

Speeding up the Detection of Line Drawings Using a Hash Table

Weihan Sun¹, Koichi Kise²

Graduate School of Engineering, Osaka Prefecture University, Japan
¹ sunweihan@m.cs.osakafu-u.ac.jp, ² kise@cs.osakafu-u.ac.jp

Abstract: Line drawings, such as comics, marks and logos, are an important part of image publications, which need copyright protection. Since line drawings are composed by lines, it is easy to make a similar copy by handwriting. Therefore, in the case of line drawings, copyright protection method should also be available for handwritten copies. Although a method applicable to handwritten copy detection has been proposed, its processing is time-consuming since it is based on matching a large number of local features extracted from images. In this paper, we apply a hash based method to speed up the detection of line drawings. Compared to the previous method, the detection time per image reduces from 6,013 ms to 75 ms with almost the same detection rate. Furthermore, we have also proved that the proposed method has effectiveness for detection with a larger database.

Key Words: Copyright protection, Detection of line drawings, Image retrieval, Speed up

1. INTRODUCTION

Line drawings are a type of images that consist of distinct straight and curved lines in monochrome or few colors placed against plain backgrounds. Such as comics, graphs and logos, line drawings are widely used in various image publications. Since development of scanners and digital cameras offer great facilities for duplicating images, the problem of illegal copies catches more peoples' attentions. Because line drawings are valuable for designs and other creative activities, the copyright of line drawings is threaten by various illegal copies.

For image copyright protection, digital watermarking is a commonly used method. By the processing of digital watermarking, copyright information is embedded into images. If the images are copied, the information is also carried in the copies. To preserve the value of images, digital watermarks are invisibly embedded into images by using redundancy of color information. However, because line drawings contain less color information, it is difficult to apply digital watermark in an unperceived way.

Image retrieval is another method to protect image copyright. By detecting similar images from the copyrighted image database, illegal copies are detected. Because local features are stable during most of the image transformations, they are often used for image retrieval. Mikolajczyk et al. [1] have compared various kinds of local features and proved that the SIFT (Scale-Invariant Feature Transform) [2] based descriptors perform best for image retrieval. Y. Ke et al. have proposed the PCA-SIFT (Principal Component Analysis SIFT) [3] for near-duplicate detection of images [4].

However, since line drawings are composed by lines, handwriting can also be applied to produce illegal copies. Therefore, for protecting the copyright of line drawings, the detection method should be applicable not only to printed line drawings but also to handwritten ones. Sun et al. [5] have

proved that the method using SIFT local features loses its effectiveness on handwritten line drawings, and proposed applying MSER (Maximally Stable Extremal Regions) [6] and HOG (Histograms of Oriented Gradients) [7] to describe features of line drawings. By using the matching of HOG feature vectors they have achieved the detection for both printed and handwritten partial copies of line drawings from complex backgrounds. However, the detection of Sun's method is time-consuming, which prevent it from being applied in practice.

Hash tables are an ordinary method for increasing the searching speed. Such as LSH (Locality-sensitive Hashing) [8], it utilizes multiple hash tables to do approximate nearest neighbor search. Kise et al. [9] have proposed applying a single hash table for approximate search based on PCA-SIFT feature vectors to reduce the burden on memory.

In this paper, we propose to introduce a hash table to speed up the detection of Sun's method. We have experimentally proved that the proposed method can increase the detection speed 80 times without reduction of detection rates, and the method is also applicable to a larger database.

