

American Journal of Advanced Computing

AJAC



A publication of

SMART

SOCIETY FOR MAKERS, ARTISTS, RESEARCHERS AND TECHNOLOGISTS

6408 ELIZABETH AVENUE SE, AUBURN, WA 98092, USA

U.S. ISSN CENTRE APPROVED

| Page No. | CONTENT |
|----------|---|
| 5 | <p>An Efficient Technique for Finding Longest Common Subsequence of DNA Sequences</p> <p><i>Molecular biologists rely very heavily on computer science algorithms as research tools. The process of finding the longest common subsequence of two DNA sequences has a wide range of applications in modern bioinformatics. Genetics databases can hold enormous amounts of raw data, for example the human genome consists of approximately three billion DNA base pairs. The processing of this gigantic volume of data necessitates the use of extremely efficient string algorithms. This paper introduces a space and time effective technique for retrieving the longest common subsequence of DNA sequences.</i></p> <p>DOI: doi.org/10.15864/ajac.1101 Tamal Chakrabarti and Devadatta Sinha</p> |
| 10 | <p>A Comparative Study of Different Techniques for Prime Testing in Implementation of RSA</p> <p><i>The RSA cryptosystem, invented by Ron Rivest, Adi Shamir and Len Adleman was first publicized in the August 1977 issue of Scientific American. The security level of this algorithm very much depends on two large prime numbers. The large primes have been taken by BigInteger in Java. An algorithm has been proposed to calculate the exact square root of the given number. Three methods have been used to check whether a given number is prime or not. In trial division approach, a number has to be divided from 2 to the half the square root of the number. The number will be not prime if it gives any factor in trial division. A prime number can be represented by $6n \pm 1$ but all numbers which are of the form $6n \pm 1$ may not be prime. A set of linear equations like $30k+1$, $30k+7$, $30k+11$, $30k+13$, $30k+17$, $30k+19$, $30k+23$ and $30k+29$ also have been used to produce pseudo primes. In this paper, an effort has been made to implement all three methods in implementation of RSA algorithm with large integers. A comparison has been made based on their time complexity and number of pseudo primes. It has been observed that the set of linear equations, have given better results compared to other methods.</i></p> <p>DOI: doi.org/10.15864/ajac.1102 Satyendra Nath Mandal, Kumarjit Banerjee and Sonjay Kumar Das</p> |
| 17 | <p>Coordinate based Routing Protocol for Mobile Networks: A Fuzzy Logic Approach</p> <p><i>An implementation of the co-ordinate based routing protocol for a mobile network is proposed in this paper with the Fuzzy logic concept. The rules for mapping between cell number and corresponding co-ordinates are defined. A flexible sense of membership function of elements supported by Fuzzy logic is used here. All possible routing paths can be enumerated in a simple way. The proposed method is one of the simpler than other techniques reported so far.</i></p> <p>DOI: doi.org/10.15864/ajac.1103 Parag Kumar Guha Thakurta and Paulami Dey</p> |
| 22 | <p>A Novel and Flexible Criterion to Improve Data Transmission in Clustering Protocols in WSNs</p> <p><i>In this paper a new criterion called Energy-Cost Function is presented according to which in each round the energy cost for any individual node is calculated. When transmitting their data, the nodes make decisions based on this very cost function. In case the nodes decides that it will cost a lower amount of energy transmitting the data to the sink by itself rather than by the cluster-head to the sink, then the node transmits the data directly. In this way the cluster overload is reduced and both the network lifetime and instability period is boosted. Based on the conditions of the problem, cost function parameters are variable and since all the decisions are made locally, network scalability potential is retained. Simulation results show that the network lifetime or its instability period based on the cost function employed will be improved up to 40% in relation to the LEACH protocol.</i></p> <p>DOI: doi.org/10.15864/ajac.1104 Mohsen Ataai and Esmaeil Zeinali Kh.</p> |

