Self similarity wide-joins for near-duplicate image detection

IEEE International Symposium on Multimedia, 2015, Miami.
http://www.producao.usp.br/handle/BDPI/49670

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
Self Similarity Wide-Joins for Near-Duplicate Image Detection

Luiz Olmes Carvalho, Lúcio F. D. Santos, Willian D. Oliveira, Agma J. M. Traina, Caetano Traina Jr.
Institute of Mathematics and Computer Sciences – University of São Paulo, Brazil

Abstract—Near-duplicate image detection plays an important role in several real applications. Such task is usually achieved by applying a clustering algorithm followed by refinement steps, which is a computationally expensive process. In this paper, we introduce a framework based on a novel similarity join operator, which is able to both replace and speed up the clustering step, whereas also releasing the need for further refinement processes. It is based on absolute and relative similarity ratios, ensuring that top ranked image pairs are in the final result. Experiments performed on real datasets show that our proposal is up to three orders of magnitude faster than the best techniques in the literature, always returning a high-quality result set.

Keywords—near-duplicate; similarity join; wide-join.

I. INTRODUCTION

Mobile devices and online applications such as social networks are increasingly gathering and storing huge volumes of multimedia data, mainly images. The resulting multimedia repositories often store many images very similar.

For example, let us consider famous tourist places such as the Statue of Liberty or the Eiffel Tower. They are repeatedly photographed so many times that many pictures are almost copies from each other. In image retrieval contexts those images are called near-duplicates: instances quite similar among themselves with only minor variations derived from capturing devices or conditions, such as rotation, scaling, cropping, resolution, illumination, framing or other transformations that do not affect the overall content [1].

Near-duplicate image detection is a research field recently gaining attention in the multimedia and information retrieval communities [1][2][3]. For instance, consider the following practical scenarios with distinct follow-ups regarding near-duplicate detection:

- **Social networks:** this scenario present high potential to generate near duplicates as users share, copy, edit and re-post pictures, and it is interesting to remove near-duplicates, fostering more diversified results. Querying the Statue of Liberty should return distinct details or angles of the statue, but not almost-copies of the same image.

- **Computer-aided medical systems:** clinical archive environments such as Picture Archiving and Communication Systems (PACS) store hundreds of thousands of exam images from patients. Aiming at decision-making support, it is desirable that medical systems retrieve the images most similar to those in the exam of the current patient (near-duplicates) once, based on their best previous outcomes, they may help physicians to choose a similar treatment or intervention.

Several techniques had been proposed to accomplish the near-duplicate task [1][2][3][4], but none of them were consolidated in terms of efficiency and efficacy. Although each approach has its own intricacies, in general, they detect near-duplicate images executing a two-phase processing:

1) **Construction:** the first step looks for the potential near-duplicate elements. It can be achieved by clustering techniques [3] or by similarity queries [2], which uses each individual image as a query and retrieves the images most similar to each one.

2) **Refinement:** the second step intends to process the result of the first one, seeking for false positives. Most methods sacrifice computational efficiency in this phase, in favor of improved result efficacy.

Operators widely employed in information retrieval area, namely *similarity joins* and *wide-joins*, can also be employed to detect near-duplicates [2][5]. Similarity joins obtain element pairs that are similar up to a maximum threshold, assuming that each pair corresponds to the basic near-duplicate candidates, which are thereafter submitted to the refinement stage. The Wide-joins [5] were designed to retrieve the overall most similar pairs from two sets, leading to a refinement step embedded naturally in their processing.

Up to now, similarity joins has been applied only to near-duplicate detection in string data [2], so they were not explored in image domains, because the refinement stage tends to be costly and achieves low efficacy. Moreover, wide-joins were designed to process two distinct sets, but the near-duplicate task usually combines a set with itself.

This paper introduces a framework model for near-duplicate image detection employing the wide-join as its core algorithm. We extended the wide-join definition in order to enable it to process a single set, defining a *self wide-join*. It ranks the pairs of near-duplicate images so that the top-ranked ones bubbles up to the duplicate-search result, thus no further refinement step is required.

The experiments performed with two real datasets show that our proposal improves the best approaches from the literature in at least 2 orders of magnitude regarding the execution time whereas returning a better-quality result set.
This paper is organized as follows. Section II describes the main concepts and related works. Section III introduces our framework to detect near-duplicate images and the definition of self wide-joins. Section IV presents the experimental evaluation and discusses the main results. Section V summarizes the main concepts and outlines future improvements.

II. BACKGROUND

Near-Duplicate Detection. Commonly-used approaches for near-duplicate image detection are the Bag-of-Visual-Words and the Bag-of-Phrases models [6]. They encode the features at each local image region as a visual word and represent images as a histogram of the resulting words [3]. Such models are less reliable, once they consider only local descriptors and do not retain the spatial relationship that may hold among the features. Moreover, they also rely on post-processing, which increases the computational cost.

