 40$), the achievable precisions are $\pm 1 / (18,285 + (150 - 1) \cdot 40) = \pm 4.1 \cdot 10^{-5}$ for the false positive rates, and $\pm 1 / (40 \cdot 150) = \pm 1.7 \cdot 10^{-4}$ for the false negative rates.

3.2. Influence of the number of levels N_L

Figure 3a shows the system performance for the two first levels of granularity. The bold dotted line represents the DET curve for $N_L = 1$ (varying the deltas). The DET curves for $N_L = 2$ are represent by the hairlines. To plot these curves, working points are first fixed for $N_L = 1$ (denoted by dots in Fig. 3a). It can be seen that each DET curve at $N_L = 2$ starts from the chosen working point and features an initial sharp decrease in the number of false positives (for a marginal increase in the number of false negatives) followed by a steep increase in the number of false negatives. This shows that using finer granularity features improves the performance. It can also be observed that a vertical asymptote exists around a false positive rate of 10^{-4} . This asymptote corresponds to four images that are always classified as positive by the system. This occurs because the image collection contains some photographs from the same scenes but at different angles.⁸ Consequently, if one of these multi-view photographs is also one of the selected originals, the proposed system is unable to distinguish the other views from the pure replicas of the corresponding original. Finally, note that using an additional granularity level $N_L = 3$ does not bring any additional increase of performance.

Additionally, the bold line in Fig. 3a represents the envelope of the DET curves for $N_L = 2$, or the best achievable performance. The envelope shows that the performance gap increases as the false positive rate diminishes. For instance, for a fixed false negative rate of 0.01 the false positive rate goes from $4 \cdot 10^{-3}$ ($N_L = 1$) down to $0.7 \cdot 10^{-3}$ ($N_L = 2$) or more than a fivefold improvement. Also note that the envelope corresponds to search boxes having the half-side lengths of $\pm \delta_s(1)$ and $\pm \delta_s(2)$ as shown in Fig. 3b. The relationship between the box half-side at different level can be estimated by the simple increasing function $\delta_s(2) = (1 - \exp(\delta_s(1) \cdot p_1)) \cdot p_2$, where p_1 and p_2 are parameters that need to be fitted (a fit is shown in Fig. 3b).

3.3. Influence of the number of dimensions d

We now turn our attention to the influence of the number of dimensions d . We changed the number of dimensions to $d = 25$ and performed the same experiments as in Sec. 3.2. The resulting curves are shown in Fig. 4. It can be seen that surprisingly the performance is poorer than when using only $d = 15$. There are several possible explanations to this phenomenon. On the one hand, the use of more NNs implies larger bounding boxes and consequently, replica partitions that are more extended. On the other hand, while 77 training examples (see Sec. 2.2) are enough to train the system for $d = 15$, their number seems insufficient for $d = 25$. Indeed, in order to avoid over-fitting more training examples are needed as the number of dimensions increases (the *curse of dimensionality*²²).

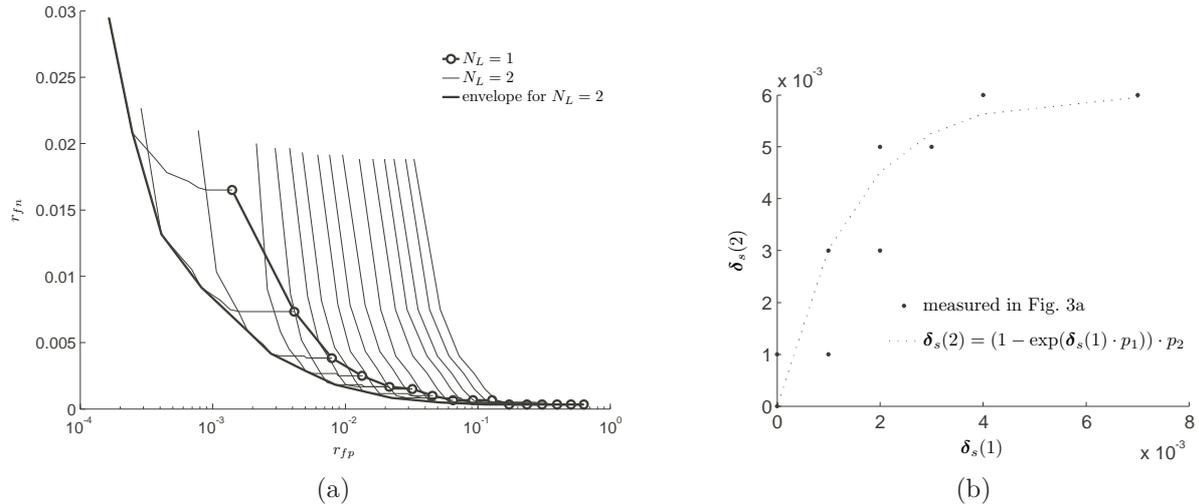


Figure 3. The DET curves for $d = 15$ are presented on the left. On the right, the half-side lengths δ_s determining the envelope are depicted.