| | |
|----|---|
| 28 | <p>Robust Synchronization of Duffing System Using Integral Action in Backstepping Design</p> <p><i>Backstepping is a realistic nonlinear control design algorithm based on Lyapunov design approach; therefore, it automatically ensures the convergence of the regulated variable to zero. In this paper, it has been proposed for robust synchronization of Duffing chaotic systems. Integral action is being used to enhance the control action of the controller in steady state against the disturbances. The salient feature of the design is that the derived control doesn't contain any derivative terms; consequently it simplifies the controller realization. The effectiveness of the proposed controller has been demonstrated in simulation studies. The performance of the controller has been evaluated not only based on its synchronizing ability but also the disturbance rejection ability of the controller has been verified.</i></p> <p>DOI: doi.org/10.15864/ajac.1105 Shubhobrata Rudra, Ranjit Barai, Madhubanti Maitra, Rupam Dewan, Paramita Mandal, Dharmadas Mandal and Kishorekumar Kowdiki.</p> |
| 33 | <p>Detection of Intruders and Flooding in VoIP using IDS, Jacobson Fast and Hellinger Distance Algorithms</p> <p><i>VoIP services are becoming increasingly a big competition to existing telephony services (PSTN). Hence, the need arises to protect VoIP services from all kinds of attacks that target network bandwidth, server capacity or server architectural constrains. SIP Protocol is used for VoIP connection establishment. It works based on either TCP or UDP Protocols. This protocol structure is almost as same as HTTP Protocol, i.e. for every request there will be some response, even though the request is invalid. HTTP Protocol is prone to flooding attacks, like SYN-Flood attack. Because of Session Initiation Protocol (SIP) is same as HTTP, SIP is also prone to Flooding attacks. The proposed Intrusion Detection System (IDS) is used to detect the intruders in telephony system. Genetic algorithm is used to recognize the authorized user. VoIP Flood Detection System (VFDS) is aimed to detect TCP Flooding attacks and SIP Flooding attacks on SIP devices using Jacobian Fast and Hellinger distance algorithms. The Jacobian Fast Algorithm fixes the threshold limit and Hellinger distance calculation is a statistical anomaly based algorithm uses to detect deviation in traffic.</i></p> <p>DOI: doi.org/10.15864/ajac.1106 Rahul Ari, Suresh Kumar and Prashanth S.K.</p> |
| 38 | <p>A Mathematical Model of Blood Flow in a Catheterized artery with Multiple Stenoses</p> <p><i>A mathematical model is developed in this investigation for studying the axi-symmetric flow of blood through a catheterized artery with multiple stenoses. Consideration of Newtonian character of blood is described following the report of Young (1968) and Srivastava (2009) with the appropriate constitutive equation governing the flow. The boundary conditions appropriate to the problem under study are the standard no slip conditions at the artery and the catheter wall. Analytical expressions for impedance (flow resistance), the wall stress distribution in the stenotic region and the shear stress at the stenosis throat in their non dimensional form are derived by using the model. The derived expressions are computed numerically and the results are presented graphically for different values of the rheological and other parameters. The study provides an insight into the effects of catheter radius and stenosis height on impedance, wall stress distribution in the stenotic region and the shear stress at the stenotic throat.</i></p> <p>DOI: doi.org/10.15864/ajac.1107 Biswadip Basu Mallik and Dr.Saktipada Nanda</p> |
| 43 | <p>A Computer Vision Framework for Automated Shape Retrieval</p> <p><i>With the increasing number of images generated every day, textual annotation of images for image mining becomes impractical and inefficient. Thus, computer vision based image retrieval has received considerable interest in recent years. One of the fundamental characteristics of any image representation of an object is its shape which plays a vital role to recognize the object at primitive level. Keeping this view as the primary motivational focus, we propose a shape descriptive framework using a multi-level tree structured representation called Hierarchical Convex Polygonal Decomposition (HCPD). Such a framework explores different degrees of convexity of an object's contour-segments in the course of its construction. The convex and non-convex segments of an object's contour are discovered at every level of the HCPD-tree generation by repetitive convex-polygonal approximation of contour segments. We have also presented a novel shape-string-encoding scheme for representing the HCPD-tree which allows us to use the popular concept of string-edit distance to compute shape similarity score between two objects. The proposed framework when deployed for similar shape retrieval task demonstrates reasonably good performance in comparison with other popular shape-retrieval algorithms.</i></p> <p>DOI: doi.org/10.15864/ajac.1108 Sourav Saha, Sahibjot Kaur, Jayanta Basak and Priya Ranjan Sinha Mahapatra</p> |

