Open Journal of Applied Sciences, 2013, 3, 16-21
Published Online March 2013 (http://www.scirp.org/journal/ojapps)

Speed-up Multi-modal Near Duplicate Image Detection

Chunlei Yang[1,2], Jinye Peng[2], Jianping Fan[2]
[1]CSIRO Sustainable Ecosystems, Carmody Road, St. Lucia Queensland, Australia
[2]CSIRO Sustainable Ecosystems, Highett, Australia
Email: {*Andrew.higgins, Leonie.pearson, Luis.laredo}@csiro.au

Received 2012

ABSTRACT

Near-duplicate image detection is a necessary operation to refine image search results for efficient user exploration. The existences of large amounts of near duplicates require fast and accurate automatic near-duplicate detection methods. We have designed a coarse-to-fine near duplicate detection framework to speed-up the process and a multi-modal integration scheme for accurate detection. The duplicate pairs are detected with both global feature (partition based color histogram) and local feature (CPAM and SIFT Bag-of-Word model). The experiment results on large scale data set proved the effectiveness of the proposed design.

Keywords: Near-Duplicate Detection; Coarse-To-Fine Framework; Multi-Modal Feature Integration

1. Introduction

The existence of duplicate or near duplicate image pairs is universally observed in text-based image search engine return results (such as Google Image: the return results for a certain query word) or personal photo collection (such as Flickr or Picasa personal photo album: photos that are consecutively taken at the same location with slightly different shooting angle), as found in Figure 1. The existence of large quantity of near duplicate image pairs will cause big burden for any type of search engine based image exploration or query operation. Considering the scale of the search results, it is very difficult if not impossible to identify such near duplicate images manually. Thus it is very important to develop efficient and robust methods for detecting the near duplicate images automatically from large-scale image collections [1-3].

It would be rather convenient for near duplication detection tasks to utilize heterogeneous features like EXIF data from photos [4], or time duration information in video sequences [5]. In fact, such information is not available for most of the data collections which forces us to seek for solution from visual content of the images only. Among content-based approaches, many focus on the rapid identification of duplicate images with global signatures, which are able to handle almost identical image [6]. However, near duplicates with changes beyond color, lighting and editing artifacts can only be reliably detected through the use of more reliable local features [7-10]. Local point based methods, such as SIFT descriptor, have demonstrated impressive performance in a wide range of vision-related tasks, and are particularly suitable for detecting near-duplicate images having complex variations.

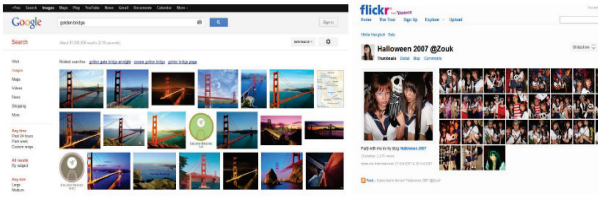

Figure 1. Left: Google Image search result for query "golden gate bridge"; right: Flickr search results for the query "Halloween". Both observes a number of near duplications

The potential for local approaches is unfortunately underscored by matching and scalability issues as discussed above. Past experience has guided us to seek for balance between efficiency and accuracy in near duplicate detection tasks. In order to speed up the duplicate detection process without sacrificing detection accuracy, we have designed a two-stage detection scheme: the first stage is to partition the image collection into multiple groups by using computational cheap global features; the second stage is to conduct image retrieval with computational expensive local features to extract the near-duplicate pairs. The output of the first stage is supposed to not separate any potential near duplicate pairs and the number of images participated in the second stage retrieval could be dramatically reduced. The visual presentation of the image used in the second stage is Bag-of-Word (BoW) model. We have conducted the interest point extraction operation to all the available images and represent each interest point with SIFT descrip-

tor. A universal code book is generated with k-means clustering algorithm from millions of such interest point descriptors. Each code word is a center of the k-means clustering result. In actually implementation, we have conducted hierarchical k-means to construct the code book. With vector quantization operation, each interest point in an image is mapped to a certain code word and the image can be represented by the histogram of the code words. For the purpose of interest point matching, we only count the matches by the points that fall into the same bin, thus the actual calculation of the histogram is not required. Besides the SIFT Bag-of-Word model, we also implement the CPAM (Color Pattern Appearance Model) feature which is built from YC_bC_r color space and quantized in a similar fashion as BoW model. We also built a universal code book for the CPAM feature with k-means clustering algorithm and each image is encoded with vector quantization technique. Finally, the detection result from both models will be combined together with our multi-modal integration design.

