Challenging the Security of Content Based Image Retrieval Systems

Thanh-Toan Do, Ewa Kijak, Teddy Furon, Laurent Amsaleg

Challenging the Security of Content-Based Image Retrieval Systems

Thanh-Toan Do [†1], Ewa Kijak [†2], Teddy Furon [‡3], Laurent Amsaleg [*4]

[†] *Université de Rennes 1, IRISA*
Campus de Beaulieu, Rennes, France
[1] thanh-toan.do@irisa.fr
[2] ewa.kijak@irisa.fr

[‡] *INRIA Rennes Bretagne - Atlantique*
Campus de Beaulieu, Rennes, France
[3] teddy.furon@inria.fr

[*] *CNRS, IRISA*
Campus de Beaulieu, Rennes, France
[4] laurent.amsaleg@irisa.fr

***Abstract*—Content-Based Image Retrieval (CBIR) has been recently used as a filtering mechanism against the piracy of multimedia contents. Many publications in the last few years have proposed very robust schemes where pirated contents are detected despite severe modifications. As none of these systems have addressed the piracy problem from a *security* perspective, it is time to check whether they are secure: Can pirates mount violent attacks against CBIR systems by carefully studying the technology they use? This paper is an initial analysis of the security flaws of the typical technology blocks used in state-of-the-art CBIR systems. It is so far too early to draw any definitive conclusion about their inherent security, but it motivates and encourages further studies on this topic.**

I. INTRODUCTION

Content-Based Image Retrieval (CBIR)[1] refers to searching for digital material in large multimedia databases. While promoting the cultural and artistic value of multimedia content archives, CBIR has so far been used in very cooperative and friendly settings where it benefits content providers business and increase users digital experience enjoyment.

However, we now observe another use of this technology. CBIR is used to filter multimedia contents in order to protect the creation of the few from the piracy of the many [1]. For instance, CBIR spots the upload of copyrighted material on sharing platforms (YouTube) to either block or monetize it (ie. advertising revenues are shared with the copyright holders). CBIR is foreseen to prevent downloads from P2P networks in the so-called "graduated response". Overall, filtering is an application of CBIR that is quite different from its primary goal: the environment is now hostile in the sense that filtering restricts users freedom, controlling and/or forbidding distribution of contents. Here, systems no longer magnify any cultural richness, but protect the commercial value of contents. Because there are valuable goods to protect, serious hackers will try to circumvent these systems. Therefore, it is legitimate to carefully investigate the *security* side of CBIR as done in this paper.

An abundant literature assesses that today's CBIR is *robust* in the sense that it recognizes contents despite editing processing (cropping, compression, blurring, etc) pirates apply on images to make identifications harder. *Security*, however, is not robustness. A pirate attacks *a particular system*, first observing and accumulating knowledge about the details of the CBIR techniques in use. He then leads attacks on very specific parts where flaws have been identified. Since the goal is to delude a particular system, security attacks have bigger success rates than usual editing processing only challenging robustness. Two short examples might help capturing the differences between robustness and security in the context of CBIR.

Content Concealment: The goal of the pirates is, in this case, to upload some illegal material inside a UGC platform such that it is not detected, concealed from the content filter. From various information sources, trials and errors, one can learn what specific technique is used to extract signatures from images ([2] does this for audio signatures). Therefore, specific attacks deluding that particular fingerprinting method can be designed (see Section IV).

Abnormally Frequent Identifications: As pirates tweak HTML pages to get ranked higher in textual search engines ("black hat SEO" attacks), one can tweak the visual contents of images such that they always get (artificially) ranked high in the result list of similarity searches. Here, a dishonest content owner may increase his revenues thanks to this exaggerated advertisement.

In the past, a problematic gap has been witnessed between security and robustness in the field of digital watermarking [3]. It is therefore of utmost importance to quantify that gap for CBIR, before they get widely deployed as filtering tools. This

[1]This paper deals with images, but of course, it extends to video and audio.

