Optimal Exact and Fast Approximate Two Dimensional Pattern Matching Allowing Rotations

Kimmo Fredriksson[1]*, Gonzalo Navarro[2]**, and Esko Ukkonen[1]***

[1] Department of Computer Science, University of Helsinki
{kfredrik,ukkonen}@cs.helsinki.fi

[2] Department of Computer Science, University of Chile
gnavarro@dcc.uchile.cl

Abstract. We give fast filtering algorithms to search for a 2–dimensional pattern in a 2–dimensional text allowing any rotation of the pattern. We consider the cases of exact and approximate matching under several matching models, improving the previous results. For a text of size $n \times n$ characters and a pattern of size $m \times m$ characters, the exact matching takes average time $O(n^2 \log m/m^2)$, which is optimal. If we allow k mismatches of characters, then our best algorithm achieves $O(n^2 k \log m/m^2)$ average time, for reasonable k values. For large k, we obtain an $O(n^2 k^{3/2} \sqrt{\log m}/m)$ average time algorithm. We generalize the algorithms for the matching model where the sum of absolute differences between characters is at most k. Finally, we show how to make the algorithms optimal in the worst case, achieving the lower bound $\Omega(n^2 m^3)$.

1 Introduction

We consider the problem of finding the exact and approximate occurrences of a two–dimensional *pattern* of size $m \times m$ cells from a two–dimensional *text* of size $n \times n$ cells, when all possible rotations of the pattern are allowed. This problem is often called *rotation invariant template matching* in the signal processing literature. Template matching has numerous important applications in image and volume processing. The traditional approach [6] to the problem is to compute the cross correlation between each text location and each rotation of the pattern template. This can be done reasonably efficiently using the Fast Fourier Transform (FFT), requiring time $O(Kn^2 \log n)$ where K is the number of rotations sampled. Typically K is $O(m)$ in the 2–dimensional (2D) case, and $O(m^3)$ in the 3D case, which makes the FFT approach very slow in practice. However, in many applications, "close enough" matches of the pattern are also accepted. To

* Work supported by ComBi and the Academy of Finland.
** Partially supported by the Millenium Center for Web Research.
*** Work supported by the Academy of Finland.

A. Apostolico and M. Takeda (Eds.): CPM 2002, LNCS 2373, pp. 235–248, 2002.

this end, the user may specify a parameter k, such that matches that have at most k differences with the pattern should be accepted.

Efficient two dimensional combinatorial pattern matching algorithms that do not allow rotations of the pattern can be found, e.g., in [5,2,4,14]. Rotation invariant template matching was first considered from a combinatorial point of view in [10]. In this paper, we follow this combinatorial line of work. If we consider the pattern and text as regular grids, then defining the notion of matching becomes nontrivial when we rotate the pattern: since every pattern cell intersects several text cells and vice versa, it is not clear what should match what. Among the different matching models considered in previous work [10,11,12], we stick to the simplest one in this paper: (1) the geometric center of the pattern has to align with the center of a text cell; (2) the text cells involved in the match are those whose geometric centers are covered by the pattern; (3) each text cell involved in a match should match the value of the pattern cell that covers its center.

Under this *exact matching* model, an online algorithm is presented in [10] to search for a pattern allowing rotations in $O(n^2)$ average time.

The model (a 3D version) was extended in [12] such that there may be a limited number k of mismatches between the pattern and its occurrence. Under this *mismatches* model an $O(k^4 n^3)$ average time algorithm was obtained, as well as an $O(k^2 n^3)$ average time algorithm for computing the lower bound of the distance; here we will develop a 2D version whose running time is $O(k^{3/2} n^2)$. This works for any $0 \leq k < m^2$. For a small k, an $O(k^{1/2} n^2)$ average time algorithm was given in [9].

Finally, a more refined model [13,9,12] suitable for gray level images adds up the absolute values of the differences in the gray levels of the pattern and text cells supposed to match, and puts an upper limit k on this sum. Under this *gray levels* model average time $O((k/\sigma)^{3/2} n^2)$ is achieved, assuming that the cell values are uniformly distributed among σ gray levels. Similar algorithms for indexing are presented in [13].

In this paper we present fast filters for searching allowing rotations under these three models. Table 1 shows our main achievements (all are on the average). The time we obtain for exact searching is average-case optimal. For the k–mismatches model we present two different algorithms, based on searching for pattern pieces, either exactly or allowing less mismatches. For the gray levels model we present a filter based on coarsening the gray levels of the image, which makes the problem independent on the number of gray levels, with a complexity approaching that of the k–mismatches model.

2 Problem Complexity

There exists a general lower bound for d–dimensional exact pattern matching. In [17] Yao showed that the one–dimensional string matching problem requires at least time $\Omega(n \log m/m)$ on average, where n and m are the lengths of the string and the pattern respectively. In [14] this result was generalized for the