2.Near Duplicate Detection

Traditional methods often require n^2 pair-wise comparisons to detect duplicate image pairs in a group of n images. When we are dealing with large scale image set, often with n at a minimum of several hundred, the traditional methods could be very time consuming. An intuitive idea is to use computationally cheap image features to conduct the comparison, meanwhile, we do not want to sacrifice the statistical correctness to gain computation speed up. As a trade-off, we conducted image clustering Algorithm based on cheap visual features, such as global features, which can roughly partition the group of images without separating the duplicate pairs. Then the relatively expensive local features can be used for near duplicate detection on a pair-wise fashion within each cluster. An illustration of the proposed coarse-to-fine structure is shown in **Figure 2**.

We first clarify the difference between "duplicate" and "near duplicate" images.

- Duplicate: duplicate images are defined as image pairs that are only different on scale, color scheme or storage format.
- Near Duplicate: By relaxing the above restriction, we can define near duplicate images as duplicate pairs further varied by contrast, luminance, rotation, translation, a slight change of the layout or background.

The relaxation of the restriction of the near duplicate definition made the traditional Hash based method, which has been successfully applied for copy detection inapplicable. Considering computation efficiency and detection accuracy, a coarse-to-fine model was designed by integrating image clustering with pair-wise image matching.

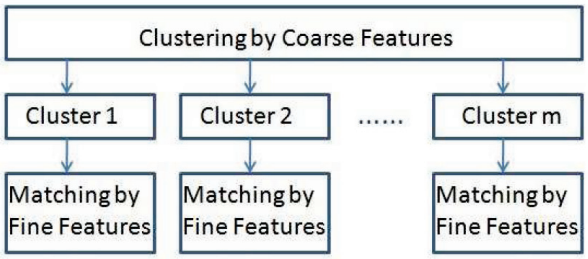

Figure 2. Coarse to fine cluster-based near duplicate detection framework.

As we have mentioned earlier, the proposed similarity detection model is sequentially partitioned into two steps: the first step is to cluster the images based on coarse features, which are usually cheap features in terms of computation cost; the second step is to conduct more complex features so as to more accurately detect duplicate image pairs within the clusters. The purpose for the first step is to roughly partition the image group while maintaining their duplication bonds within the clusters. In the second step, comparisons are conducted between image pairs in each cluster for near duplicate detection. We need more accurate object-oriented features, or in other words the local features, for the accurate detection step.

2.1. Global Approach

Object localization is very important for near duplicate detection. To utilize the global feature in near duplicate detection, the object has to be localized as accurate as possible. Meanwhile, to avoid the time consuming operation of accurate object localization, we just roughly partition the image into 5 segments and extract global color histogram feature from each of them and then concatenate the features to form our global feature.

Specifically, the features are extracted on a partition-based framework, rather than image-based or segment-based framework, as shown in Figure 3. We choose the partition-based framework based on the following consideration: a) image-based framework is not accurate enough for representing object images; b) segment-based framework is too computationally expensive and may sometimes fall into the over segmentation trouble; c) partition-based framework is a trade-off between accuracy and speed. The images are partitioned into 5 equal-sized parts, with 4 parts on the corner and 1 part at the center. We had the assumption that the object should either fill up the whole image or should lie in either one of the 5 partitions. The similarity measurement of two images will be represented as follows:

$$Similarity_{color}(X,Y) = \max_{x_i \in X, y_j \in Y}(-||x_i - y_j||^2)$$

where *i, j* are from the partition set of *X, Y*, which is composed by 5 regional partitions and one entire image. By calculating the similarity score for each of the parti-

tion pairs, the maximum score is taken as the similarity score between the two images.