2. OVERVIEW OF THE PREVIOUS METHOD

2.1 Local feature matching

The process of copyright protection using image retrieval is shown in Fig. 1. The method can be divided into two parts: database processing and query processing. In the part of database processing, copyright images are collected for building the feature vector database. By using the region detector and the feature descriptor, database feature vectors are extracted from copyright images. With the image label, each database feature vector is stored in the feature vector database. In the query processing, illegal copies are treated as queries. Through the same

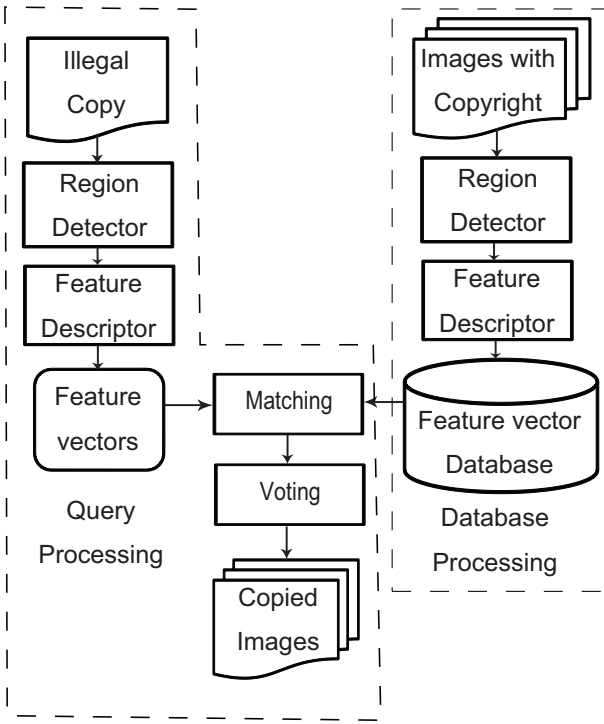


Fig. 1: Processing of image retrieval.

region detector and the feature descriptor, query feature vectors are extracted. By finding the nearest database feature vectors and voting for each copyright image label, we can identify copied images.

2.2 Region detector and Feature descriptor

To obtain stable feature vectors from handwritten line drawings, Sun et al. use MSER (Maximally Stable Extremal Region) [6] as a region detector and HOG (Histograms of Oriented Gradient) [7] as a feature descriptor.

MSERs are the maximally stable regions, in which the intensities are extremal. By diagonalizing the covariance matrix of MSERs, MSERs turn to be ellipses, as shown in Fig. 2. To make the local feature regions contain more line information and invariant to rotation, elliptical MSERs are magnified and rotated as the long axis of ellipse parallel to the y axis of the image. Finally, as shown in Fig. 3(c), the elliptical region is treated as one local feature region.

For each local feature region, HOG is applied to extracted a feature vector. As shown in Fig. 4, by calculating the gradient direction histogram based on the gradient strength for each cell, cell feature vectors are extracted. The HOG feature vector consists of overlapped block vectors and block vectors are composed by cell vectors. In Sun's method, the gradient directions are quantized into 9 bins. There are 8×8 cells for each local feature region and 3×3 cells for each block. Therefore, the HOG feature vectors have $9 \times 3 \times 3 \times 6 \times 6 = 2916$ dimensions.

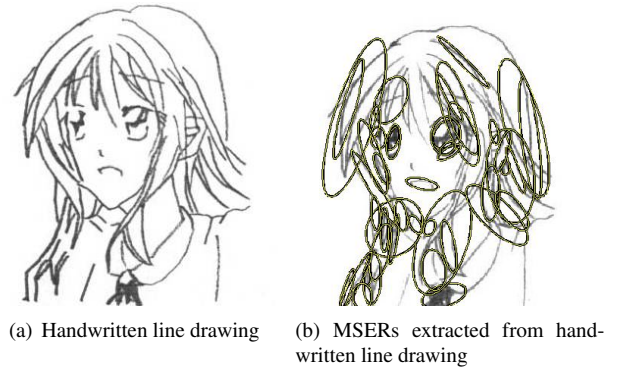


Fig. 2: MSERs from line drawings.

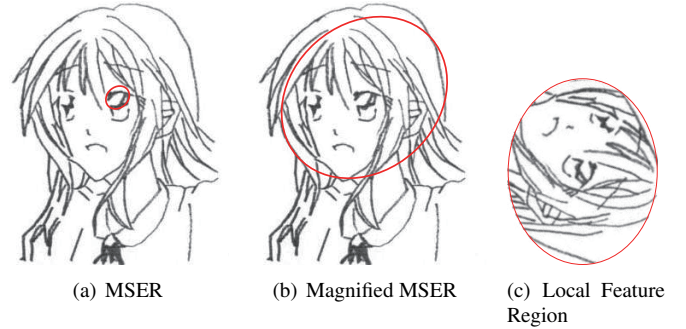


Fig. 3: Processing from MSER to Local Feature Region.

