Image Retrieval for Image Theft Detection

Ondřej Horáček[1], Jakub Bican[1], Jan Kamenický[1], and Jan Flusser[1]

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
{horacek,bican,kamenik,flusser}@utia.cas.cz

Summary. Image retrieval deals with a problem of finding similar pictures in image database. Our task is to find originals of modified images, typically stolen and republished on the web. Our problem is specific in terms of the database size (millions of photos), demanded speed of the search (seconds), and unknown image modifications (loss of quality, radiometric degradation, crop, etc.). Proposed method works in the following tree steps: 1. Image preprocess – normalization for robustness to the modifications. 2. Retrieval of candidates from the database index – stochastic decision in each vertex of the index tree is used to find the most relevant candidates. 3. Verification of the candidates – modified phase correlation is used. The method was implemented in practice with very good results. Based on wide experiments, it was shown that the success rate of the search depends on the level of image modification.

1 Introduction

Large image databases are often run on a commercial basis – browsing through and viewing images is free of charge while downloading and re-using them on your webpages and articles is a subject of a fee. However, some users republish the downloaded images without paying the fee, which is a violation of copyright law. The copyright owner thus wants to regularly scan suspicious domains or websites to check if there are any unauthorized copies of the database images.

This paper describes an original method which we developed for an international advertising and press company. This company runs a database of more than 10 millions photographs updated everyday. They estimate hundreds thousands images being used without permission on the web. Detection of illegal copies is complicated by two principal difficulties – the unauthorized images are usually modified before they are post on the web and the response of the system must be extremely fast because of an enormous number of database images. Although this problem formulation looks like an image retrieval task, this is not the case. In traditional image retrieval, we want to find in the database all *similar* images to the query image, where similarity

is evaluated by colors, textures, content, etc. Here we want to identify only the *equivalent* images to the query (we call this task *image identification*. This is why we cannot apply most of standard image retrieval techniques. By the term "equivalent images" we understand any pair of images which differ from one another by the following transformations.

- Quality reduction. Either compression to different image format or resize changes the image representation, although the image seem very similar to human eye (in Fig. 1b).
- Radiometric and color distortions. We consider changes of image brightness and contrast (in Fig. 1c), changes of color tone or conversion of the image to gray-scale (in Fig. 1d).
- Crop of the image. Image part can be cropped from the background, still we consider that major part of the image is preserved. Also, a frame can be added to the image or aspect ratio can be changed (in Fig. 1e).
- Local changes. A logo can be added to the image, or a thin label can go throw the image (in Fig. 1f).
- Combinations. Reasonable combination of distortions mentioned above is also considered. However, their increasing amount and significance will surely impact the algorithm results (in Fig. 1g).

(a) (b) (c) (d) (e) (f) (g)

Fig. 1. Possible modifications of the query image. The first image is the original stored in the database.

We decided to include the above transforms into our "equivalence relation" because, according to the earlier statistics performed by the company, they are frequently present in unauthorized copies.

It is not possible to use directly any of the existing methods. Probably the closest published method is by Obdržálek et al. in [5]. It is suitable for recognition of man-made objects with partially planar surface, such as goods in a supermarket. It can handle very general situations including partial occlusion and affine transform of the objects but this is useless for our purpose. Our method is different is several aspects. We can not use several maximum extremal regions per image, because on our database images they may not exist. Our index tree is thousand times bigger than that one considered in [5],

so we need to use different, space-efficient tree structure. Finally, we include another independent image comparison to eliminate false positives matches.

We present an original image identification method, which is based on a hierarchical structure of the database, representation of the images by proper invariant features, and a fast tree-searching algorithm. As demonstrated, our method can achieve very good identification rate in a reasonable response time. In this article, we present main idea of the method as well as selected details, as the scope allow us.

