

## Parallel string matching for image matching with prime method

Chinta Someswara Rao<sup>1</sup>,

<sup>1</sup>Assistant Professor, Dept of CSE, SRKR Engineering College, Bhimavaram, AP, INDIA-534 204.

---

**Abstract:-** String matching play the much more attention in the recent days because of its importance in several areas like retrieval of large text data, image data , mining etc.,. For this purpose, several researchers proposed solution to these problems, but still there is a scope to develop new techniques, especially in image matching. For this purpose, in his paper, we have proposed a method for image search based on the prime method. The adopted method uses a sliding window approach, in this approach each sub image clipped by narrow window is converted to code vector and these code vectors are used in image matching. Our results showed that this method performs well in spatially, computationally and efficiently.

**Keywords:-** String matching, parallel approach, WWW, image searching.

---

### I. INTRODUCTION

String matching is a special case of retrieval, where the pattern is described by a finite sequence of symbols (or alphabet). It consists of finding one or more generally all the occurrences of a short pattern  $P=P[0]P[1]...P[m-1]$  of length  $m$  in a large text  $T=T[0]T[1]...T[n-1]$  of length  $n$ , where  $m, n > 0$  and  $m \leq n$ . Both  $P$  and  $T$  are built over the same alphabet  $\Sigma$ . Several experiments on string matching algorithms have already been reported [1-8]. In this paper, we have proposed a novel method for converting multi dimensional descriptor into encoded representation. A set of integers has retrieved that can be used as the code for the purpose of sub-image detection and hence it is called code vector of the sub image.

### II. LITERATURE

In the literature we provide the different techniques proposed by several researchers, which will give the support to develop the new methods.

Terasawa et.al [9] presents a fast appearance-based full-text search method for historical newspaper images. Since historical newspapers differ from recent newspapers in image quality, type fonts and language usages, optical character recognition (OCR) does not provide sufficient quality. Instead of OCR approach, we adopted appearance-based approach, that means we matched character to character with its shapes. Assuming proper character segmentation and proper feature description, full-text search problem is reduced to sequence matching problem of feature vector. To increase computational efficiency, we adopted pseudo-code expression called LSPC, which is a compact sketch of feature vector while retaining a good deal of its information. Experimental result showed that our method can retrieve a query string from a text of over eight million characters within a second. In addition, we predict that more sophisticated algorithm could be designed for LSPC. As an example, we established the Extended Boyer-Moore-Horspool algorithm that can reduce the computational cost further especially when the query string becomes longer.

Terasawa et.al [10] proposed a novel scheme for representing character string images in the scanned document. We converted conventional multi-dimensional descriptors into pseudo-codes which have a property that: if two vectors are near in the original space then encoded pseudo-codes are 'semi equivalent with high probability. For this conversion, we combined locality sensitive hashing (LSH) indices and at the same time we also developed a new family of LSH functions that is superior to earlier ones when all vectors are constrained to lie on the surface of the unit sphere. Word spotting based on our pseudo-code becomes faster than multi-dimensional descriptor-based method while it scarcely degrades the accuracy.

Tan et.al [11] propose a method for text retrieval from document images without the use of OCR. Documents are segmented into character objects. Image features, namely the vertical traverse density (VTD) and horizontal traverse density (HTD), are extracted. An n-gram-based document vector is constructed for each document based on these features. Text similarity between documents is then measured by calculating the dot product of the document vectors. Testing with seven corpora of imaged textual documents in English and Chinese as well as images from the UW1 (University of Washington 1) database confirms the validity of the proposed method Drira et.al [12] enhancing diffusion filter is proposed for which new constraints formulated from the Perona-Malik equation are added. The new diffusion filter, driven by local tensors fields, takes benefit from both of these approaches and avoids problems known to affect them. This filter reinforces character discontinuity and eliminates the inherent problem of corner rounding while smoothing. Experiments conducted

on degraded document images illustrate the effectiveness of the proposed method compared to another anisotropic diffusion approaches. A visual quality improvement is thus achieved on these images. Such improvement leads to a noticeable improvement of the OCR system's accuracy proven through the comparison of OCR recognition rates before and after the diffusion process.

In this paper, we proposed approach uses code vectors that consist of four values obtained by applying modulo division using a prime number over each row of the entire small region.

### III. METHODOLOGY

The proposed work has been partitioned into several distinct steps:

- Image Segmentation
- Vector Extraction
- Code Generation
- String Searching of the Code.
- Image Matching

Among these steps, (A), (B) and (C) are the preprocessing steps and (D) and (E) are the searching and matching phases respectively.

In phase (D) vector string is searched into the vector code of the given image and in phase (E) search image vector is compared to the corresponding portion of the given image vector, if phase (D) has found successfully.