2.3 Matching and Voting

To match similar feature vectors, Sun utilized evaluation of Euclidean distance and applied ANN (Approximate Nearest Neighbor) [10] to increase the searching speed. ANN is a method of nearest neighbor search by using a tree structure. By shrinking the searching range at the rate of $1/(1 + \epsilon)$, ANN achieves reduction of the searching time. As shown in Fig. 5, q is a query vector, and p_1, p_2, p_3 are the feature vectors near q . We can see that ANN may miss the true nearest feature vector by applying approximation factor ϵ .

Through the voting of matched vectors for each copyright image, a list of images ranked by the similarity are reported as candidates.

3. PROPOSED METHOD

Since Sun et al. applied feature vectors of 2,916 dimensions to describe handwritten line drawings, the distance calculation between vectors is time-consuming. Therefore, we propose using a hash table to speed up the detection of Sun's method.

The basic idea is to use the most distinctive dimensions of local feature vectors for indexing, and do a hash search by considering the perturbation of these dimensions to image translations.

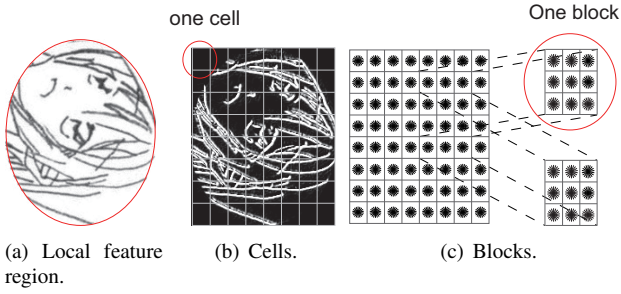


Fig. 4: Region instruction of HOG features.

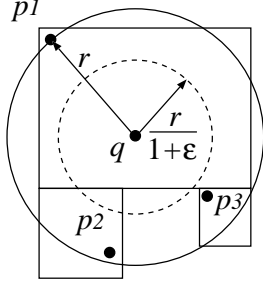


Fig. 5: Searching process by ANN.

First, we apply PCA (Principal Component Analysis) [3], which could reveal the internal structure of the data by the calculation of the eigenvalue decomposition, to refine local feature vectors of the previous method. We built the eigenspace by using 60,000 training feature vectors extracted from line drawings, which are outside the database for experiments, and retain the top 100 principal components. Therefore, we refine the HOG feature vectors to the feature vectors with 100 dimensions.

For indexing feature vectors in the hash table, we binarize each dimension by the following function:

$$u_j = \begin{cases} 1 & p_j - \mu_j \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{p} = (p_1, p_2, \dots, p_n)$ is a feature vector. μ_j is median of p_j . The first $d (\leq n)$ elements of the bit vector $\mathbf{u} = (u_1, \dots, u_d)$ are employed for the indexing by the following hash function.

$$H_{\text{index}} = \sum_{i=1}^d u_i 2^{(i-1)}$$

In the proposed method, the size of hash table is set to 2^d , which means the first d dimensions of bit vectors are utilized for the hash function. d is set less than the dimension of \mathbf{p} ($= 100$).

To reduce the memory consumed for storing feature vectors, we apply scalar quantization to feature vectors [11]. As shown in Fig. 6, by dividing the value histogram of each dimension into regions with the same size, we quantize the vectors to 2 bit/dim.