2 Algorithm outline

A kind of binary decision tree is used for the database indexing. For the indexing, some image features are needed. From them, we require robustness to considered degradations, stability, but mainly to be extensible and discriminate enough for any database size. We use image intensities in various pixel positions, surely after dealing with the image distortions. The choice is simple in principle, but we found it effective in presented method. Our image identification works in these three steps:

1. Normalization. Robustness to the modification is ensured by normalization of processed images. In other words: Each of the considered modification corresponds with a change of an evaluative image quality. During the image preprocessing, we apply the modification once again in order to set the qualities to the same value for all the images. This process annuls the impact of considered modifications.
2. Stochastic index. The database images are organized in binary decision tree for fast image identification. Decisions in the tree are based simply on image intensity at certain position. The position and threshold are set for each vertex during build of tree. For individual query image, stability of decisions in the tree is evaluated. We alternate the unstable decisions during the image identification. So, we get many candidates per query.
3. Candidate verification. Edge information of the images serves for the the final comparison of a candidate with the query image. More concretely, phase correlation restricted to low-pass fourier transform is used.

3 Normalization

Both the query image and the database images are preprocessed previously. We apply some normalizations to make the images invariant to considered modifications. To handle the radiometric and color distortions, we convert the images to gray-scale. Then, the lowest and the highest 10 percent image histogram fraction are found and their centers of gravity are set to fixed values

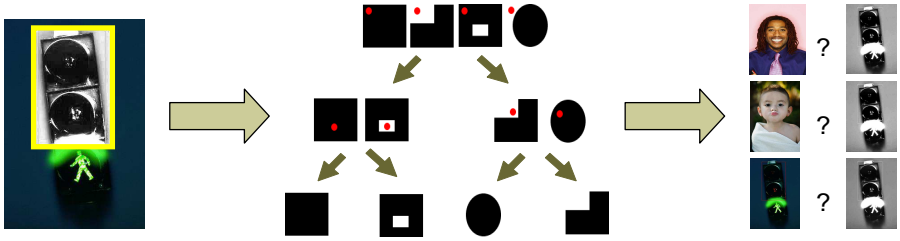

Fig. 2. Schema of the image identification. First, normalization of the query image is done. Then, the most probable candidates are found in the index tree. And finally, the candidates are verified by modified phase correlation.

(by addition / multiplication of the image intensity values). This ensures image invariance to brightness and contrast changes.

The image crop is tougher modification to handle. The first idea could be to use features invariant to crop, such as corners (found for example by Harris corner detector [1]) or intersection points. We do not use them because it is not possible to stably match those points for all the database images after crop. We consider that the crop preserves major part of the image. For most of the images, the part is separable from the background by color. So, we segment the major part of the image and bound it by a crop invariant frame.

We introduce a special color-based segmentation for separation of the image major part. It was developed heuristically, we gave respect to experimental results on the image database. The algorithm segments stable, color-specific, big enough parts of the image interior. In principle, it divides the image into blocks, computes the block color character, and finds neighborhood blocks with the same character for each possible starting block. The segmented region is broadly bounded by box, which we call a frame.

Stability and "quality" of a the frame is evaluated. For the database images, just area bounded by the best frame is used for image identification in the index tree. It is reasonable to consider that this frame will be found in the modified image as one of the best, too. Therefore we use the best five variants of the frame for the image search.

4 Stochastic index

It is the task for the index to retrieve image from the database quickly. Time of the response must be principally faster than proportional to the database size, which contains millions of images. Query images have been normalized and blurred the same way as the database images, so they should be very similar. Still, we evaluate stability of each decision in the index tree to be robust to minor image changes.

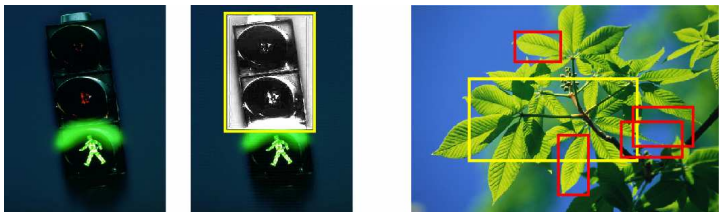

Fig. 3. Image preprocess. The original is converted to gray-scale, crop-invariant frame is found and its brightness and contrast are normalized. On the right, multiple invariant frames used for the image identification are shown.

