International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 10 – No.10, May 2016 – www.ijais.org

Quad-search: Modified Binary Search Algorithm for Efficient Fingerprint Biometrics Identification in Multicore Architecture

Oriola Oluwafemi
Department of Computer Science
Adekunle Ajasin University
Akungba Akoko, Ondo State
Nigeria

Orimoloye Segun Michael
Department of Computer Science
Adekunle Ajasin University
Akungba Akoko, Ondo State
Nigeria

ABSTRACT

Linear Search Algorithm is often applied to Fingerprint Biometrics Identification in traditional computing. With the advent of multicore systems, more attention must be placed on Binary Search Algorithms for efficient Fingerprint Biometric Identification in large databases. However, Binary Search Algorithms can conveniently be parallelized for two cores. Therefore, this study assumes that a variant of Binary Search Algorithm, which supports multi-parallelization, will take better advantage of advances in multicore systems and improve efficiency of Biometric Identification. This paper hence presents a Modified Binary Search algorithm known as Quad-search for Biometric Identification. The Fingerprint Identification Algorithm is developed by splitting the fingerprint array at the midpoint and parallelizing the search in each split to form four sub-arrays. To evaluate the proposed algorithm, a real-life Fingerprint Biometric Identification System of tested fingerprint images is used. The execution time of the Modified Binary Search Algorithm is compared with both Binary Search and Linear Search Algorithms in an Intel Quadcore Architecture. The results show that at near, middle, and far fingerprint positions in the database, the Modified Search Algorithm is executed at lower time compared to both the Parallelized Binary Search Algorithm and Linear Search Algorithm, but the parallelized Binary Search Algorithm executes at lower time at mid fingerprints locations compared to Linear Search Algorithm. The running time is estimated as O(Log(logN)) The Modified Binary Search Algorithm resulted in efficient Fingerprint Biometrics Identification in multicore architecture.

General Terms

Algorithms

Keywords

Biometrics, Fingerprint, Identification, Modified Binary Search, Multi-parallelized

1. INTRODUCTION

Among biometric traits, fingerprint is widely accepted by people because of its uniqueness and immutability. Fingerprint Biometrics is the oldest method which has been successfully used in numerous applications [5].

A fingerprint is made of a series of ridges and furrows on the surface of the finger. The pattern of ridges and furrows as well as the minutiae points can determine the uniqueness of a fingerprint.

Fingerprint Biometric Recognition Systems involve both verification and identification processes. Identification searches for the target person while verification confirms that the target person is a particular person. The processing time of identification usually depends on the number of images registered for image matching (called "templates"). Most fingerprint Identification systems use minutiae matching point because of its efficiency [3, 4, 6, 9] However, there are still rooms for improved efficiency in the aspect of Fingerprint Biometric Identification.

Searching is synonymous to Fingerprint Biometric Identification. The Search Algorithms include Linear Search Algorithm, Binary Search Algorithm and their variants [1, 2, 3, 4]. Theoretically, Binary Search Algorithm's running time is O(log N) while Linear Search Algorithm's running time is O(N). This means Binary Search Algorithm is more efficient than Linear Search Algorithm. But in practice, the manner in which the algorithm is written and the type of machine influence the performance.

Until recently that multicore systems became popular, manufacturers employed Linear Search Algorithms for Fingerprint Biometric Identification. Hence, scientists and researches focus their attention on ways to improve the Linear Search algorithms for efficiency. Some of the methods used to improve Linear Search Algorithm include Maeda et al. [6], Pan et al. [8], Pan et al. [9] and Meersman et al. [7].

With Multicore Architecture, parallelized algorithms are encouraged to solve the high performance problems. In this wise, Binary Search Algorithm is taking centre stage [1, 2] However, it runs efficiently on two cores, which is still computationally expensive. We presume that to benefit greatly from the multicore systems, an algorithm that can be multi-parallelized and run on multicore rather than dual core efficiently is required. Therefore, this paper presents Modified Binary Search Algorithm called Quad-search. The paper evaluates the algorithm in a Multicore Architecture and compares the performance with the applications of Linear and Binary Search Algorithms.

2. RELATED WORKS

Most of the existing works on efficient Fingerprint Biometric Identification have focused on improving Linear Search algorithms. Maeda et al. [6] propose an identification algorithm (MSM) that reduces the number of image comparisons in the linear search. The main idea of the algorithm is that the order of templates compared with the query image is decided according to the results of the conducted comparisons and the similarities between templates. They report that the average number of comparisons is experimentally O(√N) instead of O(N). Recently, fingercode has been demonstrated to be an effective fingerprint biometric identification scheme, in which both local and global details of fingerprints are captured in codes. The existing works in this aspect have applied Vector Quantization [7, 8, 9]. The results have shown that they are computationally efficient and accurate. However, works to improve the efficiency of Binary Search Algorithm are lacking.