An Efficient Technique for Finding Longest Common Subsequence of DNA Sequences

TamalChakrabarti

Department of Computer Science & Engineering, Institute of Engineering & Management,
Salt Lake Electronics Complex, Kolkata-700 091, INDIA

tamalc@gmail.com

DevadattaSinha

Department of Computer Science & Engineering, Calcutta University,
92, AcharyaPrafulla Chandra Road, Kolkata – 700009, INDIA

devadatta.sinha@gmail.com

Abstract— Molecular biologists rely very heavily on computer science algorithms as research tools. The process of finding the longest common subsequence of two DNA sequences has a wide range of applications in modern bioinformatics. Genetics databases can hold enormous amounts of raw data, for example the human genome consists of approximately three billion DNA base pairs. The processing of this gigantic volume of data necessitates the use of extremely efficient string algorithms. This paper introduces a space and time effective technique for retrieving the longest common subsequence of DNA sequences.

Keywords—

DNA, Genetics, Sequence, Algorithm, Longest Common Subsequence

I. INTRODUCTION

Over the past several years, the study of Bioinformatics has encompassed research areas related to both Computer Science and Biology^[6]. One of the consequences of this trend is the explosion of data that the bio-molecular researchers have to harness and exploit^[8]. For example, an average pharmaceutical company currently uses information from numerous databases, each containing huge amounts of data^[1] which need to be analysed by a variety of complex tools

Strings arise very naturally in Bioinformatics. An organism's full set of genetic material, known as its genome, is divided up into giant linear DNA molecules called chromosomes, each of which serves conceptually as a onedimensional chemical storage device^[16]. The DNA consists of two strands of Adenine (A), Cytosine (C), Thymine (T), and Guanine (G) nucleotides. We can conceptualize a DNA as an enormous linear array, containing a string over the alphabet {A, C, G and T}.

A major theme of genomics is comparing DNA sequences of two (or more) different organisms and trying to find the common parts of these two sequences^{[19], [23]}. If two DNA sequences have a large similar sub-sequence in common, then there is a good chance that they belong to closely related

organisms. Consequently, finding the longest common subsequence^{[17], [18]} of two DNA sequences is a very important area of research in the field of Bio-informatics. Two DNA sequences can have multiple common sub-sequences^[22]. We desire the retrieval of the longest of such common subsequences of the given DNA sequences.

Traditional algorithms for evaluating the longest common subsequence of DNA sequences use the well-known dynamic programming technique. These algorithms run in quadratic time complexity. Further the space required by this technique is also quadratic in nature^{[9], [10]}. But since the DNA sequences are typically very long (in the order of a few billion nucleotides) the time and space requirements of these algorithms tend to be excessively large and often unmanageable. The process of finding the longest common sub-sequence of DNA sequences can be improved by introducing a divide and conquer based method that wraps over dynamic programming, which can be then parallelised^{[24], [25]} to improve the memory consumption and the run-time of the process.

II. DNA LONGEST COMMON SUB-SEQUENCES

The longest common sub-sequence (LCS) of two DNA sequences^[12], S_1 and S_2 , is a measure of —similarityl of the two sequences. We measure the similarity by finding a third sequence S_3 in which the nucleotides (A, C, G and T) appear in the same order as both S_1 and S_2 but not necessarily consecutively. The longer the sequence S_3 , that we can find the more similar S_1 and S_2 are said to be. So our goal is to find the longest possible sub-sequence S_3 of two given DNA sequences S_1 and S_2 .

For example, let $S_1 = \text{GAATCA}$ and $S_2 = \text{ACAGTTCA}$ be any two DNA sequences. Then the longest common subsequence (LCS) of these two sequences is $S_3 = \text{AATCA}$.

III. FORMAL PROBLEM STATEMENT

We formalize the notion of similarity between two DNA sequences as follows. A sub-sequence of a given DNA sequence is the sequence itself without zero or more of its nucleotides. Let:

$\Sigma = \{A, C, G, T\}$ be the DNA alphabet and
 $X = x_0x_1 \dots x_{m-1}$, be a DNA Sequence of length m over Σ .

Another sequence $Z = z_0z_1 \dots z_{k-1}$ of length k over Σ , is called a sub-sequence of X , if and only if there exists a strictly increasing order i_0, i_1, \dots, i_{k-1} of indices of X , such that for all $j = 0 \dots k - 1$ we have $x_{i_j} = z_j$

For example $Z = GCTG$ is a sub-sequence of $X = AGCGTAG$.