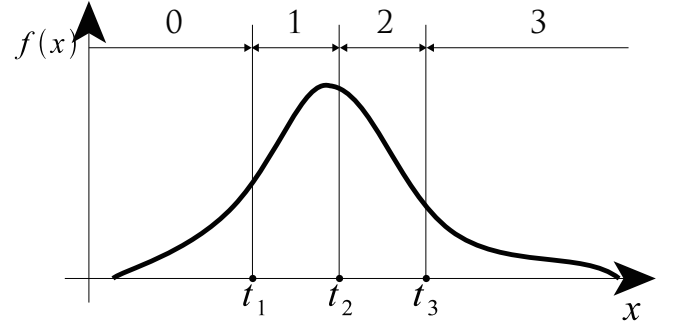


Fig. 6: Scalar quantization.

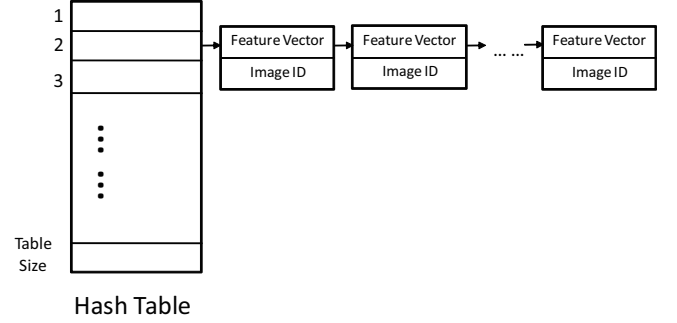


Fig. 7: Hash Table.

After scalar quantization, feature vectors with their image IDs are stored at a corresponding bin by chaining as shown in Fig. 7. If the length of the chain exceeds a threshold c , the chain is deleted. By these processing, we store all feature vectors into the hash table.

In the process of detection, considering the changes of query vectors caused by handwriting, we introduce perturbation of query feature vectors to simulate errors. Specifically, if the dimension q_j satisfying $|q_j - \mu_j| \leq \epsilon$ in the query vector $\mathbf{q} = (q_1, \dots, q_n)$, the other bit of dimension $u'_j = 1 - u_j$ is also utilized to generate bit vectors. The limited number of expanded bit vectors is set to be 2^b . By these expanded bit vectors, we apply the same hash function to find corresponding bin in the hash table. Then, the image ID with the feature vector which is nearest to the query feature vector receives one vote. By voting of matched feature vectors, the top k images are reported as candidates.

4. EXPERIMENTAL RESULTS

4.1. Condition

We utilized images of comics (about $650 \times 1,000$ pixels) for building a database. For the query images, we made printed and handwritten partial copies (100 for each) by cropping parts (300×300 pixels) from images in the database, and copied

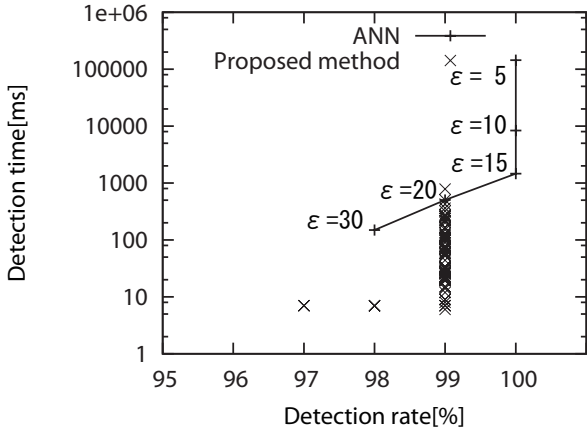


Fig. 8: Results for printed copies.

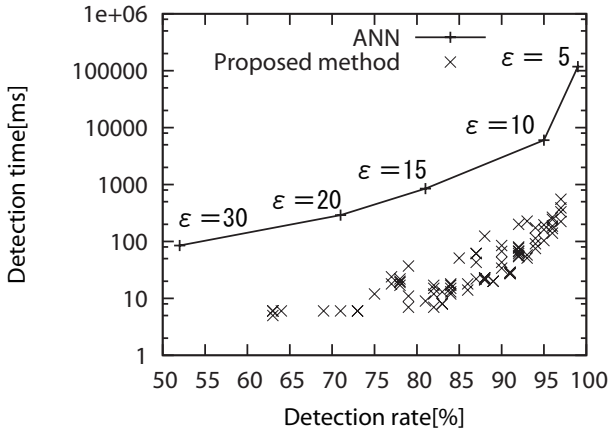


Fig. 9: Results for handwritten copies.

them by handwriting and a printer. We also embedded these partial copies into unknown backgrounds with the size 10 times as large as the partial copies. All the experiments were performed using a computer with AMD Opteron 2.8GHz CPU and 32GB RAM. The cumulative classification rate of top 5 retrieved images is treated as detection rate, and detection time is average detection time (not including the extraction time of feature vectors) per image.