The database images are organized in a binary decision tree, commonly used to handle huge amount of data (a survey is done in [4]). Decision in the tree are based on intensities of image pixels. The pixel (threshold) position is taken relatively to the valid image area (frame). Once we have two different branches of the tree, each of them can contain images differing mainly (and therefore the most stably) in a different image part. Such a threshold position has been established during the tree build. So, each node of the tree contains threshold value and relative position of the threshold pixel.

Now we describe the search for image in the index tree. At a node, pixel intensity of the query image at position of the threshold is compared with the threshold value and its sub-branch (next node) is chosen. We establish stability of this decision as a likelihood, that the image belong to the same (left/right) branch even after image distortions. We assume that the threshold pixel can change its intensity (with uniform distribution in certain intensity interval) and it can change its position (the miss-place has a two-dimensional gaussian distribution) (in Fig. 4). This stochastic model is similar to the one presented by Obdržálek [5], but the search algorithm is different.

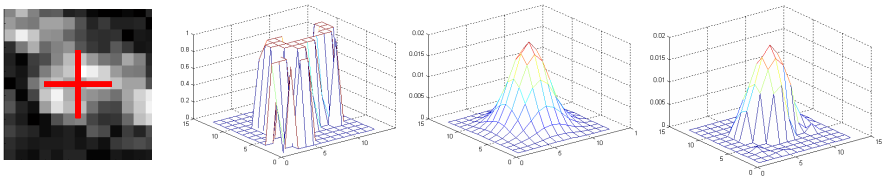

Fig. 4. Stability of decision in the tree. The image and the threshold position, likelihood that a pixel is above the threshold, likelihood of threshold position change, and their multiplication.

This way we go throw the index tree till we reach a leaf containing an image – candidate for the match. Now, we select node with the least stable decision and alternate it. So, we continue throw the other backtrack and get

next match candidate. This way we found 20 images from the index with the highes probability to match the query image.

It was mentioned above that the index tree needs to be prepared previously. Because of the huge number of images, the tree is builded gradually. The principle of used algorithm is as follows:

1. Read normalized version of a database image from a disk one by one.
2. Find a leaf in the partially builded tree for the image and register the image to the leaf.
3. If the leaf contains enough images, convert the leaf to inner tree node and divide its images as follows:
   a) Choose randomly several possible positions of a threshold.
   b) For each threshold candidate, get intensity values for of the images belonging to the leaf (at the processed threshold position).
   c) For the threshold candidate, compute the threshold value as median of the intensity values.
   d) For the threshold candidate, compute its stability as a sum of decision stabilities for all the images belonging to the leaf. It means, evaluate the likelihood described in the paragraph about search (and Fig. 4).
   e) Choose the most stable threshold and save its position and the processed node (leaf).
   f) Create left and right sub-node of the processed node. Based on the new threshold, divide the images of the processed node to them.
4. After all the database images have been processed by previous steps and are registered to some node, divide the nodes till they contain only one image. (Do it the same way as in steps 3a to 3f).

5 Rest of the method

In the last step, the candidate images are compared one by one with the query image. We use modified phase correlation (originally introduced by Kuglin [3] in 1975). Phase correlation is robust to overlap, shift and radiometric degradation. We restrict the correlation only to low-pass of the fourier spectrum to make the comparison more stable for image quality changes (in Fig. 5).

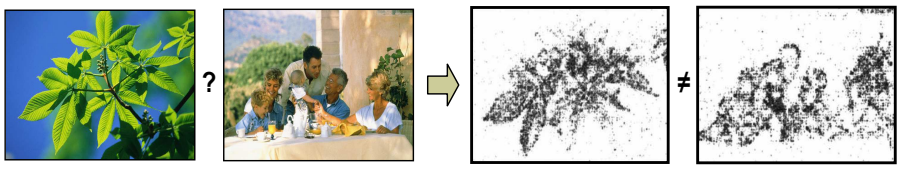

Fig. 5. Candidate verification. Low-pass of phase correlation bases the image comparison on their major edges.