In similar Biometrics, [10] describes a new method to authenticate individuals based on palmprint identification and verification. Firstly, a comparative study of palmprint feature extraction is presented. The concepts of texture feature and interesting points are introduced to define palmprint features. A texture-based dynamic selection scheme is proposed to facilitate the fast search for the best matching of the sample in the database in a hierarchical fashion. The global texture energy, which is characterized with high convergence of inner-palm similarities and good dispersion of inter-palm discrimination, is used to guide the dynamic selection of a small set of similar candidates from the database at coarse level for further processing. An interesting point based image matching is performed on the selected similar patterns at fine level for the final confirmation. The experimental results demonstrate the effectiveness and accuracy of the proposed method. Baba and Egawa [1] prepare the order of image comparison statically instead of dynamic O(N) computation of traditional Linear Search. The templates are fixed by deciding the order in advance of identification. The algorithm reduced the number of image comparison. Furtherance to the previous work, [2] applies MSM to compute the distance dynamically and the space changes for each comparison. The template is selected at the point of identification to examine the effect of the selected templates on the performance of the algorithm. The novel algorithm reduces the number of image comparisons as the previous version and does not worsen the accuracy by experiments with palmprint images.

Bayram et al. [4] presents a binary search tree (BST) data structure based on group testing to enable the fast identification. The results on the real-world and simulation data show that with the proposed scheme, major improvement in search time can be achieved.

3. MODIFIED BINARY SEARCH ALGORITHM (QUAD-SEARCH)

Binary search is a fast search algorithm with run-time complexity of O(log n). The algorithm follows the principle of divide and conquer. For this algorithm to work properly, the data collection should be in sorted form. Binary Search Algorithm sorts the array of fingerprints in ascending order. It finds the midpoint of the array and divides the array into two subsets. The first subset containing the minimum position of the original array is the lower subset while the other subset containing the maximum position of the original array is the upper subset. In the Binary Search algorithm pseudocode below, array A(1, 2, 3, 4, …, N) has two subsets A_L(1,2, ….,k) and A_U (k+1, …, N-1, N) where the Midpoint = k

```
BINARYSEARCH ( A[1, 2, …, N ], key, imin, imax)
{
  // test if array is empty
if (imax < imin)
// set is empty, so return value showing not found
   return KEY_NOT_FOUND;
   else
     {
    //calculate midpoint to cut set in half
        imid = midpoint(imin, imax)
        // three-way comparison
        if (A[imid] > key)
        // key is in lower subset
          return BINARYSEARCH(A, key, imin, imid - 1)
          else if (A[imid] < key)
          // key is in upper subset
            return BINARYSEARCH(A, key, imid + 1, imax);
            else
            // key has been found
               return imid;
            }
          }
       }
   }
}
```

The Modified Binary Search Algorithm sorts the array of fingerprints in ascending order. It finds the midpoint of the array and divides the array along the midpoint into lower subset (L) and upper subset (U). Next, each sub-array is divided into two; the first contains the odd positions while the other contains even positions. Hence, four subsets are formed.

Given that the original array or set is A(1,2, 3, …., N) then the four subsets formed are:

A_{L_ODD}(1, 3, ….,k-1)

A_{L_EVEN}(2, 4, ….,k)

A_{U_ODD} (k+1, k+3, …, N-1)

A_{U_EVEN} (k+2, k+4, …, N)

In the Modified Binary Search Algorithm pseudocode presented below, three keys are created Midpoint, k; Prefix of Midpoint, p; and Suffix of Midpoint, s.

```
mbinarysearch( A[ 1,2, 3, …, N], key, imin, imax)
{
 // test if array is empty
if (imax < imin)
// set is empty, so return value showing not found
   return KEY_NOT_FOUND;
   else
```

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 10 – No.10, May 2016 – www.ijais.org

```
    {
    //calculate midpoint to cut set in half
     imid = midpoint(imin, imax);
     // get the prefix of midpoint
     p_imid =[ imid] - 1;
     // get the suffix of midpoint
     s_imid =[ imid] + 1
     // five-way comparison
    if (A[imid] > key)
    // key is in lower subset
       return mbinarysearch(A, key, imin, imid - 2)
       else if (A[imid] < key)
       // key is in upper subset
         return mbinarysearch(A, key, imid + 2, imax)
         elseif (A[p_imid] > key)
         // key is in lower subset
           return mbinarysearch(A, key, imin, p_imid- 2)
           else if (A[s_imid] < key)
           // key is in upper subset
               return mbinarysearch(A,key, s_imid + 2, imax)
               else
               // key has been found
           return imid;
           }
         }
       }
     }
   }
  }
 }
```

4. EXPERIMENTAL EVALUATION

The algorithm is evaluated by setting up a fingerprint biometric identification system on an Intel Quadcore HP System. Unique fingerprints images of 1000 persons in a Nigeria Campus Attendance Repository are selected and inserted into MS-SQL database. The fingerprint image sample of a person is selected and placed in 10 different positions. In each simulation, the position of the fingerprint sample is recorded. When the result of the identification is generated, the process is terminated and the time taken for identification of the person's identity is recorded. This takes place before fingerprint image is placed in another position for identification. Fig. 1 presents the model of Fingerprint Biometric System.

