imPlag: Detecting Image Plagiarism Using Hierarchical Near Duplicate Retrieval

Siddharth Srivastava, Prerana Mukherjee and Brejesh Lall
Department of Electrical Engineering, Indian Institute of Technology, Delhi, India
{eez127506, eez138300, brejesh}@ee.iitd.ac.in

Abstract—**Plagiarism in any form is a serious offense especially in academia and industry where integrity and royalty from work is of utmost importance. In this work, a novel hierarchical feature extraction as well as an approximate nearest neighbor search is proposed for detecting plagiarism of images. The proposed scheme is applicable for natural images as opposed to specific image classes reported in a previous work. A comprehensive experimental analysis is provided to illustrate the efficacy of the techniques chosen for the scheme. We demonstrate that the scheme shows a lot of promise for a wide variety of attacks and is amenable to scaling.**

Keywords- Image Plagiarism, Content Based Image Retrieval, Feature Extraction, Feature Indexing

I. Introduction

Plagiarism in text is a well-known problem, but with increasing ease of access to information, plagiarism in images is also becoming widespread. As nowadays, textual data comes along with pictorial data, it has become necessary to find plagiarism in images as well. This would not only be beneficial for graphic designers, professional photographers, bloggers but also for publication agencies, legal experts trying to detect reproduction of their work without consent. The key characteristic of image plagiarism is that it may involve the reproduction of the original image using an entirely different mode such as hand made sketches. Image Plagiarism can be posed as a superset of image copy detection problems. Figure 1 gives an example to illustrate this difference.

Fig. 1. Example of difference between image plagiarism and image copy detection. (a) Original Image (b) Plagiarised image (reproduction of the source image) (c) Copied image (considered as strong attack by copy detection algorithms but an expected case for Image Plagiarism)

The users nowadays have been enabled to share multimedia content over the web through various channels such as Flickr, Youtube, Picasa, twitpic etc. This makes normal users vulnerable to their work being used without consent. The plagiarism in this case implies utilizing the quality work produced by users as someone else's work. Technically, this means that the derived images would have similar perceptual quality/ content with modifications in order to change the physical properties such that it may give viewer an appearance that the image is not from the plagiarised source. These physical deformations may involve embedding of logos, color space conversion, cropping etc.

Most relevant prior work in this area is on detecting and localizing image logo plagiarism [1]. It detects the plagiarised logos by computing geometrical distances between them. Our work is a broader treatment of the problem in the context of natural images. Other works for copy detection such as watermarking based techniques [2], are not suitable for avoiding or detecting image plagiarism for two reasons. Firstly, the image may have been modified before being watermarked. Secondly, the plagiarised image may be an artistic representation or a copy of the idea represented in the image, but in a different form (Ex: sketches). Another set of techniques that involve the similarity search between images are Content based Image Retrieval (CBIR). They have been widely used for medical image retrieval [3], tourist information systems [4], and variety of other applications as discussed in [5]. Interestingly, [6] shows by performing Content Based Image Retrieval on images of Sun that such systems need to be adapted to particular domain of problem and hence there cannot be a general method for designing such systems. Image Plagiarism is one such domain, where absence or lack of popularity of any standard dataset for this task makes comparison between various techniques a relatively difficult task.

In view of the above discussions, the major contributions of this paper are:

1) Development of a hierarchical feature extraction and feature indexing technique which provides effective and nearly interactive image plagiarism detection stack.
2) Evaluation of recent feature extraction techniques against simple, moderate and extreme deformations used specifically for plagiarising images.
3) We have also constructed a dataset for testing image plagiarism algorithms, a need generated by lack of any such database in literature.

In Section II, we give methodology used in the work. In Section III, we detail the experimental analysis and the results. Here, we have also give an exhaustive evaluation of the matching capabilities for these feature detectors as suited for image plagiarism tasks. In Section IV, we give the conclusion.

II. Methodology

In this Section, we give a detailed overview of the feature detection and indexing techniques. The workflow of the proposed method is shown in Figure 2. In the following subsections we describe the components in brief followed by a detailed description of the proposed methodology.