MMSP'10, October 4-6, 2010, Saint-Malo, France.
978-1-4244-8112-5/10/$26.00

paper sheds a security-oriented light on CBIR systems. We want to see if challenging their security is feasible in practice.

This paper is structured as follows: After a short threat analysis in Section II, Section III, first surveys the major state-of-the-art core technology blocks found today in CBIR systems and discusses an initial set of potential security flaws for each block; Section IV then shows, through three illustrations, that it is possible, for a well documented pirate, to pull down the recognition capabilities of a system.

This preliminary study suggests that it is unlikely any pirate will break an entire system by attacking only one of its components. Instead, taking advantage of many little security flaws is probably the way to go. Deluding a real system certainly asks to attack each of the various stages of image recognition, possibly also attacking earlier, at database creation time.

II. THREAT ANALYSIS

This section briefly describes some elements of the threat analysis of CBIR.

Trust Model: There are typically four different parties that may or may not be trusted in any scenario involving CBIR: (i) The image right-holder who is entitled to upload his images or their low-level signatures in the database; (ii) The image server where a collection of signatures is indexed in a database that is used for building the answers of content-based queries; (iii) The querying user who comes up with a particular image to search for; and (iv) The client software which processes the query image, connects to the server, send requests, receives and processes answers before displaying them. Note that the most classical copy detection scenario for attacks is when the user is the pirate, all other parties being trusted.

Goals: The pirate might have the following two main goals (this list is non-exhaustive): (i) Producing false negatives when the pirate manipulates images that the CBIR will subsequently fail to detect; (ii) Producing false positives when the pirate manipulates images such that it will always be detected by the system (even truly innocuous contents).

Measurements: Hacking CBIR is quite different from hacking a cryptosystem where the disclosure of the secret key grants full decryption of ciphertexts. Here, the success or failing of an attack cannot be simply measured by a binary answer. Attacking an image results in content manipulation that induces distortion. The PSNR[2] measures this distortion with respect to the original image in terms of Euclidean distance in logarithmic scales (the bigger the PSNR, the lower the distortion). Not surprisingly, the stronger the distortion, the higher the success chances of an attack. Moreover, the success of an attack depends both of the considered image and of the database content, stemming in a probability P of attack success. Therefore, an attack against CBIR is characterized by an operating curve that represents P as a function of the PSNR. The key issue of CBIR security is whether security attacks have much more powerful operating curves than edition processing, by increasing P for a given PSNR.

[2]Peak Signal to Noise Ratio.

Knowledge: The level of knowledge the pirate has on a system is of course crucial. However, it is hardly possible to perfectly know all the details of a real system: despite the fact that its algorithms might be published, the subtleties of their implementation, the exact values of the numerous parameters, etc., may all increase its obscurity level. Yet, the algorithms and their parameters have all been fined tuned to maximize robustness, and, therefore, they are not random. They are not, as such, secret keys and it might be possible to estimate their values, thus endangering the system. Overall, when the knowledge one can get on a CBIR system is simply limited by obscurity, we designate by *Worst Case Attacks* all attempts for maximizing the probability of successful attack for a given distortion value.

According to Kerckhoffs' principle, solely relying on obscurity does not make secure systems and adding secret information is important. Recently, few secret-keyed CBIR techniques were proposed [4]. However, no study analyzes their security levels. With a full access to a secret-key protected signature computation software, a pirate may create images and observe their resulting signatures which convey information about the secret key. The security level could then be measured as the needed number of couples (images, signatures) to eventually gain knowledge about this key. We designate by *Security Attacks* all attempts for disclosing secret information. Sometimes, disclosing secrets turns out to be impossible. In this case, what we designate by *Oracle Attacks* may also be relevant to CBIR.[3] *Oracle Attacks* do not try to estimate a secret key, but rather try to guess what in the data given to a system matter given the current secret key. For example, a secret key might drive the way colors get quantized at image signature creation time [4]. Instead of estimating this key, and therefore reveal the entire quantification process, it is possible to carefully play with pixel colors of pirated images, and check whether the system decides they are copies or not. By multiplying such tests, it may be possible to find the closest content to a given image which is not deemed as copy. Note that getting the secret key itself is never tried here. The number of trials, the convergence speed are metrics often associated to *Oracle Attacks*.