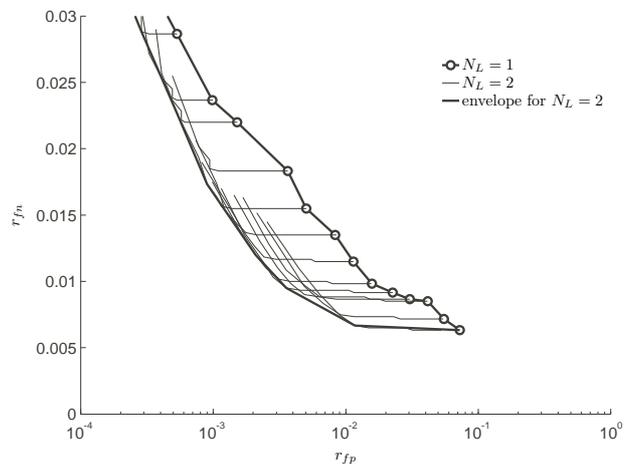


Figure 4. The DET curves for $d = 25$.

3.4. Comparison with other methods

Figure 5 compares the performance of the proposed replica detection system with state of the arts techniques in,^{8,9} as well as our previous related work.²³ The continuous line corresponds to the DET curve obtained with our currently proposed system, using $d = 15$ and $N_L = 2$. The dashed line represents the performance of a replica detection method based on Perceptual Distance Function (DPF).⁹ The circle point indicates the performance of a replica detection system based on Key Points (KPs).⁸ Both methods are set in the image retrieval framework and, therefore, give the result in terms of Precision-Recall measurements. It is however possible to translate a Precision-Recall curve into a DET curve.¹² Finally, the dotted line represents the performance obtained by our previous system.²³

It can be seen that the proposed method achieves good performance. For instance, an average false negative rate of 2% corresponds to a fixed false positive rate of $2.5 \cdot 10^{-4}$. On the one hand, the proposed method largely outperforms that of DPF for false positive rates below $4 \cdot 10^{-2}$. Our previous method is also greatly outclassed

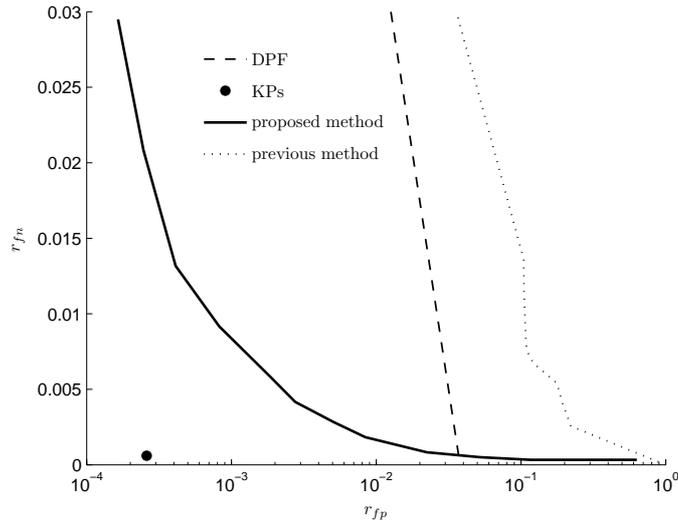


Figure 5. Comparison with other methods.

by the current scheme[‡]. Moreover, it should be noted that the features used in the current work are mainly a subset of those used in DPF: we use 133 features against 298 in the latter study. On the other hand, the proposed method is outperformed by KPs. In our method, most of the wrongly classified replicas (false negative errors) correspond to replicas for which the illumination or the intensity have been changed to a great extent. The KPs method uses features (salient points, or key points²⁴) invariant to this change but computationally more complex to extract. Indeed, the feature extraction time of KPs is between 1 and 10 seconds per image.^{8,9} This is between 10 to 100 times slower than that for the proposed method (when optimised as in⁹).

This result shows that it is possible to trade features complexity for training complexity and still obtain competitive performance. The use of simpler features permits a faster online detection. Moreover, the performance can be improved by testing each entry of \mathcal{C} with an adapted binary classifier.¹²

4. CONCLUSIONS

In this work, a replica detection system capable of retrieving from a database of originals those that correspond to a given suspect image was presented. The system is based on the estimation of the partition on which the replicas of an original lie, as well as the efficient indexing of the partition using R-trees. The experiments showed interesting performance. For instance, an average false negative rate of 2% corresponds to a fixed false positive rate of $2.5 \cdot 10^{-4}$. Moreover, this paper showed that it is possible to trade features complexity for training complexity and still obtain competitive performance. The use of simpler features permits a faster online detection.