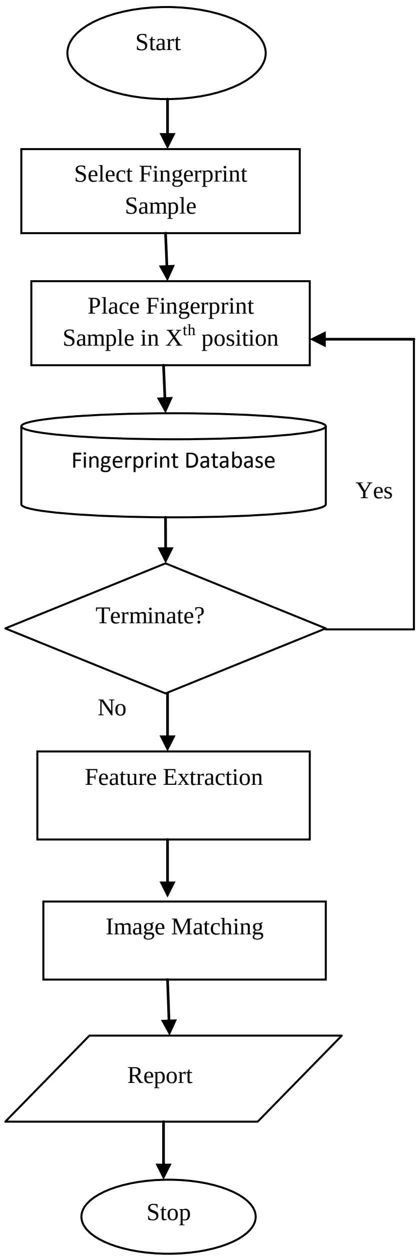

Fig 1. Model of Fingerprint Biometric Identification

The Fingerprint Biometric System procedures include searching, feature extraction, image matching and report generation. Table 1 presents the time taken in seconds to identify the selected fingerprint sample. Table 2 and Fig. 2 present the comparisons of time taken to identify for the Modified Binary Search, Binary Search and Linear Search Algorithms.

Table 1. Time Taken in Seconds to Identify the Selected Fingerprint Sample

Simulation	Position	Modified Binary Search Algorithm
#1	20th	90
#2	101st	459
#3	256th	1152

#4	350th	1575
#5	437th	1971
#6	568th	1953
#7	683th	1431
#8	760th	1089
#9	839th	729
#10	976th	117

Table 2. Comparisons of Time Taken to Identify for the Modified Binary Search, Binary Search and Linear Search Algorithms

Simu-lation	Position	Modified Binary Search	Linear Search	Binary Search
#1	20th	90	180	180
#2	101st	459	909	909
#3	256th	1152	2304	2304
#4	350th	1575	3150	3150
#5	437th	1971	3933	3933
#6	568th	1953	5112	3897
#7	683th	1431	6147	2862
#8	760th	1089	6840	2169
#9	839th	729	7551	1458
#10	976th	117	8784	225

It should be noted that the first simulation is the nearest position, 5th simulation is the mid position while 10th simulation is the farthest position.

From the simulation results presented in Table 1, the time taken increases by an average fraction of 1.5 over 100 steps within the first subset (1-5 simulations) and diminishes by an average fraction of 1.5 over 100 steps within the second subset (6-10 simulations) for Modified Binary Search Algorithm. The time taken is not above 100, 2000 and 150 seconds at nearest, mid and farthest positions. In the results of the comparisons in Table 2, the time taken by Linear Search Algorithm increases by an average fraction of 2 over 100 steps across the full set of simulations. The maximum time taken is above 100, 5000 and 8000 seconds at nearest, mid and farthest positions respectively.