Another approach presented in the literature aims at discovering near-duplicate image groups. The study introduced in [3] performs such computation by constructing clusters using the \( k \)-means algorithm. Thereafter, the coherency of each cluster is evaluated to determine the need of further processing. In the subsequent processing, clusters are subdivided again until their coherency value is less than the one of the original partition. However, the requirement for post-processing turns that technique computationally expensive. Moreover, once \( k \)-means is sensitive to outliers and the computed centroids are not real images, the resulting clusters may not achieve a good quality.

Similarity Join. A similarity join is a binary operation that receives two sets of elements and makes pairs such that the distance between them does not exceed a maximum threshold. Formally, let \( D \) be a data domain, \( d \) be a metric such that \( d : D \times D \mapsto \mathbb{R}^+ \), \( R \) and \( S \) be sets sampled from \( D \) and \( \xi \) be a similarity limiar. A similarity join \( R \bowtie \langle d(r,s) \leq \xi \rangle S \) retrieves the set \( \{ (r,s) \in R \times S \mid d(r,s) \leq \xi \} \) [5].

With respect to near-duplicates detection, similarity joins have been considered only regarding data represented as tokens [2], using metrics such as Edit and Hamming distances, while our proposal considers more general metrics, including the Euclidean distance. Our proposal departs from the wide-join operator [5], enabling it to combine a set of metrics with itself and retrieving only the pairs most similar in general, which corresponds to the near-duplicates.

III. SELF SIMILARITY WIDE-JOINS

Intuitively, an image retrieval system should retrieve the images closer to the query and that are the most relevant. Detection of near-duplicate images in multimedia repositories is motivated by distinct interests, to remove, preserve or just keeping track of them. In such context, detecting near-duplicates in query answers plays an important role to obtain a more useful result and to reduce the user’s efforts to navigate and analyze the answer. In Section III-A we present our proposed framework to detect near-duplicates and in Section III-B we define the similarity wide-join extension to accomplish such processing.

A. The Architecture of the Framework

Two modules compose the proposed framework for near-duplicate detection, as shown in Figure 1. The first one – Feature Extractor – receives an image repository as input. It extracts the visual features of each image depending on the kind of visual aspect considered, i.e., color, shape, texture, etc. It processes images in a way similar to a CBIR system, representing each image as a \( n \)-dimensional feature vector.

The second module – Near Duplicate Detection – is the framework’s core module. It receives the set of feature vectors as input and executes the specialized similarity join operator (described in Section III-B). The feature vectors are compared to each other according to a distance function. Our approach employs two user-defined parameters to tune the algorithm to follow the user’s perception of when image pairs can be considered near-duplicates.

B. The Self Wide-Join Operator for Near-Duplicates

The plain similarity join operator is troublesome when employed to detect near-duplicates, as its result set has unpredictable, often too high cardinality. In fact, the result tends to contain more pairs of elements than it is really needed or expected, including pairs quite similar to each other and pairs whose similarity are doubtful. To avoid such shortcoming, the plain similarity join answer requires a refinement step.

For example, a photo from the Statue of Liberty front is more similar to a photo taken one step to the right than a third photo taken twenty steps to the right, but they are
all near-duplicates. Furthermore, those three photos are less similar to another one taken from behind the statue, although those may yet be considered duplicates if they are yet too similar to each other. The intuition here is that there are two “similarity degrees” to be considered in near-duplicate image detection:

- **Absolute**: related to the construction phase. Two images \( a \) and \( b \) are considered near-duplicates iff the dissimilarity between them is at most a user-defined value \( \xi \), that is: \( d(a, b) \leq \xi \).
- **Relative**: establishes a similarity degree of each image pair considering a group of near duplicate images.

Our proposal assumes that regarding a certain tradeoff, two images may be considered absolute near-duplicates, whereas diverse enough for the relative criterion when they analyzed within its own similarity group. The relative degree can be obtained ranking the dissimilarities among pairs such that a given amount of top-ranked pairs in fact compose the set of near-duplicates, disregarding the similarity value.

The traditional similarity join operator intrinsically provides the absolute measure of similarity, but it requires a post-processing or additional semantic information in order to perform the relative evaluation. However, both steps can be atomically performed by the self wide-join operator, defined as follows.

**Definition 1** (Self Similarity Wide-Join). Let \( \mathbb{D} \) be a data domain subjected to a distance function \( d : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+ \). \( S \) be a set of elements sampled in \( \mathbb{D} \), \( \kappa \) be an upper bound parameter and \( \xi \) be a maximum similarity threshold. Then, a self similarity wide-join \( \sigma_{d(t,s)\leq\xi,\kappa} S \) is a unary operator that performs an inner similarity join, sort the intermediate result by the dissimilarity between each pair and returns the \( \kappa \) pairs \( \{s_i, s_j\} \) most similar. The self similarity wide-join is expressed in Relational Algebra as:

\[
\sigma_{d(t,s)\leq\xi,\kappa} S \\
\sigma_{ord\leq\kappa} \left( \pi_{\{s_i, s_j\}, F(d(s_i, s_j))\rightarrow ord} \left( S \times \sigma_{(d(s_i, s_j)\leq\xi)} S \right) \right)
\]

In (1), we employ \( F \) as an aggregate function that receives the distances between elements \( s_i, s_j \) and returns the ordinal classification of the dissimilarity values. The ordinal values are projected into the extended attribute \( ord \) and employed to filter the \( \kappa \) better ranked pairs (truly near-duplicates).