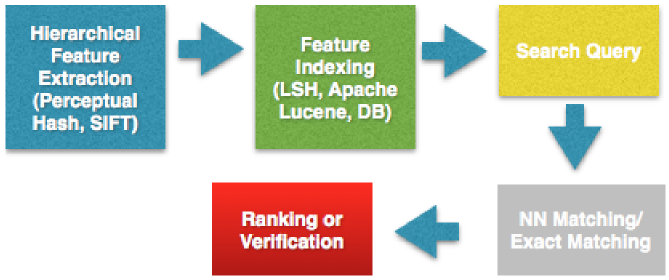

Fig. 2. Flow of an image plagiarism detection system

A. Feature Extraction

Feature detection is a crucial step in detecting image plagiarism as it enables to find similarities between images having imperceptible changes. Choice of using local or global features is a domain dependent task. Global feature detection techniques by themselves are not sufficiently discriminative for image retrieval [7] and are useful only if used along with local discriminative features. Apart from the traditionally proven SIFT [8], SURF [9] and PCA-SIFT [10], ORB [11], FREAK [12] and KAZE [13] have become popular. ORB and FREAK are binary features, claiming comparable feature matching capability with much lower computational complexity as compared to the other algorithms. ORB is based on FAST [14] keypoint detector and the descriptor it uses is based on BRIEF [15]. FREAK uses a different approach and it borrows various techniques from BRIEF and BRISK [16]. It is based on the retinal sampling pattern for comparing pixel intensities whereas ORB used a random sampling pattern to compare pair of pixels intensities. To speed up the matching it utilizes Saccadic search which doesn't compare the entire descriptor at once. Instead, it processes them in chunks of bit values. First a certain number of bits are compared and if they have a matching score above a threshold only then further bits are compared and so on. This is in striking contrast to SIFT type techniques, where the entire descriptor is compared by performing a nearest neighbour approach. the reltively recent KAZE, is based on non linear scale space which preserves the object boundaries and blurs the region around the edges.

Another technique, gaining popularity in recent times has is Perceptual hashing [17]. It is a block based feature extraction technique which aims at representing similar multimedia content (audio, image,video) with similar hash values. Perceptual Hashes of images preserve their basic structure and the qualitative aspects of the visual information content. Perceptual hashes have been widely used for authentication [18][19], digital watermarking [20], image reconstruction [21], multimedia content retrieval[22] etc.

B. Feature Indexing

Various techniques for exact and nearest neighbour matching have been discussed in [23]. A very effective technique is Locality Sensitive Hashing [24], which guarantees the nearest neighbor of a point with a reasonably high probability. There are many variations of LSH, a few notable ones are E2LSH for Euclidian space [25], Kernelized LSH [26] and Multi-probe LSH [27].

Fundamentally, LSH works by hashing each training input to t hash tables, where each point may be hashed using a k-bit hash function g.

$$g_i = [h_1(p), h_2(p), ..., h_k(p)] \tag{1}$$

where $i \in \{1, t\}$. Each hash function h is a bit hash for corresponding bit of point p in $\mathbb{R}^d$. The hash functions h are chosen at random from a family of Hash Functions H. The family of hash functions is locality sensitive in nature i.e. closer points are mapped to same hash values. It is calculated by random projections on to a plane chosen from a distribution such as Gaussian. The aim is to maximize the probability of the collision of similar values and minimize the collision of dissimilar values. At query time, the query q is hashed using the same function and buckets from all the tables are retrieved. A search is performed over the union of such buckets to give the final output.

C. Proposed Methodology

As noted in Section I, for a technique to be effective it has to be adapted to a particular domain.Therefore, in order to chose the features best suited for characterizing the deformations dominant in image plagiarism. We compare various feature detectors according to the steps outlined below:

1) The corresponding descriptors are calculated and stored for each image.
2) The original reference image is used as a query and a feature matching using FLANN [28] based matcher is performed. For matching binary descriptors LSH is used. It is important to note that for an exhaustive comparison and noting the importance of feature extraction in the overall quality of the system, false positive matches in the descriptors are discarded by comparing the actual transformation of the keypoints to the target images.
3) The average feature distance between each image is calculated and the results are ranked in the order of the distance. A smaller distance here means more similarity between the images since similar descriptors would have smaller distance.
4) For calculation of accuracy, the total number of relevant images in top $2 * N$ results are evaluated using Equation 2.

$$acc = \frac{\text{No of correct matches in top-k Results}}{\text{Total number of correct results (N)}} * 100 \tag{2}$$

where $k \approx 2 * N$.