4.2. Experiment 1

First, we compared the proposed method with Sun’s method. We built a database of 1,000 line drawings, and utilized the handwritten partial copies and the printed partial copies without backgrounds as queries. We applied different ε to Sun’s method and different b, e, d ($b = 10, 13, 15, 17, e = 0.3, 0.4, 0.5, 0.6, d = 20, 22, 24, 26, 28$) of the proposed method. The results are shown in Figs. 8 and 9.

From the results, we can see that Sun’s method can achieve the detection of printed line drawings at 100% detection rate and 99% for handwritten ones without considering the detection

Table 1: Detection results of printed partial copies.

	no background	10x backgrounds
Detection rate	99%	98%
Detection time	930 ms	15,688 ms

Table 2: Detection results of handwritten partial copies.

	no background	10x backgrounds
Detection rate	93%	80%
Detection time	494 ms	8,846 ms

time. On the other hand, by adjusting the parameters b, e, d , the proposed method can achieve 99% detection rate for printed line drawings and 97% for handwritten ones, and consumes much less detection time than Sun’s method for the same detection rate. Also, the proposed method obtained a better detection rate than Sun’s method with the same detection time. Considering both the detection rate and the detection time, we have set $e = 0.5, b = 15, d = 24$ experimentally, and have obtained a result of 94% detection rate for handwritten line drawings and 99% for printed ones in 75 ms per query. Compared to results of Sun’s method which takes 6,013 ms per image with 95% detection rate for handwritten copies and 100% for printed copies, detection by the proposed method is 80 times faster than the previous method with almost the same detection rate.

4.3. Experiment 2

In this experiment, we tested scalability of the proposed method by applying a database with 11,600 comic pages. We applied two kinds of queries: (1) printed and handwritten partial copies for Experiment 1, (2) partial copies with unknown backgrounds with the size 10 times as large as the partial copies. The parameters were set to $e = 0.5, b = 15, d = 24$.

As shown in Tables 1 and 2, the proposed method can achieve a detection rate of 99% for printed line drawings in 930 ms and 93% for handwritten ones in 494 ms. Even embedding into the large backgrounds, the partial copies can also be detected with 98% detection rate for printed line drawings and 80% for handwritten ones. As shown in Fig. 10, the handwritten partial copy was detected from the background by local feature matching.

5. CONCLUSION

In this paper, we focus on the detection of line drawings for their copyright protection, and have proposed a method using a hash table to speed up the detection. Compared to the previous method, the proposed method achieves faster detection with the same detection rate as before. We have also proved that the proposed method is scalable by experiments based on a database of 11,600 comic pages.

Future work is to further improve the detection rate under a wider variety of disturbances.