III. SECURITY OF CBIR

CBIR systems are composed of a small set of building blocks and this section emphasizes where, in each block, pirates are likely to concentrate their attacks.

A. Image Description

CBIR assesses the similarity between images by comparing some low-level high-dimensional visual features extracted from the pictures that are called descriptors. A large variety of description schemes exist, based on colors, textures, edges or any combination, or on image transforms like Fourier, wavelets, DCT. To be useful in practice, descriptors are to be robust and can absorb at various degrees some image editing processing (rotations, rescalings, color changes, ...).

[3]*Oracle Attacks* as defined by the watermarking community [5].

The description algorithm may create a single descriptor from the entire image, in this case referred to as being *global*. Global descriptors are often quantized for the purpose of compactness and robustness, producing a compact and discriminative robust hash. Alternatively, many *local descriptors* can be extracted per image, each describing a different image region. They are obtained using a two step approach: regions are first extracted from images [6] and then descriptors are computed using the contents of the region [7].

When a pirate attacks that part of the system, its goal is to produce specific modifications of the images entirely driven by the deep understanding of the properties of the description scheme, such that matches with descriptors of the database will subsequently fail. In the case of local descriptors, the two phase extraction process gives pirates several handles: either perturbing the region extraction, the descriptor computation, or both. Attacking the region extraction step, a pirate may want to artificially add new regions, delete or move detected regions by applying well-chosen local modifications. Having new regions by patching the images in turn creates new descriptors, that may make the identification of the pirated image harder.

B. Indexing and Retrieval

Indexing inserts the descriptors in a particular data structure. Retrieval navigates very efficiently in that structure to retrieve the k-nearest neighbors or to do an ε-range search for each low-level query descriptor. These methods take advantage of some form of partitioning of the descriptor collection into cells. At query time, confining the search to few cells provides efficiency [8]. The dimensionality curse increases the likelihood of having descriptors close in space separated by a partition border. Therefore, several (neighboring) cells get analyzed during the processing of one query, or descriptors get assigned to multiple cells at index construction time [9]. All in all, the accuracy of a system depends on the local quality of the neighborhood around descriptors. Its response time largely depends on the number of elements from the database to compare the query descriptor to.

Given these typical technology traits, pirates may envision two grand angles for attacking the database part of CBIR. Since efficient schemes are all approximate, one angle is to tweak the descriptor collection such that the approximations made by the system have a more severe impact on the accuracy of the result. Another way is to increase the cost of queries by forcing extra processing as the search has more difficulties to find query results and therefore needs to go deeper in the collection. By significantly increasing that cost, the system may become overloaded and unresponsive, therefore unable to play its role.

There are various possibilities for generating these attacks but they all boil down to modifying the descriptor collection such that specific phenomenons arise. Impacting accuracy and/or cost is possible by artificially exaggerating the likelihood that the system runs into frontier problems or faces severe skews in the cells' cardinality. A pirate knowing about the location of partition borders can try to produce descriptors that are slightly shifted in space, such that they get assigned to different partitions, effectively separating near neighbors, thus reducing accuracy.

C. Typical Optimizations

To either improve the quality of the results and/or to reduce query response times, most CBIR systems use various optimizations. We briefly review here two tricks found in existing systems, and show how pirates can divert their effects.