A color histogram was used as the global feature in this model and performed on the partition-based framework. We performed on the HSV color space to extract the histogram vectors. HSV color space outperforms the RGB color space by its invariance to the change of illumination. We conducted the histogram extraction on the Hue component and formed a bin of 10 dimensions.

The data set is then clustered with Affinity Propagation algorithm into several clusters. Affinity Propagation has the advantages that it can automatically determine the number of clusters; treats each image node equally and has a relatively fast merging speed. For the next step, a more accurate pair-wise comparison of near-duplicates will be conducted within each of the clusters with local features.

2.2. Local Approach

We will use the Bag-of-Word model to represent the local visual features. The images are fine partitioned into blocks or represented by interest points; then CPAM descriptor and SIFT descriptor are applied respectively to represent each block or the neighborhood of an interest point.

There is evidence to prove the existence of different visual pathways to process color and pattern in the human visual system. Qiu et. al. [11] proposed the CPAM feature to represent color and pattern visual representations on YC_bC_r color space which gives state-of-the-art performance on content-based image retrieval applications. CPAM feature captures statistically representative chromatic and achromatic spatial image patterns and use the distributions of these patterns within an image to characterize the image's visual content. We build a 256-dimensional codebook for each of the two channels, P channel for pattern representation and C channel for Color representation, from large collection of training data. Each image is encoded with vector quantization technique into a 256-dimensional feature vector. The two types of codebook is concatenated into one 512-dimensional codebook, so the corresponding combined feature vector is a 512-dimension vector, with 256-dimension for pattern channel (P) and 256-dimension for color channel (C).

We also consider another well know local feature descriptor, the Bag-of-Words model with SIFT descriptor, to represent the local visual patterns of the image. For each image, the interest points are extracted with Differe-nce of Gaussian and represented with a 128-dimensional SIFT descriptor. Specifically, the descriptor is formed from a vector containing the values of all the orientation histogram entries. We have followed David Lowes's implementation [12] with a 4 by 4 array of histogram with 8 orientations in each bin. Therefore, the dimension of the feature vector for each interest point is 128. A universal code book is then constructed by collecting all the available interest points from the data set. One critical issue for code book construction is to determine the size of the code book. A size too large or too small may both defect the content representation ability for the images. We have used the grid search method to browse through different size of code book and choose the best code book size in terms of the near duplicate detection performance. In our experiment, we choose a codebook size of 3000 and use the vector quantization technique to encode the images into a 3000-dimensional feature vector.

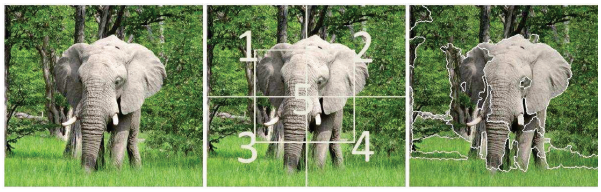

Figure 3. Global feature extraction framework: left to right: image-based, partition-based, segment-based.

2.3. Multi-modal Detection Integration

The CPAM feature and SIFT feature map the input images into different high-dimensional feature spaces respectively. The similarity for the feature points in these two different feature spaces are characterized by different visual aspects of the images, such as color visual pattern and texture visual pattern. As a result, each feature could be used as a strong supplement to the other in nearest neighbor search step, or the near duplicate detection task. Before we could fuse the detection results from two different perspectives, we need to firstly separate the correct matches from the entire return ranking list.

For CPAM feature, given the existence of a large number of negative matches, the similarity between the query image and the negative matches are distributed uniformly in a certain range. If there are positive matches, the similarity between query image and the matched images should be out of the bound of the above range. For a true positive near-duplicate match, the query image and the matched image should be very close to each other on the feature space, while all the negative images are significantly far away from the matched pairs and uniformly distributed in the feature space. If we draw the distance diagram with respect to the returned ranking list, then the negative distances will form a linear function in the larger range, and the top few true matches, if exist, are out- liers of this linear function. This assumption ignites us to use linear regression to reconstruct a linear function to reproduce the distribution of the distances for all the images to the query image, and the top