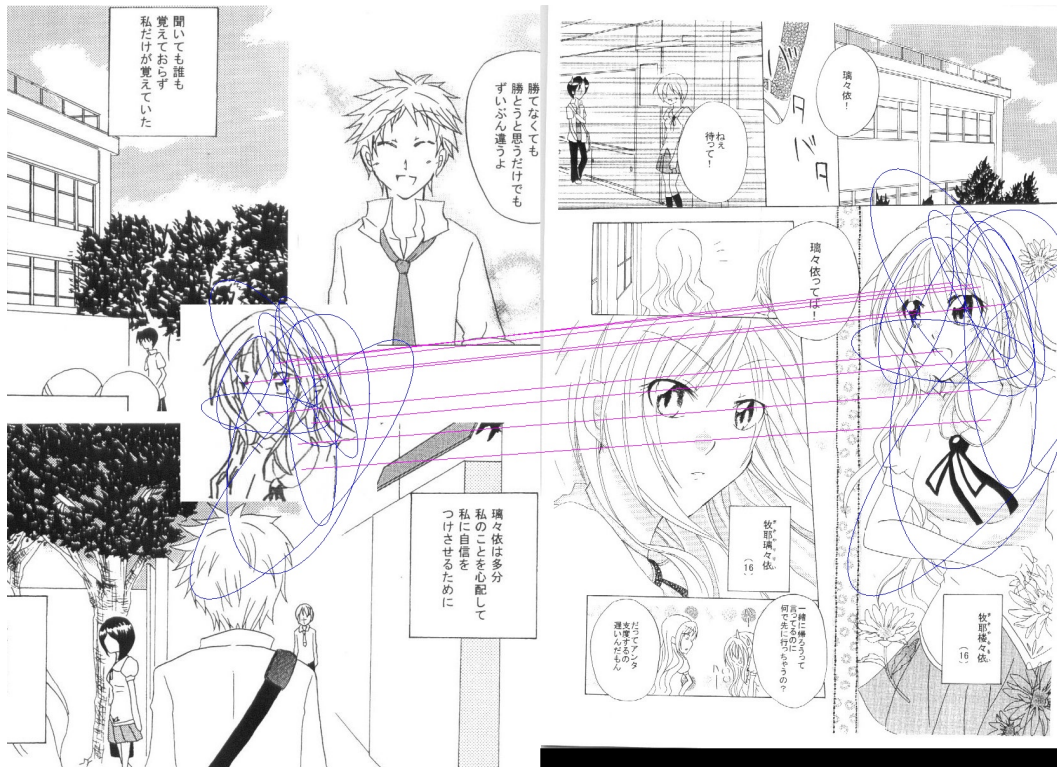


Fig. 10: Example of handwritten copies detected from a 10x background.

ACKNOWLEDGEMENTS

This work was supported in part by the Grant-in-Aid for Scientific Research (B)(19300062) from Japan Society for the Promotion of Science(JSPS).

References

- [1] K.Mikolajczyk, C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.27, No.10, pp.1615–1630, 2005.
- [2] D. G. Lowe. Distinctive image features from scale-invariant key-points. *International Journal of Computer Vision* , Vol.60, No.2, pp.91–110, 2004.
- [3] Y. Ke, R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *Proc. CVPR 2004*, 2004, Vol.2, pp.506–513.
- [4] Y. Ke, R.Sukthankar, L. Huston. Efficient near-duplicate detection and sub-image retrieval. *Proc. ACM Conf. on Multimedia 2004*, 2004, pp.869–876.
- [5] W. Sun, K. Kise. Detecting Printed and Handwritten Partial Copies of Line Drawings Embedded in Complex Backgrounds. *Proc. ICDAR 2009*, 2009, pp.341–345.
- [6] J. Matas, O. Chum, M. Urban, T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *Proc. BMVC 2002*, 2002, pp.384–393.
- [7] N. Dalal, B. Triggs. Histograms of Oriented Gradients for Human Detection. *Proc. CVPR 2005*, 2005, Vol.1, pp.88–893.
- [8] A. Andoni, N. Immorlica, P. Indyk, V. Mirrokni. Locality-sensitive Hashing Using Stable Distributions. *Nearest-Neighbor Methods in Learning and Vision*, The MIT Press, pp.61–72, 2005.
- [9] K. Kise, K. Noguchi, M. Iwamura. Simple Representation and Approximate Search of Feature Vectors for Large-Scale Object Recognition. *Proc. BMVC 2007*, 2007, Vol.1 pp.182–191.
- [10] S. Arya, D. Mount, R. Silverman, A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, Vol.45, No.6, pp.891–923, 1998.
- [11] Koichi Kise, Kazuto Noguchi, Masakazu Iwamura. Memory Efficient Recognition of Specific Objects with Local Features. *Proc. ICPR 2008*, 2008, WeAT3.1.