#### A. Image Segmentation:

First the image is converted into a matrix and then segmentation is done. The segmented image was divided into small regions, which were used to form vectors, e. g. first segment contain  $a_{ij}$  for  $i=1, 2, 3, 4 \dots m$  and  $j=1, 2, 3, 4 \dots n$  where  $a_{ij}$  is the element of original image matrix.

#### B. Vector Extraction:

Each segmented image was fed to vector extraction. The method to obtain vectors is as follows: A small matrix (segmented image) is used to generate the vector. The matrix is recomputed by dividing all of its elements by a prime number (in our case, the prime number is 101) and then taking the remainder, e.g.  $a_{ij} = \text{remainder of } a_{ij}/p$  for  $i=1, 2, 3, 4 \dots m$ , and  $j=1, 2, 3, 4 \dots n$ . And  $p$  is the prime number. Now each row of the small matrix will form one vector.

#### C. Code Generation:

In this process, each vector is converted into its relevant code. To be more exactly, suppose the segmented image was divided into small regions, the code vector is generated corresponding to each row of the small region. This code vector consists of the four values corresponding to the four rows of the small region. These vector codes are obtained by taking the remainder of sum of the elements of each row vector of the small region. i.e. if  $V=[v_1, v_2, v_3, v_4 \dots V_n]$  is a vector then  $v_i = (\sum a_{ij})/p$  for  $i=1, 2, 3, 4 \dots m$  and  $j=1, 2, 3, 4 \dots n$ . Stores these code values into an array so that these code values can be further used in searching.

#### D. String Matching of the Code:

The string matching [14] can be defined as: the text is an array  $T[1 \dots n]$  of length  $n$  and the pattern is an array  $P[1 \dots m]$  of length  $m (\leq n)$ . We further assume that the elements of  $P$  and  $T$  are the characters drawn from a finite alphabet  $\Sigma$ . For example, we may have  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  or  $\Sigma = \{a, b, c, d, e, f\}$  etc. The character arrays  $P$  and  $T$  are often called strings of characters. All string matching algorithm scan the text with the help of a window which is equal to the length of the pattern. The first process is to align the left end of the pattern with the text window and then compares the corresponding characters of the window and the pattern. This process is known as an attempt. After a whole match or a mismatch of the pattern, the text window is shifted in the forward direction until the window is positioned at the  $(n-m+1)$  position of the text. This approach is the naive brute-force algorithm. In brute force algorithm, window is shifted to the right by one character after an attempt. This is the most primitive method of sequential scanning, which check all position in the text  $T$  whether an occurrences of the pattern  $P$  starts there or not. This can be implemented in complexity  $O((n-m+1)m)$ . We implement it work on our code vectors. Each time the code vector of searching image is compared against the code vector of a piece of given image, such that the size of the piece of the given image is exactly the same as the size of the searching image. In case of a mismatch we select next piece of given image whose size is same as the size of the searching image. In case of a match the control is passes over to next, image matching, step.

#### E. Image Matching:

When the code-vector of the searching image is matched with the code-vector of a piece of the given image, then the searching image matrix is compared to the corresponding portion of the given image matrix, and each pixel of the searching image is compared with each pixel of the image. If each pixel is matched, then the