Some CBIR systems have been specifically designed to handle local descriptors. In this case, query time is often high because hundreds of descriptors are used to probe the database at search time. In [10], special stop-rules have been designed to abort as soon as possible the search process, trying to use only a small subset of the descriptors in the query. In a copyright enforcement scenario, stop-rules assume that pirated images receive a lot of descriptors matches, while unrelated images receive few random matches, roughly distributed over the whole image collection. Therefore, after having processed a fixed number of query descriptors, a stop-rule aborts the processing if one particular image very rapidly collects many matches; another rule terminates the search if all scores stay roughly equal and low, suggesting no copy detection. In [10], as few as 20 descriptors matches (this is roughly 2%) were found sufficient for finding a copy, and about 100 descriptors had to be used to decide it is not a copy. A possible attack is to try to add to a pirated copy a bunch of non-matching descriptors that will be used first in the querying process. In this case, stop-rule may incorrectly decide a failure, as none of the descriptors describing the true contents to protect are used, the search being stopped before getting to them.

In addition to these mechanisms, CBIR may include optimizations for removing false positives. One typical method checks the geometrical consistency between the query image and the candidate images. Many approaches rely on Hough transforms, which do a good job for detecting affine transforms. Two quasi-identical images differing by non affine modifications, however, fail to match. It is therefore possible to attack such schemes by deforming a portion of an image in a non-affine way.

IV. EXPERIMENTS

Through three independent experiments, this section shows that it is possible to pull down the recognition capabilities of a system by attacking existing techniques. The scenario is the following: the pirate is a dishonest user aiming at producing false negatives, i.e., concealing copies from the CBIR. We also suppose that no secret key mechanism exists, which is the case for actual CBIR.

A. Challenging Keypoint Detection

The SIFT local description scheme [11] is a popular choice for CBIR based copyright infringement detection. In this scheme, some interest points of the image, referred as keypoints, are first detected and regions are defined by the neighborhood of keypoints. Then, the descriptor, a 128-dimensional

TABLE I
REMOVAL OF KEYPOINTS, VARYING δ

δ	total	rem.	unch.	new	psnr
0.01	190	124	159	31	49.4
0.02	193	181	102	91	38.2
0.03	197	214	69	128	34.9
0.06	205	264	19	186	30.6

vector, is computed from each region. We solely focus here on the effects of preventing keypoints (and then regions) from being detected.

Roughly speaking, keypoint detection relies on the values of coefficients $D(x, y, \sigma)$, so called Difference-of-Gaussian at scale σ, extracted from the query image at location (x, y). A keypoint $\mathbf{x} = (x, y, \sigma)^T$ is detected if the three following conditions hold: (i) $D(\mathbf{x})$ is a local extrema over a neighborhood of $\mathbf{x}$, (ii) a sustainable contrast is present (i.e. $|D(\mathbf{x})| > C$ with C a fixed contrast threshold), and (iii) $\mathbf{x}$ is not located on an edge.

Removing a keypoint $\mathbf{x}$ asks to invalidate at least one of these three conditions. This is possible by modifying the image I on the neighborhood $\mathcal{V}(x, y)$ on which $D(\mathbf{x})$ is computed:

$$I'(i, j) = I(i, j) + \epsilon(i - x, j - y), \forall (i, j) \in \mathcal{V}(x, y),$$

where ϵ is a patch of limited support. [12] proposed a patch for invalidating (i) such that the original local extremum $D(\mathbf{x})$ becomes equal to the second extremum in its neighborhood. In the same way, invalidating (ii) asks to create ϵ such that $|D(\mathbf{x})|$ becomes smaller than C. Besides [12], we also implemented our own method based on a Lagrangian solution which finds patches with the minimal Euclidean norm in order to reduce the perceptual degradation. Limited space, however, forbids including detailed results (see [13] for details).