In addition to the keypoint based methods, block-based methods like DCT have been found to be effective in the domain of copy-move-forgery detection [29]. The attacks expected for plagiarising an image typically consist of camouflaging a part of the image more than deforming the entire structure. The structure preserving capability of such

1) Calculate and Store Perceptual hash (64bit) for each image in a database.
2) Calculate and Store SIFT features. Convert the image to a bag of visual word representation and store the corresponding words and histograms in a database as described in the classical plagiarism detection approach.item Index Perceptual Hash strings using LSH.
3) While querying, calculate locality sensitive hash and return the nearest strings with corresponding distances.
4) These results are used as candidates. Now, instead of exact matching of keypoint features, calculate the distances between histograms and rank the results in reverse order of distances.

Fig. 3. Algorithm for Proposed Methodology

techniques, makes them a strong candidate for evaluation. Perceptual hash is one such popular technique. The images are hashed to a 64-bit binary string using the following method adapted from [30]:

1) Scale down the image to thumbnail size of 32x32 and convert to grayscale. This is done in order to preserve only the structure of the image.
2) Discrete Cosine Transform of the image is calculated and except the average of first 8x8 DCT vectors, all other coefficients are discarded. This is done to exploit the property of energy compaction; the structure of the image can be represented by the lower order coefficients.
3) Now, the DCT values which are above the mean are set to 1 and rest to 0. This matrix is then flattened to a 64bit binary vector, which is the perceptual hash of the image.

The proposed methodology is a hierarchical feature representation scheme utilizing Perceptual Hash and SIFT. In addition, a hierarchical retrieval scheme is also proposed. These steps are described in Figure 3 while the results and discussion follow in the next section.

III. Experiments and Results

A. Dataset

Image plagiarism doesn't exactly correspond to natural transformations of images but it deals with a synthetic deformation of images. The existing datasets either exploit only a few common synthetic deformations such as scaling, cropping etc targeted towards a specific domain or consist of natural transformations. Hence, there was a need for generating a suitable dataset for this task. A dataset of 200 base images was constructed from natural images collected from Google Image Search and Flickr. Out of the total 200 images, 100 images were collected with the query terms: "nature", "safari ", "flood", "mountain", "natural scene", "natural beaty", "agriculture", "animals", "birds", "environment". The other 100 distinct images were obtained by performing reverse image search queries on Google Reverse Image Search and TinyEye for a few of the first 100 images. Our experiments indicate that Google Image Search seems to use perceptual hashing, color information along with image metadata as a major similar image search criteria whereas TinyEye seems to utilize object detection, segmentation, face recognition etc in addition to those used by Google Reverse Image Search. The benefits of this selection criteria are two fold. First, it introduces noise in the dataset as is expected to be found in a practical collection of images over the web. Secondly, it requires the techniques to be reasonably robust against false positives and false negatives. The images were then exposed to various attacks as listed below:

T1. Cropping by 10%, 20% and 50% (around the center).

T2. Blurring by using Gaussian kernel of radius 5 and 10.

T3. Applying a color histogram on the images.

T4. Watermarking: Two watermarks (center and bottom left corner) were superimposed on the images.

T5. Rotation: Rotation of 10, 20, 50 and 90 degrees.

T6. Scaling: Scaled with a factor of 0.25, 0.5, and 2.

T7. Channel Separation: Each of R, G and B channels were separated and saved as a different image.

T8. Conversion to Grayscale.

T9. Format Change: Conversion to PNG, GIF.

T10. Cropping without scale change: T1 was applied again with image borders padded with black pixels to maintain the original size.

T11. Affine Transform: Five distinct affine transformations were applied to the reference images as shown in Figure 4.

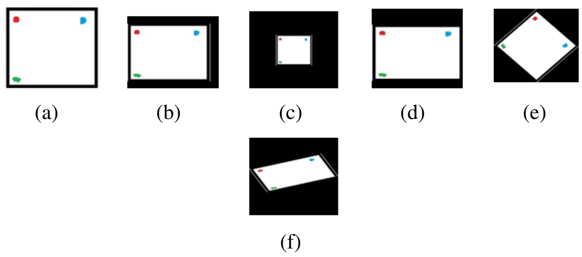

Fig. 4. Affine Transformations applied to the images.Figure 4(a): The reference image with red, blue and green patches considered for obtaining the transform matrix. Figure 4(b) - Figure 4(f): The corresponding corners in the affine transformed images.

T12. Sub-Imaging: Each image was scaled to 200x200. A flat super imposition of this image was performed on a different image, which is common for all the images in the dataset.

T13. Hand made sketches: A line drawing of the original image.

The dataset hence in total consists of 6400 images with 31 transformations on each image.

Now, an empirical discussion of various methods for applicability to image plagiarism as per the steps in Figure 2 are presented.