Given two sequences X and Y , thesequence Z is said to be a common subsequence of X and Y if and only if Z is a subsequence of both X and Y .

For example, if $X = AAGGGCCTTTAG$ and $Y = AGAGACTTG$, then $Z = AGCT$ is a common sub-sequence of both X and Y .

However Z is not the longest common sub-sequence (LCS) of X and Y , since its length is 4 and another common subsequence of X and Y , $AGGCTTG$, of length 7 exists. $AGGCTTG$ is the LCS of X and Y since common subsequence of X, Y of length 8 can be found.

In the longest common sub-sequence (LCS) problem, we are given two DNA sequences X and Y and we wish to retrieve their maximum length common sub-sequence Z .

IV. RELATED WORK

The longest common sub-sequence (LCS) problem is typically solved by a recursive approach. Let X and Y be two given DNA sequences, and x, y are their rightmost elements respectively. Let X' and Y' be X and Y with their rightmost elements x and y chopped off (i.e. $X' = X - x$ and $Y' = Y - y$). Then we have the following recurrence relation for evaluating LCS of X and Y .

$$\begin{aligned} \text{LCS}(X, Y) &= \text{LCS}(X', Y') + 1 && \text{if } x = y \\ &= \max(\text{LCS}(X', Y), \text{LCS}(X, Y')) && \text{if } x \neq y \end{aligned}$$

The problem with the recursive algorithm above is that it calculates the LCS of the same prefix pair multiple times. Thus the same work is repeated again and again with each recursive call involving an associated stack push-pop operation.

For example finding the LCS of the DNA sequences ACCGGTCGAGTGCGCGG and GTCGTTTCGGAATGCCA incurs well over a few million recursive calls. The algorithm runs in exponential complexity and is unsuitable for extremely large strings, such as DNA.

To avoid this issue, a dynamic programming method^[4] is used to calculate the LCS of two DNA sequences. The idea behind dynamic programming is simple. We have already seen

that LCS of two sequences can be built from the LCS of prefixes of these sequences. That is an optimal solution to the problem can be found from an optimal solution to its subproblems. This property is known as optimal sub-structure.

Thus, rather than re-calculating a function repeatedly for the same input values, we cache the result of each call. So when a function is called for the second time onwards we can just look-up the cache to retrieve the result. This function works in $O(m*n)$ time complexity, when m and n are the lengths of the two sequences respectively. The look-up table stores results of each sub-problem and hence its size is $m*n$. So the space complexity of the program is $O(m*n)$ as well.

In biological applications, such as the DNA longest common sub-sequence problem, one often encounters strings of very large size. In these cases the $O(m*n)$ space requirement can potentially be a more severe problem than a $O(m*n)$ time requirement. For example if we are comparing two DNA sequences of 100000 nucleotides each, then in the modern computing environment performing roughly ten billion additions/comparisons can be much less cumbersome than working with a ten gigabyte array.

Hirschberg proposed an algorithm^[5] to compute the longest-common-subsequence of two DNA sequences in linear space, proportional to the length of inputs. Hirschberg's linear space algorithm combines dynamic programming with the classic divide-and-conquer technique.

V. PROPOSED APPROACH

We start by constructing a table with $(m + 1)$ rows and $(n + 1)$ columns, in which we build up partial results, m and n being the length of the two DNA sequences. We list one of the sequences across the top and the other down the left, as shown in the figure below.

| | | | | | |
|---|--|---|---|---|---|
| | | T | A | T | A |
| | | | | | |
| A | | | | | |
| C | | | | | |
| T | | | | | |
| A | | | | | |

Figure 1: Table for computation of LCS

In this example shown, the table is the initial score matrix for computing the LCS of two DNA sequences $X = TATA$ and $Y = ACTA$.