For both techniques, we observed that adding patches to images to remove keypoints is indeed effective: usually, a significant portion of the keypoints in original images are removed. However, we also observed that a patch tends to trigger the creation of new, unintentional, keypoints. This phenomenon is illustrated in Table I using the well-known "Einstein" image, that has originally 283 keypoints. It shows how many keypoints are detected in the patched image (total), how many existing keypoints are removed (rem.), how many keypoints are left unchanged (unch.), how many keypoints get created as a side effect of the removal (new), and what is the resulting PSNR. In this experiment, $C = 0.02$ and keypoint removal is controlled by defining a value δ that targets the keypoints having contrast values within a particular range defined as $\mathcal{E}_\delta = \{\mathbf{x} : C < |D(\mathbf{x})| < C + \delta\}$. Note the more keypoints removed, the smaller the PSNR. The same experiment was conducted on 1,000 images, and the same behavior has been observed: on average about 77% of keypoints are removed (over 1,026 keypoints per image on average), and 53% are created.

We also observed that new created keypoints tend to be located very close to the one removed, and tend to define very similar regions. Therefore, the descriptors eventually generated on the attacked image tend not to be that different from the ones computed over the original picture.

Several major lessons can be drawn from our observations. First, it is unlikely that all SIFT-like keypoints can be removed without severely impacting the quality of the image. The remaining keypoints have matching descriptors, and very likely, so are the ones unintentionally created. We thus believe it will be difficult, if not impossible, to conceal a pirated image by solely relying on keypoint removal. This contradicts the conclusions of [12]. Second, some keypoints are easier to remove than others, due for example to their scale; some do not trigger the creation of a nearby keypoint; some might not be removed but either shifted in space or in scale. We observed that by carefully choosing which keypoints to remove, it is possible to reduce the score with which one query image matches with images from the database, therefore making the recognition less pronounced, hence potentially problematic if combined with other attacks.

B. Challenging a Global Description

Not limiting ourselves to local descriptors, we investigate here one possible attack against a global description scheme: Since many schemes quantize features for robust hash creation, we attack here the well-known Scalable Color Descriptor (SCD) from MPEG-7 [14] that also performs quantization. Quantization not only provides a compact representation of the extracted features into a binary hash, but also sets the trade-off between robustness and diversity. A big quantization step improves the robustness because a feature of the distorted copy is still quantized in the same bin as its undistorted version, but it also lowers the entropy of the hash, and hence the likelihood of collisions. Although this descriptor is not sufficiently robust to be used for copy-detection purpose, it is widely used in CBIR system proposing similarity searches.

The key assumption made here is that the pirate knows about the location of the quantization bins. For a given content, some features are obviously more prone to be attacked than others: the features lying near the quantization frontiers should be moved first outside their bin. This tends to deeply modify the descriptor while minimizing the visual distortion.

The SCD creation process on one image starts with a uniform quantization of the HSV space, with 16 levels for H, 4 levels for S as well as for V: $\hat{h}(i, j) = Q_{16}(H(i, j))$, $\hat{s}(i, j) = Q_4(S(i, j))$, and $\hat{v}(i, j) = Q_4(V(i, j))$. This leads to a 256 bins histogram: $p(\hat{h}, \hat{s}, \hat{v})$. The histogram values are then non-uniformly quantized in a 4-bit representation to achieve more efficient encoding: $\hat{p}(\hat{h}, \hat{s}, \hat{v}) = Q_{16}(p(\hat{h}, \hat{s}, \hat{v}))$. This gives higher significance to the small values, while important ones may be truncated. Finally, the histogram is encoded using a Haar transform. SCDs are compared using the L_1 norm.

The attack principle is the following: the pirate is allowed to change a given percentage of pixels color values, in order to deeply modify the descriptor, while minimizing the visual distortion. Knowing the quantization step, we can determine, for each bin $(\hat{h}, \hat{s}, \hat{v})$, the minimum number of pixels $\delta_{\hat{h},\hat{s},\hat{v}}$ to

