Pattern Recognition Letters 32 (2011) 2119–2127

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Adaptive Low Resolution Pruning for fast Full Search-equivalent pattern matching

Federico Tombari [a,*], Wanli Ouyang [b], Luigi Di Stefano [a], Wai-Kuen Cham [b]

[a] DEIS/ARCES, University of Bologna, Bologna, Italy
[b] Dept. of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 18 October 2010
Available online 24 August 2011
Communicated by S. Todorovic

Keywords:
Pattern matching
Template matching
Low Resolution Pruning
Full Search-equivalent

ABSTRACT

Several recent proposals have shown the feasibility of significantly speeding-up pattern matching by means of Full Search-equivalent techniques, i.e. without approximating the outcome of the search with respect to a brute force investigation. These techniques are generally heavily based on efficient incremental calculation schemes aimed at avoiding unnecessary computations. In a very recent and extensive experimental evaluation, Low Resolution Pruning turned out to be in most cases the best performing approach. In this paper we propose a computational analysis of several incremental techniques specifically designed to enhance the efficiency of LRP. In addition, we propose a novel LRP algorithm aimed at minimizing the theoretical number of operations by adaptively exploiting different incremental approaches. We demonstrate the effectiveness of our proposal by means of experimental evaluation on a large dataset.

1. Introduction

Pattern matching is a computationally expensive technique aimed at locating the instances of a pre-defined *pattern*, or *template*, within an image. Pattern matching is widely adopted for tasks such as industrial inspection (quality control, defect detection) and fiducial-based pick-and-place, though it has also been used for image compression, face detection, action recognition. The standard technique for pattern matching, here recalled as Full-Search (FS), is based on the *sliding window* approach: i.e. the pattern is compared with all the equally-sized subsets contained in the image (also referred to as *image candidates*) by means of a similarity or dissimilarity score. Typical dissimilarity measures employed for this task are those derived from the *p-norm*, such as the *Sum of Absolute Differences* (SAD) and the *Sum of Squared Differences* (SSD). With these measures, when dissimilarity turns out to be below a pre-defined matching threshold, an instance of the pattern is located. The use of the sliding window approach guarantees that all below-threshold instances are located. Nevertheless, this leads to a computationally intensive process, so that many proposals in literature attempt to decrease its computational burden. In such a framework, several techniques have been proposed that try to speed-up the FS algorithm by means of an approximated search (Rosenfeld and Vanderburg, 1977, Vanderburg and Rosenfeld, 1977; Barnea and Silverman, 1972).

More interestingly, other proposals aim at speeding-up pattern matching without deteriorating the outcome of the search, that is, while achieving exactly the same results as the FS approach: we will recall these approaches as *exhaustive* or *FS-equivalent*. Recently this research topic has been particularly active, with many novel proposals attaining dramatic speed-ups with respect to the FS. The state of the art in the field is currently represented by the Low Resolution Pruning (LRP) algorithm (Gharavi-Alkhansari, 2001), the Incremental Dissimilarity Approximations (IDA) algorithm (Tombari et al., 2009), and by approaches based on the Walsh–Hadamard Transform such as Projection Kernels (PK) (Hel-Or and Hel-Or, 2005) and Grey-Code Kernels (Ben-Artz et al., 2007; Ouyang and Cham, 2009). Even more recently, a novel approach based on the Haar Transform has been proposed (Ouyang et al., 2010). Such wealth of new proposals has inspired a comprehensive comparative analysis based on a theoretical investigation of their commonalities and differences as well as on an experimental evaluation carried out on a large image dataset characterized by different nuisance factors and with different hardware platforms (Ouyang et al., in press). The results reported in (Ouyang et al., in press) show clearly, that, overall, LRP turns out to be the best performing algorithm. More precisely, LRP and GCK are the most efficient methods with the SSD measure under different nuisances (Gaussian noise, blurring, JPEG compression), while LRP and IDA are the most efficient ones under the same nuisances and with the SAD measure.

Given its prominence among *FS-equivalent* fast pattern matching methods, in this paper we investigate on how to further improve the LRP algorithm. In particular, LRP requires the computation of the image candidates at different resolution levels, this step being performed efficiently by means of incremental techniques that exploit recursive schemes. However, we have carried out a computational

* Corresponding author.
E-mail address: federico.tombari@unibo.it (F. Tombari).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.07.030

analysis of LRP showing that, given the high efficiency of the overall algorithm, in most cases the above step accounts for a significant fraction of the total number of operations, the only exception being those applications where a large number of patterns needs to be compared to the same target image. Hence, a specific investigation on the most efficient strategies for optimizing the computation of image candidates at different resolutions holds the potential to further improve the performance of the LRP algorithm.

This paper includes a twofold contribution. Firstly, we propose a computational analysis of LRP which allows us to demonstrate that the hierarchical incremental scheme for attaining the image candidates originally proposed in (Gharavi-Alkhansari, 2001) represents a non-optimal solution and, accordingly, to highlight more efficient methods suited to LRP. Secondly, based on this analysis, we propose a novel LRP algorithm aimed at minimizing the theoretical number of operations by adaptively exploiting different incremental approaches. By means of an extensive quantitative evaluation performed on a vast dataset, we show significant computational benefits in terms of measured execution times brought in by the proposed approach.