few outliers, if exist, of this reconstructed linear function should be the true

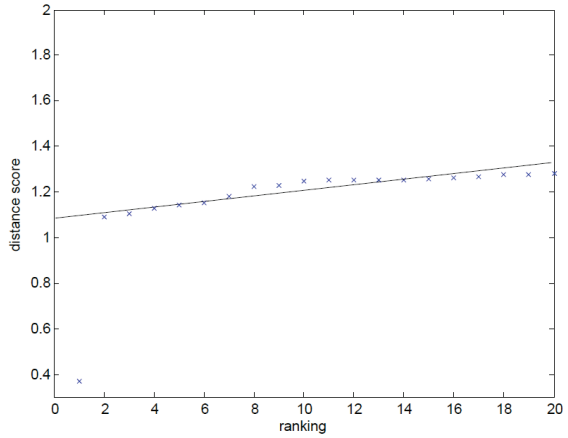

Figure 4. True positive match determination by linear regression on the CPAM feature; x-axis represents the returned ranking index; y-axis represents the corresponding distance values to the query image.

positive matches. As shown in Figure 4, there is one true positive match in the database for the given query image (the first indexed point). The majority of the non-matched distance score can be perfectly modeled by a linear regression function, and the first indexed distance score is left as an outlier to this regression function, which results in a true positive detection of near-duplicate. In actual implementation, we will randomly sample the retrieval results from large range for linear regression and repeat the linear regression operation for several times, to make sure the correct distribution of false matches is discovered and the true matches are left as outliers.

The above linear regression framework models the distribution of the distances to the query image and detect the outliers with respect to the generated linear function with a predefined threshold. The duplicates are extracted as the outliers discovered with the above regression model.

For SIFT BoW model, even true matches does not distinguish itself from all the others by having a significant larger normalized number of matched interest points. The reason for this is due to the limited representation ability of the Bag-of-Word model: non-matched points may also fall into the same bin of the codebook. In order to strictly detect true duplicate pairs, we have calculated the similarity value of the query image with itself, and compared this value with the returned values from the ranking list. We set a threshold heuristically to strictly enforce that only the true duplicates are detected. Specifically, only the similarity values that are close enough to the similarity value of the query image to itself will be accepted as duplication, which can be defined below.

$$ratio = \frac{Similarity_{sift}(i,j)}{N_j \times Similarity_{sift}(i,i)} > threshold, i \neq j$$

where *i* is the query image, *j* is the database image.

Finally, the duplicate extraction results with CPAM BoW model is merged with the SIFT BoW model to form the final result, which realizes the multimodal integration of the two different features. Specifically, for each query, we will retrieve with both CPAM and SIFT BoW models; afterwards, we extract the duplicates from the CPAM BoW model results with linear regression model, and from SIFT BoW model results with self comparison, and then combine the two extraction results to get the final duplications.

3. Evaluation and Discussion

We have evaluated near duplicate detection performance with two similar evaluation metrics, which are *precision/recall* and *average precision*. For *precision/recall* evaluation, we only investigate the first return image from the ranking list, which is the top one detection result. If it is a true positive match, then we say the match for the query image is successfully found; if not, then the match is not detected for this query image. For a ranked list of return results, we also use the *mean average precision* (mean AP) to evaluate the performance, which is more accurate than only evaluating the first returned result. For each query, we consider the average precision up to top 10 return results. Mean AP equals to the mean of the average precision values from all the queries.

For the near duplicate detection, we evaluate our proposed framework with other two baseline algorithms by comparing their performance on both detection accuracy (precision/recall) and computation cost. The first baseline algorithm, Hash-based algorithm, proposed in [13], partitioned the image into blocks, applied a hash function, took the binary string as the feature vector and then grouped the matches based on their Hamming distance. The second baseline algorithm, pair wise-based algorithm, used the CPAM and SIFTS BoW model matching algorithms directly without applying the clustering step. We manually labeled 20 clusters from 20 different categories for duplicate pair detection. We have the following observations: a) the three models have comparable detection precision. The cluster-based model and hash-based model perform similarly and they both outperform the pair wise-based model. b) The hash-based model has a low recall score compared to the other two which means a large false positive rate. The reason is that hash-based method can successfully detect all the duplicate image pairs but miss most of the near-duplicate pairs which varies slightly on the background. On the other hand, the cluster-based model successfully detects most of the near-duplicate pairs. The average performance for the 20 categories can be found in Table 1 and we have

the conclusion that cluster-based model achieved the best average performance among the three models.

