Fast exhaustive-search equivalent pattern matching through norm ordering

Mohamed Yousef *, Khaled F. Hussain

Faculty of Computers and Information, Assiut University, Assiut, Egypt

ARTICLE INFO

ABSTRACT

Pattern matching is a fundamental problem in computer vision, and image and video processing. Exhaustive-search equivalent algorithms yield the same results as exhaustively searching all patterns in the image but significantly faster. In this paper, we propose a novel exhaustive-search equivalent algorithm that is combined with a number of state-of-art algorithms to provide a significantly faster alternative in the problem of finding nearest pattern according to a predefined distance measure. Our technique also shows high resilience to both blurring and JPEG compression types of noise. This is demonstrated in the paper with results from over 15 million runs for *each* compared algorithm.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Template matching or pattern matching is a fundamental problem in computer graphics, computer vision, signal processing, image and video processing, with a wide variety of applications such as video block motion estimation [1], image based rendering [2], image inpainting [3], object detection [4], super resolution [5], texture synthesis [6], texture edge extraction [7], image filtering [8], image summarizing [9], and even image compression [10].

The template matching problem can be defined as follows: given an $N = n \times n$ pixels image sub-window q (called Template or Patch) find either the most similar patch to it in the image, or all patches p where the distance between p and q is below a certain threshold T according to a predefined dissimilarity measure. The basic algorithm for solving this problem is full search or exhaustive-search, in which the distance between q and all template-sized sub-windows in the image (overlapped) is computed and we return either the patch with the smallest distance or all patches with distance below T, according to the instance of the problem we are considering.

In this work we are interested in the first instance where only the closest patch is required and many number of queries is done on the same image, as this is the interest of most of the applications that require template matching (however, all algorithms presented in the paper can be used for the two instances of the pattern matching problem). This has important implication on our evaluation method of algorithms, as the most important property we require is exact fast evaluation of many number of queries on the same image, so a relatively slow initialization of an algorithm is acceptable as long as the total time (initialization + query processing) is fast compared to the respective times of other algorithms.

In this paper we propose a novel exhaustive-search equivalent pattern matching algorithm that shows significant speed-ups over current state of art algorithms. This is done through combining a novel pattern rejection and pattern evaluation order scheme with rejection ideas from other current state of art algorithms. Our algorithm shows very high resilience to blurring noise and medium resilience to JPEG compression noise, the results are demonstrated through experiments that included more than 15 million runs for each measured algorithm, using the dataset introduced in [11]. It should be noted that the results introduced here cannot be directly compared in numbers with those on [11], as the application is different, also in that paper they used a hand-computed input threshold, where here there is no input from user and the process is completely autonomous.

2. Related work

There is a huge history of work in accelerating pattern matching process, methods can be classified into two broad categories: Exact and Approximate pattern matching. A complete review is beyond the scope of this article, the reader might refer to [12] for extensive reviews. Here we are only interested in exact methods and will give a brief overview on them.

---

* Corresponding author.
E-mail addresses: mohamed.mahdi@compit.au.edu.eg (M. Yousef), khussain-n@aun.edu.eg (K.F. Hussain).

One of the simplest and earliest trails to accelerate Exact pattern matching is Partial Distortion Elimination (PDE) [13] which simply terminates the evaluation of the current candidate sub-window as soon as the current value of norm is guaranteed to exceed the current minimum distance.

Fast-Fourier transform-based approaches [14] have also been traditionally used for accelerating pattern matching in the L_2 norm, especially for large pattern sizes. The idea is based on observing that the L_2 norm between two M pixels-sized sub-windows X, Y can be written as

$$\|X - Y\|_2^2 = \|X\|_2^2 + \|Y\|_2^2 - 2 \cdot \sum_{i=1}^{M} x_i \cdot y_i \qquad (1)$$

FFT allows fast computation of the third term $\sum_{i=1}^{M} x_i \cdot y_i$ in the frequency domain using the correlation theorem. $\|X\|_2^2$ is calculated for the all sub-windows in the image using fast incremental techniques like Summed Area Tables [15] or Box-Filtering [16], $\|Y\|_2^2$ is computed once where Y is the input query.

Projection Kernels (PK) based algorithms [17, 18, 19] are based on the idea that we can get approximate value of the distance $d(X - Y)$ (where d is a norm) by projecting both X and Y into a number of basis vectors $U = \{u_1, \ldots, u_k\}$, generally if U are mutually orthonormal it can be shown that:

$$d(X - Y) \geqslant \sum_{n=1}^{k} \frac{1}{d(u_n)} d(X^T u_n - Y^T u_n) \qquad (2)$$

this provides a lower bound on the actual distance value between X, Y. The algorithm works by providing very fast method of projecting all image sub-windows to U, then when examining candidates, it keeps summing the values of $d(X^T u_n - Y^T u_n)$ (assuming U are unit vectors) and rejects the candidate once the sum exceeds the current minimum or the input threshold value. In all these methods U are basis vectors of the Walsh–Hadamard transform.