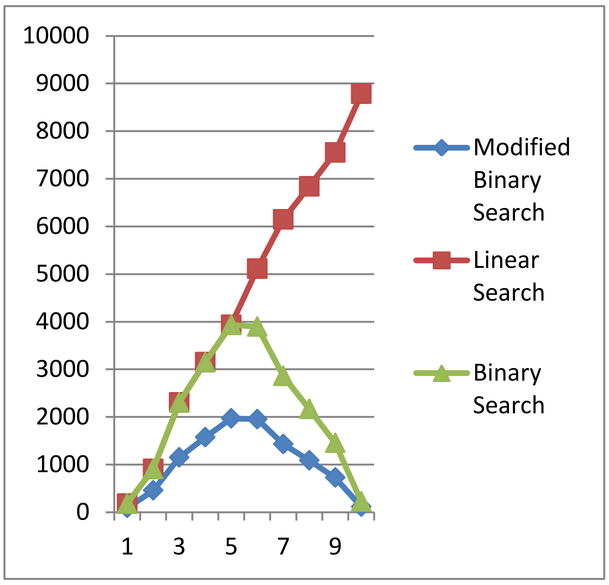

Fig.2. Line Graph for Comparison of Time Taken to Identify for the Modified Binary Search, Binary Search and Linear Search Algorithms

For the Binary Search Algorithm, the time taken increases by an average fraction of 2 over 100 steps within the first subset (1-5 simulations) and diminishes by an average of 1.5 over 100 steps within the second subset (6-10 simulations). The maximum time taken is above 100, 3000 and 200 seconds at nearest, mid and farthest positions respectively. The line graph in Fig. 2 shows that both Binary Search and Linear Search Algorithms completed identification process in the first 5 simulations at the same time but the former completed the processing before the later in the last 5 (6-10) simulations. The Modified Binary Search Algorithm however completed the identification process at a much more reduced time than Binary Search and Linear Search Algorithms in all the simulations.

5. CONCLUSION

Fingerprint Biometric Identification incurs huge computational costs especially in large database. Traditional computing has motivated improvement of Linear Search Algorithms. With popularity of multicore system and parallel computing, Binary Search Algorithm with theoretical running time of O(LogN) therefore becomes a natural choice. However, since a lot of cost is expended on extraction and matching, the focus of this paper is to improve Binary Search Algorithm further by developing Quad-search (Modified Binary Search Algorithm) to limit extra cost of searching through database.

The results show that the Modified Binary Search Algorithm improved the efficiency of Fingerprint Biometric Identification by reducing the time taken to identify the near, mid and far positioned fingerprint image sample by about half of the time taken by Binary Search Algorithm. The running time of the Modified Binary Search Algorithm is therefore estimated as O(Log(logN)).The results also show that the parallelized Binary Search Algorithm executes at lower time at mid fingerprints positions compared to Linear Search Algorithm. In future, the error rates of the Fingerprint Biometric Identification for the Modified Binary Search Algorithm, Binary Search Algorithm and Linear Search Algorithm will be examined.

6. REFERENCES

[1] Baba, K. and Egawa S. 2013, "A data structure for efficient biometric identification," in Proc. of the 2013 International Conference on Information and Communication Technology (ICT-EurAsia'13), Yogyakarta, Indonesia, LNCS, vol. 7804. Springer-Verlag, March 2013, pp. 528–533.

[2] Baba, K. and Egawa, S. 2013. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 2, pp. 97-103

[3] Bayram, S. 2012. Applications of Multimedia Forensics. PhD Thesis. Polytechnic Institute of New York University, United States of America.

[4] Bayram, S., Sencar, H. T. and Memon, N. 2014. Sensor Fingerprint Identification Through Composite Fingerprints and Group Testing IEEE Transactions on Information Forensics and Security .Volume 10, Issue 3.

[5] Jain, A.K., Prabhakar, S., Hong, L. and Pankanti, S.2000. Filterbank-based fingerprint matching. IEEE Trans on Image Processing, 2000, 9(5): 846-859.

[6] Maeda, T., Matsushita, V., and Sasakawa, K. 2001. Identification algorithm using a matching score matrix. IEICE Transactions on Information and Systems, vol. E84-D, no. 7, pp. 819–824, 2001.

[7] Meersman, R., Tari, Z., and Herrero, P. 2006. An Efficient Algorithm for Fingercode-Based Biometric Identification OTM Workshops 2006, LNCS 4277, pp. 469–478, 2006.

[8] Pan, Z., Kotani, K., and Ohmi, T., 2005. Improved fast encoding method for vector quantization based on subvector technique. IEEE International Symposium on Circuits and Systems, 2005: 6332-6335.

[9] Pan, Z., Kotani, K., and Ohmi. 2004. "A memory-efficient fast encoding method for vector quantization using 2-pixel-merging sum pyramid", 2004 IEEE Int'l Conf on Acoustics, Speech and Signal Processing, 2004, 3: 669-672.

[10] You, J., Li, W., and Zhang, D. 2001. Hierarchical palmprint identification via multiple feature Extraction. Pattern Recognition 35 (2002) 847–859.