2. Derivation of the method

2.1. The LRP algorithm

Let $X = [x_1, x_2, \ldots, x_N]$ be a N-sized vector representing the pattern and $Y_1, \ldots, Y_K$ be the K N-sized vectors representing the image candidate vectors against which X must be matched, each candidate extracted from a different squared *sliding window* on the image. Also, let J be the length of image vector (i.e. the total number of image pixel). Fig. 1 helps to better understand the notation used throughout the paper and how image candidates are defined.

The first step of LRP is a *Rejection step* carried out over $T + 1$ levels of resolution, starting from level T (lowest resolution) up to level 0 (full resolution). As illustrated in Fig. 2, at each level t, the length of each image candidate vector Y_j and of the pattern vector X is reduced by a factor M by computing each new element as the sum of the M neighboring elements at the finer resolution. Then, each candidate undergoes checking the following pruning condition:

$$\delta_p\left(X^t, Y_j^t\right) > D^t \tag{1}$$

with X^t, Y_j^t being, respectively, the pattern and current candidate at resolution t, and δ_p representing the dissimilarity measure induced from the *p-norm* (SSD with $p = 2$, SAD with $p = 1$):

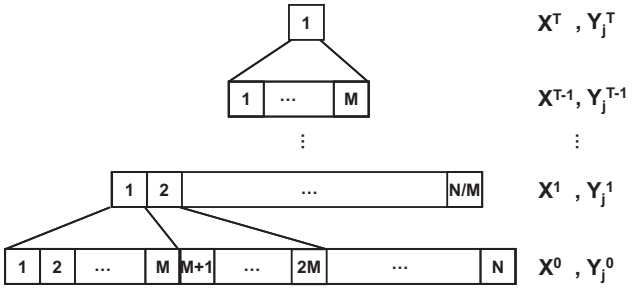

Fig. 2. In LRP, each image candidate Y_j^t and the pattern X^t are transformed to their lower-resolution level by summing up M-sized subsets of neighboring elements. The number reported in each block is the index of the corresponding vector element.

$$\delta_p(X, Y_j) = \|X - Y_j\|_p^p = \sum_{i=1}^{N} |x_i - y_{j,i}|^p. \tag{2}$$

Finally, D^t is obtained from a matching threshold computed, e.g. by selecting a minimum allowed distance δ_p^{min} as:

$$D^t = \|A\|_{p,t}^p \cdot \delta_p^{min} \tag{3}$$

with $\|A\|_{p,t}$ defined as Gharavi-Alkhansari (2001):

$$\|A\|_{p,t} = M^{\frac{t\cdot(p-1)}{2p}}. \tag{4}$$

If Eq. (1) holds, then Y_j is removed from the list of possible pattern instances for the next levels. Once the pruning conditions are tested at all levels, during the second step (*FS step*) of the algorithm a FS process is carried out over the remaining candidates by checking:

$$\delta_p(X, Y_j) < \delta_p^{min}. \tag{5}$$

All candidates for which (5) holds represent a pattern instance in the image.

2.2. Computational analysis

Table 1 reports the computational analysis of the LRP algorithm in terms of different operations (i.e. additions, multiplications (if $p = 2$), absolute values (if $p = 1$), branches) for the two steps of the LRP algorithm. Term Ω represents the cost to compute the image candidates at different resolutions as shown in Fig. 2 (whilst the pattern at different resolutions is computed and stored once

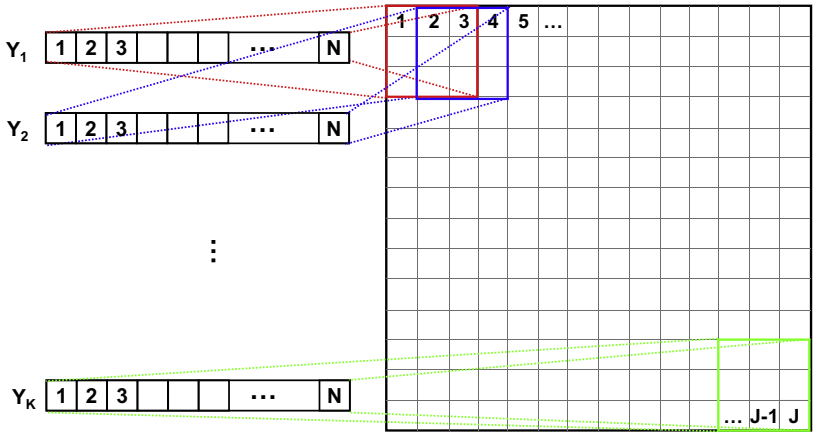

Fig. 1. Notation used throughout the paper. K image candidates, each of length N, are extracted from different squared *sliding window* out of the image, the latter having size J pixels. The number reported in each block is the index of the corresponding vector element.