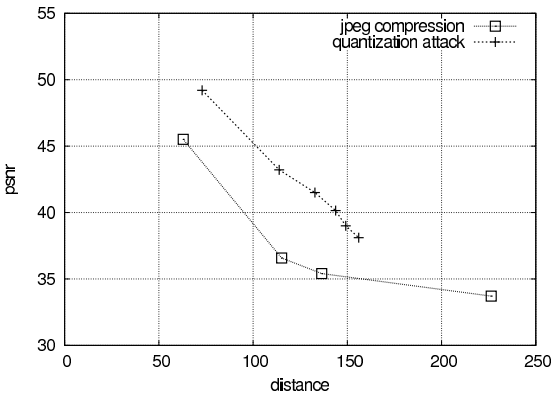

Fig. 1. Attacking SCD

Fig. 2. Original & modified images

move in order to change the quantized value $\hat{p}(\hat{h},\hat{s},\hat{v})$. Each bin $(\hat{h},\hat{s},\hat{v})$ is ranked in increasing $\delta_{\hat{h},\hat{s},\hat{v}}$ order and iteratively proceed until the allowed amount of changed pixel is reached. The severity of the attack depends on this last parameter.

In a bin $(\hat{h},\hat{s},\hat{v})$, the $\delta_{\hat{h},\hat{s},\hat{v}}$ pixels to modify are chosen among those which $H(i,j), S(i,j), V(i,j)$ values are the nearest from quantization frontiers. To ensure coherence, a pixel (i,j) can not be modified if (i) it has already been modified in a previous iteration, (ii) its new quantized value $(\hat{h}',\hat{s}',\hat{v}')$ corresponds to a bin already processed.

This *Worst Case Attack* is powerful if the distance between the SCD of the original and attacked images is greater than any distance between the original image and quasi-copies forged using typical editing processing challenging robustness, like a JPEG compression. Figure 1 compares the average descriptor distance versus the average PSNR computed over a set of 100 images randomly taken from Flickr, so that they present a great content diversity. We vary the attacks severity through, respectively, the percentage of modified pixels, and the quality factor of jpeg compression. For each attacked image, we compute its distance and PSNR to each corresponding original image. The JPEG quality factor is set to 80, 60, 40, and 20, while for the quantization attack, the percentage of modified pixels takes value among 1%, 5%,10%, 15%, 20%, and 25%.

Results show that for a given descriptor distance, the PSNR of the quantization-hacked image is greater than for the JPEG compression by an amount of 5dB. Note that the increase in image quality is limited due to the inherent low robustness of the SCD to editing transformations. Conversely, for a given PSNR, the average descriptor distance between the original and hacked images is larger than with the JPEG compression. This means the hacked images are further away in the feature space, offering opportunities for reducing the recognition quality of the system. This illustrates that a dedicated attack has a better operating curve (as defined in Section II) than an editing processing.

C. Challenging Retrieval

To demonstrate the feasibility of attacking the database part of CBIR systems, we show it is possible to conceal copies when the internals of the retrieval process is known. In this toy example, the pirate knows that (i) at search time, query descriptors are processed sequentially, as stored in the query file and (ii) a stop rule mechanism exists (see Section III-C). This example uses a real CBIR system [10] indexing 1,127,918 high-resolution images randomly downloaded from Flickr, resulting in a total of 1,222,920,344 SIFT descriptors. On top, we simulated stop rules by evaluating, after having processed a varying percentage of the query descriptors (20%, 30%, 40% and 50% here), how confident can we be that the query is a copy, that the query is likely not a copy even with further processing, or that more query descriptors need to be processed to eventually push the decision one way or the other. This gives a value to a decision flag, respectively "found", "failed" or "cont.", impacting the continuation of the search process.

To challenge the system, we take one particular image that is in the database and produce a copy by inserting a small box of text at the top (Figure 2). The original image has 511 descriptors, the modified 640, and all the 129 new descriptors are around that text box. Then, we sort all descriptors according to their increasing row location and submit that sorted file to the system. Note all the 129 new SIFT appear at the beginning of this file. For comparison, we also submit the original unmodified query file sorted on row location. Table II details the results.

