Efficient One Dimensional Real Scaled Matching

Amihood Amir\textsuperscript{1,*}, Ayelet Butman\textsuperscript{2}, Moshe Lewenstein\textsuperscript{3},
Ely Porat\textsuperscript{3}, and Dekel Tsur\textsuperscript{4}

\textsuperscript{1} Bar-Ilan University and Georgia Tech.
ami@cs.biu.ac.il
\textsuperscript{2} Holon Academic Institute of Technology
butmoss@zahav.net.il
\textsuperscript{3} Bar-Ilan University
\{moshe,porately\}@cs.biu.ac.il
\textsuperscript{4} University of Haifa
dekelts@cs.haifa.ac.il

Abstract. Real Scaled Matching refers to the problem of finding all locations in the text where the pattern, proportionally enlarged according to an arbitrary real-sized scale, appears. Real scaled matching is an important problem that was originally inspired by Computer Vision. In this paper, we present a new, more precise and realistic, definition for one dimensional real scaled matching, and an efficient algorithm for solving this problem. For a text of length \(n\) and a pattern of length \(m\), the algorithm runs in time \(O(n \log m + \sqrt{nm^{3/2}} \sqrt{\log m})\).

1 Introduction

The original classical string matching problem \([10,13]\) was motivated by text searching. Wide advances in technology, e.g. computer vision, multimedia libraries, and web searches in heterogeneous data, have given rise to much study in the field of pattern matching.

Landau and Vishkin \([15]\) examined issues arising from the digitization process. Once the image is digitized, one wants to search it for various data. A whole body of literature examines the problem of seeking an object in an image.

In reality one seldom expects to find an exact match of the object being sought, henceforth referred to as the \textit{pattern}. Rather, it is interesting to find all text locations that “approximately” match the pattern. The types of differences that make up these “approximations” are:

1. \textit{Local Errors} – introduced by differences in the digitization process, noise, and occlusion (the pattern partly obscured by another object).
2. \textit{Scale} – size difference between the image in the pattern and the text.
3. \textit{Rotation} – angle differences between the pattern and text images.

\textsuperscript{*} Partially supported by ISF grant 282/01. Part of this work was done when the author was at Georgia Tech, College of Computing and supported by NSF grant CCR–01–04494.
Some early attempts to handle local errors were made in [14]. These results were improved in [8]. The algorithms in [8] heavily depend on the fact that the pattern is a rectangle. In reality this is hardly ever the case. In [6], Amir and Farach show how to deal with local errors in non-rectangular patterns.

The rotation problem is to find all rotated occurrences of a pattern in an image. Fredriksson and Ukkonen [12], made the first step by giving a reasonable definition of rotation in discrete images and introduce a filter for seeking a rotated pattern. Amir et al. [2] presented an $O(n^2m^3)$ time algorithm. This was improved to $O(n^2m^2)$ in [7].

For scaling it was shown [9,5] that all occurrences of a given rectangular pattern in a text can be found in all discrete scales in linear time. By discrete scales we mean natural numbers, i.e. the pattern scaled to sizes 1, 2, 3,...

The first result handling real scales was given in [3]. In this paper, a linear time algorithm was given for one-dimensional real scaled matching. In [4], the problem of two-dimensional real scaled matching was defined, and an efficient algorithm was presented for this problem.

The definition of one-dimensional scaling in [3] has the following drawback: For a pattern $P$ of length $m$ and a scale $r$, the pattern $P$ scaled by $r$ can have a length which is far from $mr$. In this paper, we give a more natural definition for scaling, which has the property that the length of $P$ scaled by $r$ is $mr$ rounded to the nearest integer. This definition is derived from the definition of two-dimensional scaling which was given in [4]. We give an efficient algorithm for the scaled matching problem under the new definition of scaling: For a text $T$ of length $n$ and a pattern $P$ of length $m$, the algorithm finds in $T$ all occurrences of $P$ scaled to any real value in time $O(n \log m + \sqrt{nm^{3/2}} \sqrt{\log m})$.

**Roadmap:** In section 2 we give the necessary preliminaries and definitions of the problem. In section 3 we present a simple algorithm that straightforwardly finds the scaled matches of the pattern in the text. In section 4 we present our main result, namely, the efficient algorithm for real scaled matching.

## 2 Scaled Matching Definition

Let $T$ and $P$ be two strings over some finite alphabet $\Sigma$. Let $n$ and $m$ be the lengths of $T$ and $P$, respectively.

**Definition 1.** A pixel is an interval $(i-1, i]$ on the real line $\mathbb{R}$, where $i$ is an integer. The center of a pixel $(i-1, i]$ is its geometric center point, namely the point $i - 0.5$.

**Definition 2.** Let $r \in \mathbb{R}$, $r > 1$. The $r$-ary pixel array for $P$ consists of $m$ intervals of length $r$, which are called $r$-intervals. The $i$-th $r$-interval is $[(i-1)r, ir]$. Each interval is identified with the value from $\Sigma$: The $i$-th interval is identified with the $i$-th letter of $P$. For each pixel center that is inside some $r$-interval, we assign the letter that corresponds to that interval. The string obtained by concatenating all the letters assign to the pixel centers from left to right is denoted by $P^r$, and called $P$ scaled to $r$. 