In order to evaluate the computation efficiency of the proposed framework, we counted the number of comparisons and recorded the actual runtime for each of the models on "outdoor commercial" set as appeared in Table 2. Experiment ran on a Intel Duo3.0G PC. We observed that, without considering the detection power, hash-based algorithm ran much faster than the algorithms based on local features. If the detection of near-duplicate was a must, the cluster-based model outperformed the pair wise-based model dramatically by saving more than 2/3 of the computation cost. The computation burden for global feature clustering was insignificant, which ran at almost real time (2 sec) compared to local feature step. Furthermore, the cost saving was even remarkable as the scale of the data set increased.

We will evaluate the performance of CPAM and SIFT BoW model on near duplicate detection task; our design of using linear regression to extract true positive near duplicates; and whether or not the multimodal integration structure will improve the performance when compared with using single feature model only. The near duplicate detection techniques are designed for general image collections. In order to make more clear comparison, we will evaluate the proposed techniques and frameworks on a challenging data set, which is composed by 15000 images with both scene and object images. We manually extracted 190 query images. For each image, there is at least one near duplicate can be found in the data set.

Table 1. Duplicate detection performance (precision/recall)

	Cluster-based	Hash-based	Pair-wised
Average	0.72/0.71	0.68/0.50	0.64/0.71

Table 2. Computation efficiency for category "out-comm"

	# of comparisons	Runtime (sec)
Cluster-based	22136	22
Hash-based	400	1
Pair-wised	79600	69

We have compared the effectiveness of our proposed true positive detection extraction framework. The evaluation results are reported in Table 3: The false removal (FR) rate equals to 0.1537, which means the percentage of true positive detections that are falsely removed by the proposed extraction framework. We can observe that a very small percentage of true positives are removed by the proposed framework. Moreover, the removed true positive detection will be recovered by the SIFT BoW model, which is major benefit of our multimodal integration framework. The Recall value equals to 0.9007, which means a very little percentage of false positive detections will be retained in the final detection result. The mean AP and Precision measurement are defined similarly as before. From the recall value and false removal rate, we have the conclusion that the proposed near duplicate detection framework is effective in terms of maintaining the true positive detections, as well as eliminating false positive detections.

Table 3. Performance evaluation of True Positive (TP) extraction with CPAM BoW model.

	Mean AP	Precision	Recall	FR rate
TP extraction	0.7695	0.7751	0.9007	0.1537

For accurate detection with local features, we have evaluated the performance of the CPAM and SIFT BoW model in comparison with the "simply-designed" feature, such as global color histogram. The detection result can be found in Table 4. We can see from the result that the proposed "CPAM + SIFT" model performs the best, especially when compared with using single model of CPAM or SIFT. If using only single feature model, CPAM model performs better than SIFT model on average, however, we have observed both cases that, some queries work better with CPAM models, while some others work better with SIFT model, such as the examples shown in **Figure 5**. The top image pair in Figure 5 can be successfully detected with CPAM model while not be able to detected by SIFT model; the bottom image pair in Figure 5, on the other hand, works with SIFT model rather than CPAM model. As a result, successful combination of the detection ability of both models will inevitably increase the detection performance. Some more detection result with the proposed "CPAM + SIFT" design can be found in **Figure 6**. The top 3 rows in **Figure 6(a)** show successful detection, with the near-duplicate images bounded by a red box; the 4th row in **Figure 6 (b)** shows a negative detection result, where the near-duplicate pair is not detected. The evaluation result of multimodal integration framework compared with single feature model and global feature model can be found in Table 4. From this table, we can observe that local feature models perform significantly better than global features on near duplicate detection task, by at least 50% of performance improvement in terms of mean AP. The proposed the CPAM and SIFT integration model performs the best, followed by using CPAM model alone.