In this table, each line shows the number of descriptors used by the search process (#qd) at each percentage. It also shows the score of the best image candidate found so far and the value of the stop-rule flag. The 20% line says, for the original image, that after 102 query descriptors, the best score is 12, and this is not enough to decide to stop. Therefore, the search

TABLE II
TRACES OF EXECUTIONS WITH STOP-RULES.

% of query	Search with original (511)			Search with modified (640)		
	#qd	Score	Flag	#qd	Score	Flag
20%	102	12	cont.	128	¬ 4	failed
30%	153	18	found	192	10	cont.
40%				256	13	cont.
50%				320	20	found

carries on, and after having processed 30% of the query, the flag is set to "found", the search stops and the recognition is successful (saving 358 query descriptors).[4]

In contrast, for the attacked image, that 20% line says that after 128 descriptors, the best score is 4, the best candidate image is indeed the wrong one (this is indicated by the ¬ sign; execution traces show the second best score is also 4), and that this score is so small with respect to #qd that the stop-rule decides to abort the search, failing to return any result. This was to be expected as only the descriptors associated to the text box have been considered so far in the attacked image and it turns out that they match with nothing from the database. Therefore, a system with stop-rules can be deluded because of the order according to which query descriptors are processed. Note that once more than 129 descriptors are considered for the attacked image (as if instead of first checking the stop-rule at 20%, the first check was at 30%), then the original image is starting to be found. The counter-attack is quite simple: process the local descriptors in a randomized order. Yet, as far as we know, no CBIR system does so.

V. CONCLUSION

In this paper, we have motivated why it is time to analyze content-based image retrieval systems from a *security* point of view. The starting point is the intuition that a pirate might severely delude a system by exploiting his knowledge on the system. We detailed a first set of security flaws found in some of the main state-of-the-art blocks used in existing systems. We then performed three experiments showing how to reduce their recognition capabilities. However, no conclusion about the security of nowadays CBIR can be drawn easily. On one hand, solely relying on keypoint removal is not at all a threat against filtering system. On the other hand, the very naive attack of subsection IV-C is effective, although the counter attack is quite simple.

Overall, we are convinced that the security of CBIR is a hot topic as it will eventually get seriously challenged. It deserves, however, a big amount of research efforts to design successful attacks. If these attacks exist, they are likely not focused on a particular block of the system (there does not seem to be a weak link in CBIR systems), but rather spread all over, stepping on every single breach in systems.

[4]We also queried the index with 49 standard modifications obtained by applying Stirmark on original images (crops, rotations, rescalings, filters, ...). The original images were always found, showing how robust descriptors are.

REFERENCES

[1] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *CIVR*, 2007.

[2] S. Smitelli, "Fun with YouTube's audio content ID system. http://www.csh.rit.edu/~parallax/."

[3] F. Cayre, C. Fontaine, and T. Furon, "Watermarking security: Theory and practice," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, 2005.

[4] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 2, 2006.

[5] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, "Blind newton sensitivity attack," *IEEE Proc. on Information Security*, vol. 153, no. 3, 2006.

[6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, 2005.

[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *PAMI*, vol. 27, no. 10, 2005.

[8] H. Samet, *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.

[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[10] H. Lejsek, F. H. Ásmundsson, B. Þ. Jónsson, and L. Amsaleg, "Scalability of local image descriptors: A comparative study," in *ACM Multimedia*, 2006.

[11] D. Lowe, "Distinctive image features from scale invariant keypoints," *IJCV*, vol. 60, no. 2, 2004.

[12] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Secure and robust SIFT," in *ACM Multimedia*, 2009.

[13] E. Kijak, T. Furon, and L. Amsaleg, "Challenging the Security of CBIR Systems," INRIA, Research Report RR-7153, 2009. [Online]. Available: http://hal.inria.fr/inria-00441816/en/

[14] ISO/IEC-15938-3, "Multimedia content description interface - part 3: Visual. ISO/IEC/JTC1/SC29/WG11/N4062," Singapore, 2001.