Let $c_{i,j}$ denote the score of the (i,j) th cell of the matrix, and x_i and y_j be the i th and j th element of X and Y respectively. Then:

$$\begin{aligned} c_{i,j} &= 0 && \text{if } i = 0 \text{ or } j = 0 \\ c_{i,j} &= c_{i-1,j-1} + 1 && \text{if } i, j \neq 0 \text{ and } x_i = y_j \\ c_{i,j} &= \max(c_{i,j-1}, c_{i-1,j}) && \text{if } i, j \neq 0 \text{ and } x_i \neq y_j \end{aligned}$$

This means

that we can arrive at the score of a cell at position (i, j) in one of three ways:

- From the cell above, i.e. from cell (i - 1, j): this corresponds to LCS (X', Y)
- From the cell to the left, i.e. from cell (i, j - 1): this corresponds to LCS(X, Y')
- From the cell to the left-diagonally above, i.e. from cell (i - 1, j - 1): this corresponds to LCS (X', Y') + 1

For example, the first row and first column of the score table correspond to i = 0 and j = 0. Hence the first row and column are filled with zeros. If we consider the cell c(1, 1) the corresponding elements in X and Y are T and A respectively. So, we see which of the cells c(0, 1) and c(1, 0) is greater. In this case both have the same value, which is 0. So we choose 0 from c(1, 0) and put it in c(1, 1). Similarly for cell c(1, 2) the corresponding element in X and Y are the same, i.e. A. So we take the value of the cell c(0, 1), which is 0 and add 1 to it, such that c(1, 2) = c(0, 1) + 1 = 1. Using the above logic, we can fill-up the entire score table as shown in the figure below.

| | | | | | |
|---|---|----|----|----|----|
| | | T | A | T | A |
| | 0 | 0 | 0 | 0 | 0 |
| A | 0 | ←0 | ↖1 | ←1 | ↖1 |
| C | 0 | ←0 | ↑1 | ↑1 | ↑1 |
| T | 0 | ↖1 | ←1 | ↖2 | ←2 |
| A | 0 | ↑1 | ↖2 | ←2 | ↖3 |

Figure 2: Filled-in score table

The score in the bottom-right cell contains the maximum LCS score for X and Y. We trace back from this bottom-right cell and follow the pointers to build up the LCS in reverse. Following rules are used for the trace-back mechanism:

- We start in the lower-right corner cell and then follow the pointer arrows backward
- Every time we follow a pointer to a diagonal cell to the above-left, we prepend the corresponding common character to the LCS
- We continue in this fashion until we finally reach a 0.

Using this process we find that the LCS of X and Y is ATA.

We can improve the space complexity of the above algorithm if we only care about the length of the LCS of two DNA sequences and not the LCS itself. The crucial observation here is that each $c_{i,j}$ entry depends on only three other score table entries: $c_{i-1, j-1}$, $c_{i-1, j}$ and $c_{i, j-1}$. Thus computing the length of LCS needs only two rows of the score table c at a time: the row being computed and the previous row. The following pseudo-code illustrates the logic involved. *for* (i

= 0 to m) $c_{i,0} = 0$ for (j = 1 to n) $c_{0,j} = 0$; for (i = 1 to m) if ($x_i - 1 == y_{j-1}$) $c_{i,j} = c_{i-1,j} + 1$ else $c_{i,j} = \max(c_{i,j-1}, c_{i-1,j})$
for (i = 1 to m) $c_{i,0} = c_{i,1}$

This technique gives us significant space benefit, but it has the issue that only the length of the LCS can be obtained. Since now retain only the last two columns of the score table we will run out of information if we want to trace-back the pointers to build the LCS.

We can also retrieve the LCS while using the space efficient LCS algorithm if we introduce a divide and conquer strategy in our approach. The key idea is to split one of the input sequences (say Y) into two along its mid-point. Say this split, creates two sub-sequences Y_1 and Y_2 , then

$$Y_1 = y_0 \dots y_{n/2-1} \text{ and } Y_2 = y_{n/2} \dots y_{n-1}$$

We then calculate the LCS lengths for the pair (X, Y_1) and (reverse(X), reverse(Y_2)). If we find an index p in X which maximizes the sums of these forward and reverse LCS lengths, then we have a suitable point to split X, i.e. p. Splitting X along p gives rise to two sub-sequences X_1 and X_2 .

$$X_1 = x_0 \dots x_{p-1} \text{ and } X_2 = x_p \dots x_{m-1}$$

Thus our problem now reduces to finding the solution of two sub-problems.

$$\text{LCS}(X_1, Y_1) \text{ and } \text{LCS}(X_2, Y_2)$$

We continue solving the sub-problems recursively until we reach a sub-problem of size one, which corresponds to a one element sequence.