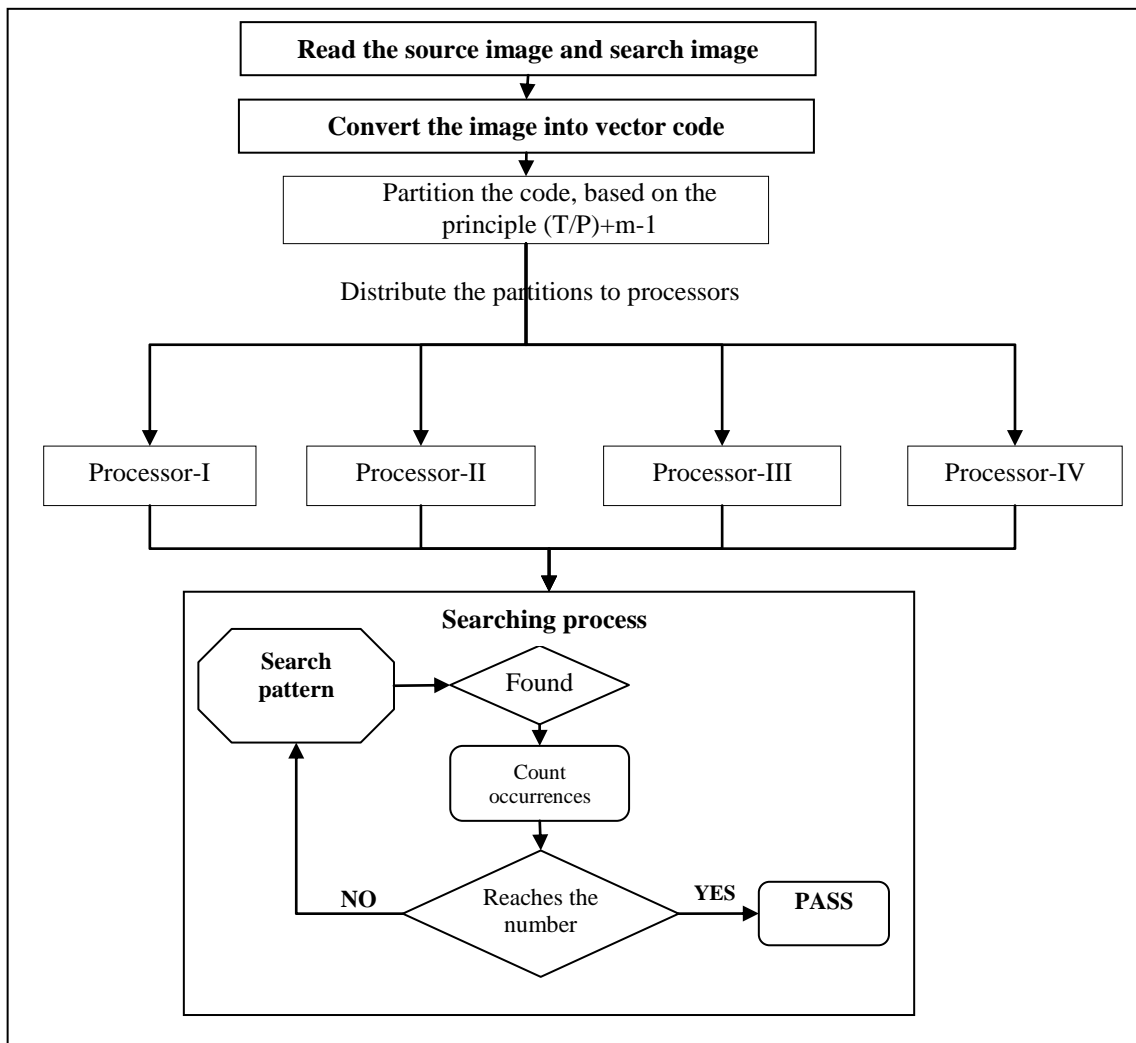
portion of the sub-image found, will contain the searching image. This step compares the images only if previous step successfully passed.

**F. Algorithm description**

The algorithm makes use of elementary number-theoretic notations such as the equivalence of two numbers modulo a third number. In our algorithm each character is a decimal number, and compute values by modulo  $p(=101)$ , where  $p$  is a prime number. The values of the image pixels lie between 0 - 255. There are 54 prime numbers between 0 and 255. And the mid prime number is 103(if lower median is selected). We selected 101 as the value of  $p$ , which is near to mid prime number. Since  $n$  modulo 101 gives a number between 0 to 100, where  $n$  is any integer  $\in [0, 255]$ .

**IV. PARALLEL APPROACH**

The proposed parallel algorithm makes use of the message-passing parallelism model by using  $p$  processors. The following assumptions for the model of communications in the parallel computer are made. The parallel computer comprises a number of nodes. Each node comprises one or several identical processors interconnected by a switched communication network and it is depicted in 1. The links between two nodes are full-duplex and single-ported. Here the proposed system use the functional decomposition, in which the initial focus is on the computation that is to be performed rather than on the data manipulated by the computation. It assumes that both image text  $t$  and pattern  $p$  are stored locally on each processor. This can be done by using a one-to-all broadcast operation as shown in Fig 1













**Fig 1. Parallel approach for extended encoding method**

**V. CASE STUDY**

The size of the Source image is  $m$ -by- $n$  pixels and let the size of the searching image be  $x$ -by- $y$  pixels. The Pre-Processing time and the Matching time of the proposed algorithm are calculated for the variable

size of the search image and the size of the source image is fixed (size=16x16) in case of the source image. The spurious hits are also calculated for the proposed algorithm, to obtain the accuracy of the proposed algorithm. The spurious hits are fake code values of the search image that matches with the code values of the source image, but in actual the search image is not matched within the source image. The results are shown in Table 1 and Table 2. From the fig 2 we clear say that the proposed approach have reaches the require criteria's.