Future works will focus on improving the performance, by including additional features, as well as determining the optimal training examples for a given set of image transformations.

ACKNOWLEDGMENTS

The first author is partly supported by the Swiss National Science Foundation — “Multimedia Security”, grant number 200021-1018411. This research was also partially supported by the European Commission under contract FP6-027026-K-SPACE;²⁵ this work presents the view of the authors but not necessarily that of the community.

[‡]Note that for lower false positive rates our previous method also outperforms DPF.

REFERENCES

1. ISO/IEC JTC1/SC29/WG11 WG11N4358, "Final Draft Int'l Standard 15938-3 Information Technology - Multimedia Content Description Interface - Part 3 Visual," July 2001.
2. Martínez, J.M. and Koenen, R. and Pereira, F., "MPEG-7: The Generic Multimedia Content Description Standard," *IEEE Multimedia* **9**, pp. 78–87, April 2002.
3. ISO/IEC JTC1/SC29/WG1 WG1N3684, "JPSearch Part 1 TR - System Framework and Components," July 2005.
4. F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. of the IEEE* **87**, pp. 1079 – 1107, July 1999.
5. F. Lefèbvre, B. Macq, and J.-D. Legat, "Rash: Radon soft hash algorithm," in *EURASIP European Signal Processing Conf.*, (Toulouse, France), September 2002.
6. J. Seo, J. Haitisma, T. Kalker, and C. Yoo, "Affine transform resilient image fingerprinting," in *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, pp. III 61–64, (Hong Kong, China), April 2003.
7. R. Venkatesan, S.-M. Koon, M.-H. Jakubowski, and P. Moulin, "Robust image hashing," in *IEEE Int'l Conf. on Image Processing*, pp. 664–666, (Vancouver, Canada), September 2000.
8. Y. Ke, R. Sukthankar, and L. Huston, "An Efficient Parts-Based Near-Duplicate and Sub-Image Retrieval System," in *ACM Int'l Conf. on Multimedia*, pp. 869–876, (New York, USA), 2004.
9. A. Qamra, Y. Meng, and E. Y. Chang, "Enhanced Perceptual Distance Functions and Indexing for Image Replica Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **27**, pp. 379–391, March 2005.
10. Y. Maret, F. Dufaux, and T. Ebrahimi, "Image Replica Detection based on Support Vector Classifier," in *Proc. SPIE Applications of Digital Image Processing XXVIII*, (Santa Barbara, USA), August 2005.
11. Y. Rui, T. Huang, and S. Chang, "Image retrieval: current techniques, promising directions and open issues," *J. of Visual Communication and Image Representation* **10**, pp. 39–62, April 1999.
12. Y. Maret, F. Dufaux, and T. Ebrahimi, "Adaptive Image Replica Detection based on Support Vector Classifiers," *Signal processing: Image Communication*, 2006. To be published.
13. B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **18**, pp. 837–842, August 1996.
14. M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Trans. on Information Theory*, pp. 179–187, 1962.
15. J.-L. Leu, "Computing a Shape's Moments from its Boundary," *Pattern Recognition* **24**(10), pp. 949–957, 1991.
16. J. Smith and A. Natsev, "Spatial and Feature Normalization for Content Based Retrieval," in *IEEE Int'l Conf. Multimedia and Expositions*, **1**, pp. 193–196, (Ischia, Italy), 2002.
17. R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience, 2000.
18. A. Guttman, "R-trees: a dynamic index structure for spatial searching," in *Proc. ACM Int'l Conf. on Management of Data*, 1984.
19. "ImageMagick: a free software suite to create, edit, and compose bitmap images, URL <http://www.imagemagick.org>."
20. T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," 2003. Technical Report HPL-2003-4.
21. A. Martin, G. Doggintgton, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," in *Proc. Eurospeech*, pp. 1895–1898, (Rhodes, Greece), 1997.
22. D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," August 2000.
23. Y. Maret, S. Nikolopoulos, F. Dufaux, T. Ebrahimi, and N. Nikolaidis, "A novel replica detection system using binary classifiers, r-trees, and pca," in *International Conference on Image Processing, IEEE*, October 2006.
24. D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. of Computer Vision* **60**(2), pp. 91–110, 2004.
25. "A European Network of Excellence on "Knowledge Space of Semantic inference for automatic annotation and retrieval of multimedia content", URL <http://www.kspace-noe.net/>."