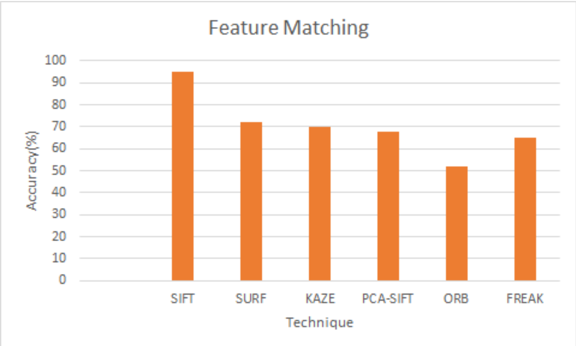

Fig. 5. Comparison of Feature matching techniques

B. Comparison of the feature extraction techniques

The results showing matching accuracies of the various feature detectors are shown in Figure 5:

As observed in the experiments, SIFT was the most robust algorithm with 95% accuracy followed by SURF and then KAZE and PCA-SIFT. ORB and FREAK were at the bottom as far as matching accuracy is concerned. There have been many studies justifying the relative performance observed above for SIFT, SURF and PCA-SIFT [31], [32]. ORB has the worst accuracy whereas FREAK has higher accuracy than ORB, which can be attributed to its descriptor considering a weighted pattern of pixel densities similar to SIFT around each keypoint. Given that the recent techniques(ORB, FREAK) have usually been evaluated on datasets consisting of only 4 transformations (scale, blur, intensity variation and rotation), which constitute of 35.4% of the total images in the our dataset. ORB (52%) and FREAK (68%) haven't performed poorly but given the comparison with SIFT, they are not suitable for the task of detecting Image Plagiarism.

C. Performance of Perceptual Hashing

The accuracy of the perceptual hash function for image retrieval was calculated using the using hamming distance, and it came out to be 84% which is very close to SIFT in terms of accuracy. The result can be understood by noting that the dataset is that of natural images which exhibit uniformity in the texture, color, edge transition that are preserved in the hash values computed from DCT and acts as a distinguishing factor. But as observed, an accuracy of 84% (and not 100%) also highlights that structural similarity depends upon the visual content in the image such as mountains, trees etc which might be prominent features when analyzed on a thumbnail size image.

Next, we compare the retrieval time performances.

The time was computed by taking an average over the total time taken to match all the images in the dataset plus the time taken to extract the features from the query image as shown in Equation (3):

$$T_{avg} = \frac{\sum_{k=1}^{N}(t_k + \sum_{i=1}^{N} t_i)}{N} \tag{3}$$

where,

T_{avg} = Average Retrieval Time,
t_k = Time taken to compute features for image k,
t_i = Time taken to match features of image k with N images, and
N = Total number of images in the dataset.

As expected and as can be seen from Figure 6, Perceptual hash took lesser time than both SIFT and SURF. In fact Perceptual hash has outperformed SURF for the accuracy as well. Hence, Perceptual Hash is a good choice for filtering the initial results for the task at hand.

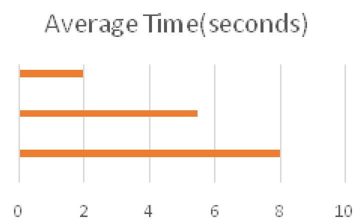

Fig. 6. Average time taken by SIFT, SURF and Perceptual Hash

Based on the above results, perceptual hashing alone should have been sufficient for image retrieval. But, as shown in Figure 2 and from user perspective, ranking of results is also of paramount interest. To address this constraint, the accuracies of SIFT and Perceptual Hash are compared in Figure 7 based on relevance ranked retrieval i.e. how many relevant results are ranked in top N results.

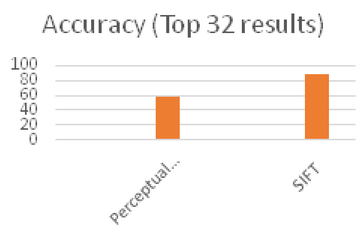

Fig. 7. Comparison of ranked retrieval

Figure 7 suggests that the accuracy drops significantly for perceptual hashing when considering top N results given that natural images usually have repetitive contents in them. Hence it is not suitable as a stand alone tool for the image plagiarism detection task. Moreover, the time taken by both the techniques is very high for an interactive application. Also noting that top N (N=60) results give far better accuracy, an obvious approach is to look for a technique which would be suitable for minimizing the time taken for such retrieval. Approximate nearest neighbor search [33] techniques have shown potential in optimizing such problems and are discussed next.