Self similarity wide-joins can be implemented following the procedure shown in Figure 2. The algorithm takes as inputs the set \( S \) of images to be analyzed and the maximum threshold \( \xi \). Steps 3 and 4 correspond to the inner similarity join, and the absolute similarity check is performed in step 6. The pairs qualifying as near-duplicates are added into a priority queue (step 7), which performs the relative step. The priority in the queue is the similarity distance: a lower distance corresponds to a higher priority for removal. When the procedure finishes, the priority queue \( P \) contains the near-duplicate images, where the top-\( \kappa \) are the most similar.

**IV. EXPERIMENTS**

This section aims at evaluating the proposed framework regarding the computational performance and the answer quality as well.

We describe the results obtained processing two real datasets: USPTex\(^1\) and ShapesCN\(^2\). The USPTex dataset contains 2292 texture images. In this dataset, 191 are completely distinct from each other, and each texture has 11 similar (near-duplicates). We represent this dataset using the Texture Spectrum extractor, which obtains 8 features, and used the \( L_2 \) metric to perform the comparisons. The ShapesCN dataset contains 5500 fish contours, represented by Zernike moments which results in 72 features. There are 1100 distinct fishes contours and each one has 4 near-duplicates. The \( L_2 \) metric was also employed to compare this dataset. Also, we compared our self wide-join (WJ) proposal with the adaptive cluster (AC) [3] technique.

The experiments were performed in a computer with an Intel® Core™ i7-4770 processor running at 3.4 GHz, with 16 GB of RAM under Ubuntu 14.04. All methods were implemented in C++ using the same framework. The performance of both approaches were evaluated measuring the total running time and the answer quality was measured using Precision and Recall (\( P \times R \)) graphs.

Figure 3(a) shows the execution time required for the two techniques to detect the near-duplicate images in USPTex. The self-similarity wide-join was 3 orders of magnitude faster than the adaptive cluster. The WJ gain over AC corresponds to 99.89%.

Figure 3(c) presents the runtime of both methods executed over the ShapesCN dataset. As it can be seen, the wide-join was again the fastest technique, being two orders of magnitude faster than the adaptive cluster, which represents a gain of 98.91%.

---


---
The main reason for those improvements is that AC clusters the search space containing the images (using the k-means) and recursively re-clusters each previous cluster in a hierarchical way until their coherency do not exceed the objective value. Therefore, each further re-clustering step and the corresponding local refinement are much costlier than the WJ algorithm. The presented measurements correspond to the execution of the Near-Duplicate Detection Module only (Section III), as the feature extraction was performed just once to provide the same data for both techniques.

Figure 3(b) presents the P × R curves of the two methods processing USPTex. As it can be seen, the self-wide-join consistently obtained the highest precision for every recall amount. The precision gain of WJ over AC was at least 8.81%, for a recall level of 100%. The maximum improvement occurred at a recall level of 30%, when WJ was 32.46% more precise than AC. In average, WJ was 21.86% more precise.

Figure 3(d) shows the P × R curves obtained to process ShapesCN. Again, the self wide-join obtained the highest precision at all recall levels. Both methods had similar precision for recalls up to 20%, but thereafter WJ improves to obtain a gain peaking at 32.81% for recall levels of 50% and 60%. In average, WJ was 16.90% more precise than AC.

In general, the result of our self wide-join was in average 21.86% more precise than AC to process the USPTex dataset and at least 16.90% more precise than AC regarding to ShapesCN dataset, in average.

V. CONCLUSION

Detecting near-duplicate images in multimedia repositories has practical application in several real scenarios. The methods in the literature aimed at supporting it generally iterate two phases: construction and refinement, where the latter impacts the efficiency of the entire process.

This paper presented a framework model for near-duplicate image detection employing the similarity wide-join operator as its key algorithm. Wide-joins enable retrieving the most similar element pairs and naturally present the results ordered by the similarity among the elements. The order information allows defining a relative similarity degree among groups of pairs considered as near-duplicate images, so that the top-ranked ones compose the most similar in the search result.

We extended the wide-join operator to enable computing the self-similarity of a single set, providing the definition of self wide-joins. We also presented an algorithm to process it. The experiments, performed on two real datasets, showed that our framework improves the performance of the existing techniques in at least 2 orders of magnitude, whereas also providing a significantly better result than other recent alternatives. As a future work, we are exploring combining the images content with externally assigned metadata, aiming at improving even more both the precision and performance of our technique.

ACKNOWLEDGMENT

The authors are grateful to FAPESP, CNPq, CAPES and Rescuer (EU FP7-614154 / CNPq 490084/2013-3) for their financial support.

REFERENCES