VI. INJECTION OF PARALLELISM BY MULTI-THREADING

Introduction of the divide and conquer strategy opens up the scope for parallelism ^{[2], [3]} in our algorithm. We can now improve the performance of the overall process by solving the independent sub-problems in parallel, which was otherwise difficult to achieve because of the inter-dependency of the dynamic programming sub-problems^[7]. The idea here is to execute the sub-problems together in multiple threads^[15].

A Thread is defined as an independent stream of instructions that can be scheduled to run simultaneously and/or independently with other threads by the operating system. The primary motivation for using threads is to realize potential gains of program performance. Usage of threads enables us to solve sub-problems in parallel^{[11], [13]}. When compared to the cost of creating and managing a process, a thread can be created with much less operating system

overhead. Managing threads requires fewer system resources than managing processes.

By simultaneously processing the sub-problems $LCS(X_1, Y_1)$ and $LCS(X_2, Y_2)$ in two different threads we achieve gain in program throughput. This process is repeated at every step of the recursion, thereby exploiting the independence of the sub-problems. The overhead of thread creation and deletion on the fly can be avoided if we use a pool of threads. This technique reduces the execution time of retrieving the longest common sub-sequence of DNA sequences.

VII. IDENTIFICATION AND OPTIMIZATION OF CRITICAL SECTION

The program can be further optimized if we carefully observe the key set of instructions, which are repeated most often during the execution. A close look at our program will reveal that filling up a cell of the score table is the most commonly used portion in the whole program. We call this the critical section of the code, as this is segment in which most of the computational time is spent.

Say $c_{i,j}$ be the cell in the score table under consideration.

Let us assume:

- l be the score of the cell to the left, i.e. $c_{i-1,j}$
- t be the score of the cell to the top, i.e. $c_{i,j-1}$
- d be the score of the cell diagonally above, i.e. $c_{i-1,j-1}$
- f be the boolean flag that denotes if $x_i = y_j$

Then the score of the cell $c_{i,j}$ can be computed by the critical section, as illustrated by the pseudo-code below.

```

BEGIN Critical_Section
    calculateScore(l, d, t, f) if (f !=
        0) return d + 1 else if
        (l > t) return l
        else return t
END Critical_Section

```

We can replace the high-level instructions of the critical section by their equivalent assembly instructions to enhance the efficiency of the process. The assembly instructions for the critical section have been depicted below:

```

cmp edx, 0
jne c3 cmp
eax, ecx jge
done mov
eax, ecx jmp
done c3: inc
ebx mov eax,
ebx done

```

The optimized assembly instructions further improve the performance of the evaluation of the longest common

subsequence technique, and enable us to compute the LCS of DNA sequences with lesser time and space requirements.

VIII. RESULTS

We compared the time taken by the Hirschberg algorithm, with that of our algorithm. The tests were conducted under the same environment, with all the other factors remaining same. The results of our tests are shown in the figure below:

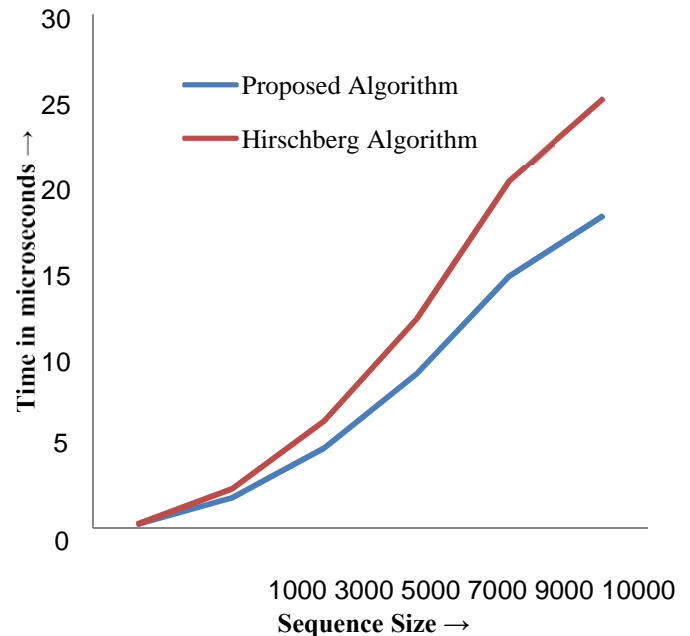


Figure 3: Comparative analysis of Hirschberg vs. the proposed approach

As can be seen, the proposed approach works much faster than the Hirschberg algorithm when the sequence size is appreciably big, i.e. 3000 or more.