D. Performance of Hierarchical Approximate Matching and Indexing

Based on analysis and discussions in the previous section, the accuracy obtained by the hierarchical approximate matching and indexing is outlined in the following subsections.

a) Accuracy: For top-60 results an accuracy of nearly 70% (Figure 8) was observed which increases to 81.2% when top 100 results are considered. This increase can be attributed to the fact that there might be images similar to the query images which were not able to make it to the top-60 list.

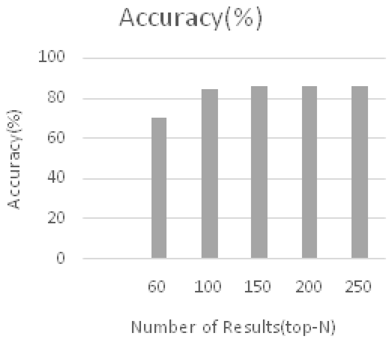

Fig. 8. Accuracy for Approach 2

b) Ranked vs Non-Ranked accuracies: The top-32 results were evaluated by ranking the results as returned by LSH (from Perceptual hash) and also by ranking by matching SIFT features using the Bag of Visual Words approach.

As observed from Figure 9, the ranking using Bag of Visual words does help in ranking the relevant results at the top of the list.

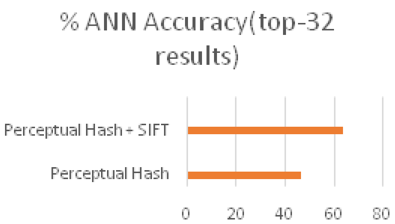

Fig. 9. Ranked V/s Non Ranked Retrieval

c) Retrieval Time: The most important improvement is observed in the time taken for such retrieval. As shown in Figure 10, this approach takes 10 folds lesser time than approach 1 and the increase in time is nearly linear with increasing number of candidates requested.

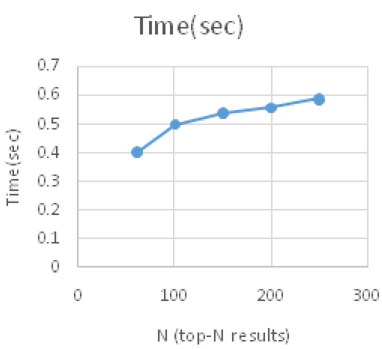

Fig. 10. Time vs Number of results

d) Scalability: To test the scalability of the approach, the query retrieval time for top-60 results was calculated by varying the number of images in the dataset.

As can be observed in Figure 11, the slope of increase of time taken is nearly constant as the number of images increase. Hence, the algorithm should be scalable to higher number of images as well. But a challenge with very large datasets is not only the retrieval time but also maintaining the relevance of retrieved results as well as the general techniques that seem to work on moderately sized datasets behave entirely differently on very large scale datasets [34] [35].

Fig. 11. Time vs Number of Images in the dataset

1) Approach Using Lucene: Another scope for optimization was observed from the fact that matching of histograms is being done by retrieving them from the database. Since, Lucene is essentially designed for document indexing, each image histogram was indexed as a separate document in the Lucene index [36]. So instead of exactly matching the histograms, another layer of approximate search has been added.

a) Accuracy: The accuracy for top-60 results using Lucene was found to be 68% which is slightly less(2%) than exact matching of histograms as shown before. This as stated above is due to the additional layer of approximate search.

b) Retrieval Time: Some improvement was observed in the retrieval time for varying the dataset size as shown in Figure 12 where we reproduce the results from Figure 10 for comparison. This improvement can be attributed to the database connection time and access time.

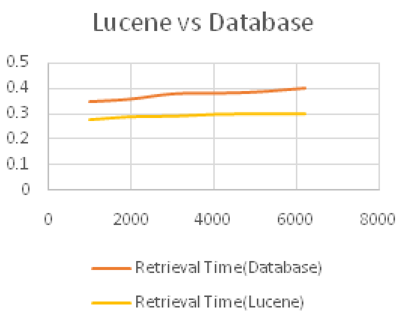

Fig. 12. Lucene v/s Database Retrieval time

IV. Conclusion

The work evaluates various stages of a CBIR system with empirical analysis for choosing the relevant techniques for the task of image plagiarism. A method for CBIR is proposed, consisting of perceptual hashing and SIFT with hierarchical approximate matching scheme. This scheme did result in a slight loss of accuracy but the reduction in retrieval time as compared to tf-idf approach without LSH is approximately 10 fold. It is shown that this approach can be scaled to large scale dataset but a comprehensive analysis would be necessary before claiming the same for very large scale datasets. The approach can be further improved by utilizing variations of these techniques such as Kernelized LSH for indexing or fast matching of binary features [37], Hierarchical k-means for clustering, or using multiple clusters of machine and utilizing map-reduce for distributive processing. The results clearly indicate that choice of a technique depends upon the tradeoff between the accuracy and time constraints.