Source image	Search image	Result
		Pass
		Pass
		pass
		Fail
		Fail

Size of the Source image	Size of the Search image	Matching Time
16x16	16x16	0.436
12x12	12x12	0.428
8x8	8x8	0.433
4x4	4x4	0.432

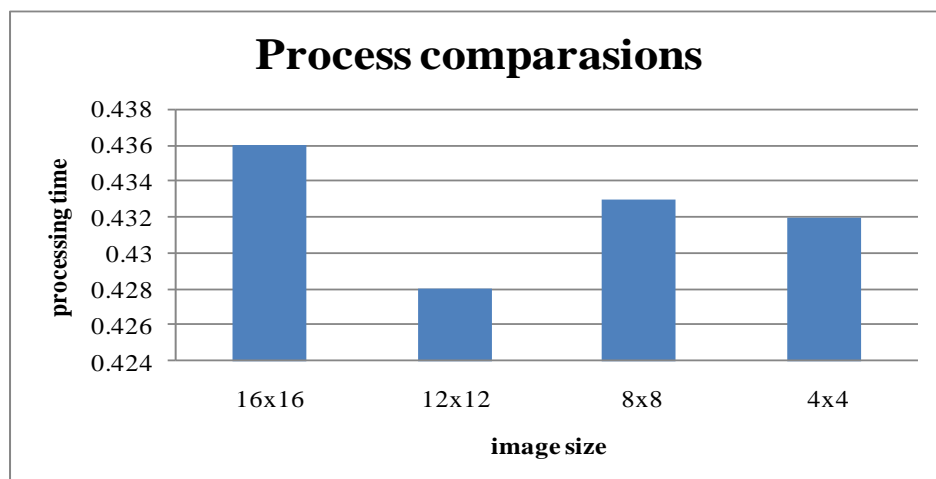


Fig 2 process comparisions

## VI. CONCLUSION

In this paper we proposed a novel image searching method for images has proposed with prime approach. The source and searching image are successfully read and stored. Segmentation is done successfully and each segment is also converted into code vectors. Parallel string search method is applied successfully and finally image is compared. The proposed method increases the search speed with exact matching. In future we will try to apply this approach other sources and downstream applications.

## REFERENCES

- [1]. Chinta Someswararao, K Butchiraju, S ViswanadhaRaju, "Recent Advancement is Parallel Algorithms for String matching on computing models - A survey and experimental results", LNCS, Springer, pp.270-278, ISBN: 978-3-642-29279-8, 2011.
- [2]. Chinta Someswararao, K Butchiraju, S ViswanadhaRaju, "PDM data classification from STEP- an object oriented String matching approach", IEEE conference on Application of Information and Communication Technologies, pp.1-9, ISBN: 978-1-61284-831-0, 2011.
- [3]. Chinta Someswararao, K Butchiraju, S ViswanadhaRaju, "Recent Advancement is Parallel Algorithms for String matching - A survey and experimental results", IJAC, Vol 4 issue 4, pp-91-97, 2012.
- [4]. Simon Y. and Inayatullah M., "Improving Approximate Matching Capabilities for Meta Map Transfer Applications," Proceedings of Symposium on Principles and Practice of Programming in Java, pp.143-147, 2004.
- [5]. Chinta Someswararao, K Butchiraju, S ViswanadhaRaju, "Parallel Algorithms for String Matching Problem based on Butterfly Model", pp.41-56, IJCST, Vol. 3, Issue 3, July – Sept, ISSN 2229-4333, 2012.
- [6]. Chinta Someswararao, K Butchiraju, S ViswanadhaRaju, "Recent Advancement is String matching algorithms- A survey and experimental results", IJCIS, Vol 6 No 3, pp.56-61, 2013.
- [7]. Chinta Someswararao, " Parallel String Matching Problems with Computing Models - An Analysis of the Most Recent Studies", International Journal of Computer Applications , Vol.76(15), pp.7-25, Published by Foundation of Computer Science, New York, USA, 2013.
- [8]. Chinta Someswararao, "Parallel String Matching with Multi Core Processors-A Comparative Study for Gene Sequences", Global Journal of Computer Science and Technology, Vol-13, Issue-1, pp.27-41, 2013.
- [9]. Terasawa, K, "A Fast Appearance-Based Full-Text Search Method for Historical Newspaper Images", International Conference on Document Analysis and Recognition (ICDAR), pp. 1379-1383, 2011
- [10]. Terasawa, K. , Locality Sensitive Pseudo-Code for Document Images, International Conference on Document Analysis and Recognition, pp.73-77, 2007.
- [11]. Tan, C.L, Imaged document text retrieval without OCR, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.838-844, 2002.
- [12]. Drira, F , Document Images Restoration by a New Tensor Based Diffusion Process: Application to the Recognition of Old Printed Documents, International Conference on Document Analysis and Recognition, pp.321-325, 2009.