Low Resolution Pruning (LRP) [20] is based on the idea of transforming sub-windows in the image into a lower resolution variants, by summing the pixel values of partitions of the sub-windows of size M, this is repeated for T levels of resolution, after that an upper bound on the value of $d(X - Y)$ (where d is an L_p norm) is established for every resolution level t, assuming min is current minimum distance or input threshold value, as follows

$$d(X - Y) \leqslant M^{\frac{t(p-1)}{2p}} \cdot min \qquad (3)$$

As shown in [11] LRP has the same underlying structure as PK but with different U and lower bounding function.

Incremental Dissimilarity Approximations Algorithm (IDA) [21] is based on the idea of partitioning each candidate sub-window into a number r of disjoint sets by defining a partition P of set $S = \{S_1, \ldots, S_M\}$, using this partitioning it is able to establish a lower bound on the value $d(X - Y)$ (where d is an L_p norm), at level k we have

$$d(X - Y) \geqslant \sum_{t=1}^{k-1} \left(\sum_{i \in S_t} |x_i - y_i|^p \right) + \sum_{l=k}^{M} \left| \|X\|_{p,S_l} - \|Y\|_{p,S_l} \right|^p \qquad (4)$$

The right side of the right hand side of the inequality represents the source of the speed-up as it is an estimation of the rest of the distance that is not actually calculated (the calculated part is the left side). So a candidate is rejected at the point when right hand side of the inequality exceeds current minimum distance or input threshold value min.

Tree-based Hierarchical methods have long been used for exact pattern matching problems, as it resembles a special case of the famous Nearest Neighbour (NN) problem which has been extensively studied in computer science (see [12] for a recent example), a recent survey on exact hierarchical tree structures

for accelerating pattern matching is [22], the problem with all these methods is that their performance deteriorates significantly as the dimensionality (here, the size of query template) increases. This is illustrated in [22].

Another approach [23] was making use of the sequential overlap of image sub-windows to decrease the computational complexity of the process through eliminating redundant similarity computations of sequentially overlapped regions in both the query template and image sub-windows. The main drawback of this method is that it assumes sequential overlap in the query template, however in many cases and for many applications this is not the case and query templates may be totally different or from different parts from the image.

3. Norm ordered matching

3.1. Derivation

We represent each $N = n \times n$ pixels image sub-window as an N dimensional vector $X = [x_1, \ldots, x_N]^T$, which is represented as a point in an N-Dimensional space $\Re^N$ as shown in Fig. 1.

Let $d(p)$ be a norm defined over the vector space $\Re^N$, so $d(p - q)$ represents the dissimilarity measure between the two N-Dimensional points p, q derived from the norm d.

Since d is a norm, it satisfies the triangle inequality [24], so we have:

$$|d(p) - d(q)| \leqslant d(p - q) \qquad (5)$$

This splits into two inequalities

$$d(p) - d(q) \leqslant d(p - q) \qquad (6)$$
$$-(d(p) - d(q)) \leqslant d(p - q) \qquad (7)$$

Eq. 6 can be written as:

$$d(p) \leqslant d(q) + d(p - q). \qquad (8)$$

If m is the current minimum distance to input query q as in Fig. 1, and for a candidate sub-window p we found that (as the case for the c_x points in the figure):

$$d(p) > d(q) + m. \qquad (9)$$

Comparing Eqs. 6 and 9, we can simply find that $m < d(p - q)$ so we can directly reject p without any further calculations, as it cannot improve the current minimum m.
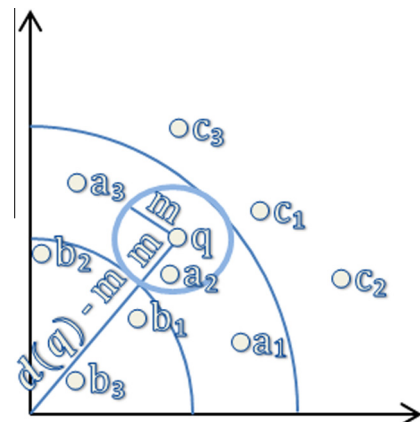
In the same manner, Eq. 7 can be written as:

Fig. 1. Each m-Dimensional vector is represented by a point. a_x, b_x, c_x represent image sub-windows (candidate matches), q is the query template. m is the current minimal distance.