References

[1] J. P. Yoon and Z. Chen, "Localization and detection of vector logo image plagiarism," in *Digital Forensics and Cyber Crime*. Springer, 2010, pp. 43–58.

[2] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, 1999.

[3] H. D. Tagare, C. C. Jaffe, and J. Duncan, "Medical image databases a content-based retrieval approach," *Journal of the American Medical Informatics Association*, vol. 4, no. 3, pp. 184–198, 1997.

[4] P. Desai, J. Pujari, N. Ayachit, and V. K. Prasad, "Classification of archaeological monuments for different art forms with an application to cbir," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. IEEE, 2013, pp. 1108–1112.

[5] E. Chalom, E. Asa, and E. Biton, "Measuring image similarity: an overview of some useful applications," *Instrumentation & Measurement Magazine, IEEE*, vol. 16, no. 1, pp. 24–28, 2013.

[6] J. M. Banda, M. A. Schuh, T. Wylie, P. McInerney, and R. A. Angryk, "When too similar is bad: A practical example of the solar dynamics observatory content-based image-retrieval system," in *New Trends in Databases and Information Systems*. Springer, 2014, pp. 87–95.

[7] C. Beecks, M. S. Uysal, and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1552–1557.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[10] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–506.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2564–2571.

[12] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.

[13] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 214–227.

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.

[16] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[17] C. Zauner, M. Steinebach, and E. Hermann, "Rihamark: perceptual image hash benchmarking," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, pp. 78 800X–78 800X.

[18] L. Wang, X. Jiang, S. Lian, D. Hu, and D. Ye, "Image authentication based on perceptual hash using gabor filters," *Soft Computing*, vol. 15, no. 3, pp. 493–504, 2011.

[19] D. Hu, B. Su, S. Zheng, and Z. Zhang, "Secure architecture and protocols for robust perceptual hashing," in *Computational Intelligence and Security (CIS), 2013 9th International Conference on*. IEEE, 2013, pp. 550–554.

[20] H. L. Brunk, K. L. Levy, and J. Tian, "Assessing quality of service using digital watermark information," Mar. 6 2012, uS Patent 8,130,811.

[21] V. Voronin, V. Frantc, V. Marchuk, and K. Egiazarian, "Fast texture and structure image reconstruction using the perceptual hash," *Image Processing: Algorithms and Systems XI*, 2013.

[22] A. Kumar, A. Anand, A. Akella, A. Balachandran, V. Sekar, and S. Seshan, "Flexible multimedia content retrieval using infonames," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 455–456, 2011.

[23] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 459–468.

[24] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *VLDB*, vol. 99, 1999, pp. 518–529.

[25] A. Andoni and P. Indyk, "E2lsh: Exact euclidean locality-sensitive hashing, 2004," *Implementation available at http://web. mit. edu/andoni/www/LSH/index. html*.

[26] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2130–2137.

[27] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe lsh: efficient indexing for high-dimensional similarity search," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 950–961.

[28] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration." in *VISAPP (1)*, 2009, pp. 331–340.

[29] V. Christlein, C. Riess, J. Jordan, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 6, pp. 1841–1854, 2012.

[30] C. Zauner, "Implementation and benchmarking of perceptual image hash functions," *Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus*, vol. 43, 2010.

[31] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[32] N. Y. Khan, B. McCane, and G. Wyvill, "Sift and surf performance evaluation against various image deformations on benchmark dataset," in *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*. IEEE, 2011, pp. 501–506.

[33] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[34] L. Bottou and O. Bousquet, "13 the tradeoffs of large-scale learning," *Optimization for Machine Learning*, p. 351, 2011.

[35] O. Chum and J. Matas, "Large-scale discovery of spatially related images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 371–377, 2010.

[36] M. Lux and S. A. Chatzichristofis, "Lire: lucene image retrieval: an extensible java cbir library," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 1085–1088.

[37] M. Muja and D. G. Lowe, "Fast matching of binary features," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*. IEEE, 2012, pp. 404–410.