For better performance in practical experiments, many improvements have been made on the basic method scope described above. It is clear, that the algorithm use several parameters and options, that affect both the identification rate and algorithm speed. Further more, the more normalization we use, the more image information is lost and it is harder to identify the original image. On the other hand, heavy normalization during the preprocess is necessary to find the original for images with several distortions. For each used version of the preprocess, we need to build different index tree to search in. We use two independent index trees in the prototype implementation: one is build from images with normalized histogram, the second is build and searches for the image parts bounded by the invariant frames.

The method is being implemented as a configurable framework. A structured configuration file controls application of zero to several normalization algorithms per image, build and search in several index trees as well as all parameters for all the algorithms. (Note, that some simple preprocess algorithms, such as mirroring or blurring, are not described in this article). We have also optionally improved the index tree ability to distinguish images depending on their color. The single threshold value was replaced by a plain in the RGB space. The plain is set to divide the RGB pixels to two groups stably. Well known principal component analysis is used to establish the plain orientation (for a PCA review, see book see a book [2] by Jolliffe).

6 Experiment results

Prototype of the proposed method was implemented in Matlab. Tests have been done on 100 000 image database. First, a thousand of query images was generated for each considered modification (the images are still "equivalent", see samples on Fig. 1). Strength of the modification is varying around the level expected from the practice. The identification ratio is very good, better than we expected. Original images are found successfully in 99.5 % of cases, in the rest 0.5 % cases a database contained a very-similar image (e.g. with some retouching), that has been identified before the original one. For typical automatically republished images, identification rate is more than 90 %, which is very good for practical use of the method. The identification rate surely decrease for harder modifications, but even for combinations of radiometric degradations, crop and logo is still about 20%. See table 1 for more details.

Response speed of the Matlab prototype is up to 20 seconds per image. Image retrieval from the index tree takes about 0.2 second, the one by one candidate verification is more time consuming due to the fourier transform. Build of the index tree takes less than second per image, but after some non-trivial normalization (invariant frame takes about 3 seconds per image) the database build takes several days. The method identification rate and response time depends on appropriate set of parameters. There is a trade-off between

Degradation	True positives	False positives
Original	99.5 %	0.4 %
Logo added	98.2 %	0.2 %
Scale	94.6 %	0.4 %
Brightness and contrast	71.2 %	0.7 %
Crop	45.0 %	0.2 %
Scale + logo	93.4 %	0.2 %
Scale + logo + frame	35.8 %	0.4 %
Radiometric deg. + crop + logo	18.2 %	0.5 %
Not in the database	0.0 %	0.3 %

Table 1. identification rate on 100000–image database. The identification rate depends on kind of image modification.

speed, robustness to image distortion, identification rate, and false positives rate.

7 Conclusion

In this article, we presented our method for modified image identification. The task is specific by character of the modifications, the database size, and required response speed. The method is novel in normalization during the image preprocessing (invariant frame, normalization) and in stochastic backtracking throw the image index. It was shown in experiments on huge database that the method performs very well. It is ready to catch majority of illegally republished database images on scanned web sites.

References

1. C. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Conference*, pages 147–152, 1988.
2. I. T. Jolliffe. Principal component analysis. In *Principal Component Analysis*. Springer Verlag, New York, 1986.
3. C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. *Assorted Conferences and Workshops*, pages 163–165, September 1975.
4. Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Min. Knowl. Discov*, 2(4):345–389, 1998.
5. Štěpán Obdržálek and Jiří Matas. Sub-linear indexing for large scale object recognition. In WF Clocksin, AW Fitzgibbon, and PHS Torr, editors, *BMVC 2005: Proceedings of the 16th British Machine Vision Conference*, volume 1, pages 1–10, London, UK, September 2005. BMVA.