4. Conclusion

We have designed a coarse-to-fine near duplicate detection structure which speeds up the detection process

Table 4. Detection model evaluation: global feature, single local feature and integrated local feature model.

	RGB Color	SIFT	CPAM	CPAM + SIFT
Mean AP	0.3540	0.5650	0.8424	0.8836

Figure 5. Comparison between CPAM and SIFT model: (a) CPAM model works while SIFT model do not work; (b) SIFT model works while CPAM model do not work.

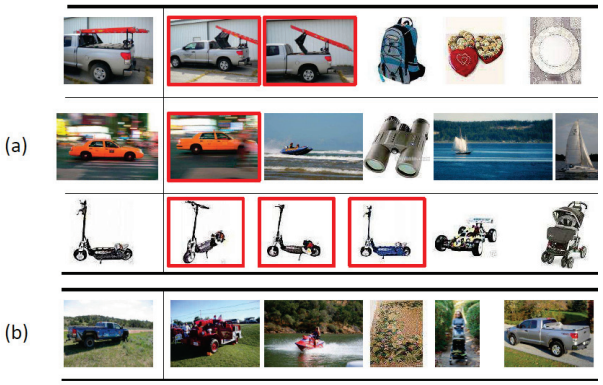

Figure 6. Example results with "CPAM+SIFT" design; query image on left-most column; red bounding box indicates a positive detection.

dramatically compared to traditional pair-wise detection. We further tested the multi-modal feature combination and achieves impressive near duplicate detection results compared to using single feature.

REFERENCES

[1] Sebe, N., Lew, M., and Huijsmans, D. "Multi-scale sub-image search," *Proceedings of the seventh ACM international conference on Multimedia (Part 2)* (1999), *ACM*, pp. 79–82.

[2] Wang, B., Li, Z., Li, M., and Ma, W. "Large-scale duplicate detection for web image search," *In Multimedia and Expo*, 2006, pp. 353–356.

[3] Thomee, B., Huiskes, M., Bakker, E., and Lew, M. "Large scale image copy detection evaluation," *In Proceedings of the 1st ACM international conference on Multimedia information retrieval* (2008), ACM, pp. 59–66.

[4] Tang, F., and Gao, Y. "Fast near duplicate detection for personal image collections," *In Proceedings of the 17th ACM international conference on Multimedia* (2009), ACM, pp. 701–704.

[5] Wu, X., Ngo, C., Hauptmann, A., and Tan, H. "Real-time near-duplicate elimination for web video search with content and context," *IEEE Transactions on Multimedia* 11, 2 (2009), 196–207.

[6] Jaimes, A., Chang, S., and Loui, A. "Detection of non-identical duplicate consumer photographs," *In Information, Communications and Signal Processing*, 2003, vol. 1, IEEE, pp. 16–20.

[7] Zhang, D., and Chang, S. "Detecting image near-duplicate by stochastic attributed relational graph matching with learning," *In Proceedings of the 12th annual ACM international conference on Multimedia* (2004), ACM, pp. 877–884.

[8] Tang, F., and Gao, Y. "Fast near duplicate detection for personal image collections," *In Proceedings of the 17th ACM international conference on Multimedia* (2009), ACM, pp. 701–704.

[9] Jing, Y., Baluja, S., and Rowley, H. "Canonical image selection from the web," *In Proceedings of the 6th ACM international Conference on Image and Video Retrieval* (2007), ACM, pp. 280–287.

[10] Ke, Y., Sukthankar, R., and Huston, L. "Efficient near-duplicate detection and sub-image retrieval," *In ACM Multimedia* (2004), vol. 4, p. 5.

[11] Qiu, G. "Image coding using a coloured pattern appearance model," *In Visual Communication and Image Processing* (2001).

[12] Lowe, D. "Distinctive image features from scale-invariant key points," *International journal of computer vision* 60, 2 (2004), 91–110.

[13] Wang, B., Li, Z., Li, M., and Ma, W. "Large-scale duplicate detection for web image search," In Multimedia and Expo, 2006 IEEE International Conference on (2006), pp. 353–356.