IX. CONCLUSIONS

In this paper, we introduced an optimized dynamic programming and divide and conquer based approach, which is then parallelized to reduce the time of the retrieval DNA longest common sub-sequence process. We conducted a comparative analysis of the run time of the traditional Hirschberg Algorithm vs. the proposed approach. The results show that the proposed approach works well, when the sequence sizes are appreciable. For smaller sequences, conventional approach, like the Hirschberg algorithm, can give similar time, but larger the size of the sequence, better is time benefit of using the proposed technique.

At the end it may be mentioned that the subject opens up a wide scope of investigative study with a view to explore further improvement, if any. The authors suggest the following areas of future research:

- Experimenting with real data
- Comparing our approach, with other approaches^{[20], [21]} to exploit their complementary strengths
- Exploiting higher levels of parallelism
- Applying these techniques to similar problems, like the DNA sequence alignment^[14] problem

REFERENCES

- [1] Alexander A. Morgana,b,* , Lynette Hirschmana, Marc Colosimoa, Alexander S. Yeha, Jeff B. Colombe, Gene name identification and normalization using a model organism database, *Journal of Biomedical Informatics* 37 (2004) 396–410
- [2] Babu, K.N. and S. Saxena, 1997. Parallel algorithms for the longest common subsequence problem. *Proceedings of the Fourth International Conference High Performance Computing, (HPC' 1997)*, p: 120-125
- [3] Chen, Y., A. Wan and W. Liu, 2006. A fast parallel algorithm for finding the longest common sequence of multiple biosequences. *BMC Bioinformatics*.
- [4] Chowdhury, R.A., L. Hai-Son and R. Vijaya, 2010. Cache-oblivious dynamic programming for bioinformatics. *IEEE/ACM Transact. Computat. Biol. Bioinformat.*, 7(3).
- [5] D. S. Hirschberg, 1975, A linear space algorithm for computing maximal common subsequences, *Communications of the ACM*, Volume 18 Issue 6
- [6] David W. Mount, *Bioinformatics – Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, 2001
- [7] Dmitry, K., Q. Wang and Y. Shang, 2008. An efficient parallel algorithm for the multiple longest common subsequence (mlcs) problem. *Proceedings of the 37th International Conference on Parallel Processing IEEE Computer Society*. Washington, DC, p: 354-363.
- [8] E.S. Lander, Langridge R., and D.M. Saccocio. Mapping and Interpreting Biological Information. *Communications of the ACM*, 34(11):33 – 39, November 1991.
- [9] Farzaneh Sadat Tabataba, SayyedRasoulMousavi, 2012, A hyperheuristic for the Longest Common Subsequence problem,
- [10] Hunt, J.W. and T.G. Szymanski, 1977. A fast algorithm for computing longest common subsequences. *Comm. ACM*, 20(5): 350-353.
- [11] J.M. Arnold, D.A. Buell, and E.G. Davis. Splash 2. In *ACM Symposium on Parallel Algorithms and Architectures*, pages 316 – 322, June 1992.
- [12] Kang Ning, 2010, Deposition and extension approach to find longest common subsequence for thousands of long sequences, *Computational Biology and Chemistry*, Volume 34, Issue 3, Pages 149-157
- [13] Korkin, D., 2001. A New Dominant Point-Based Parallel Algorithm for Multiple Longest Common Subsequence Problem. Technical Report TR01-148, University of New Brunswick.
- [14] Lin, C. H., Chen, S. J., and Chen, S. M. 2003. A new method for multiple DNA sequence alignment based on genetic algorithms. *Proceedings of the 2003 Joint Conference of AI, Fuzzy System, and Grey System*, Taipei, Taiwan, Republic of China.
- [15] Madej, T., Gibrat, J. E, and Bryant, S. H. (1995). Threading a database of protein cores. *Proteins*, 23:356-369.
- [16] P. Smith-Keary. *Molecular Genetics*. Macmillan Education Ltd, London, 1991.
- [17] Qingguo, W., K. Dmitry and S. Yi, 2011. A fast Multiple Longest Common Subsequence (MLCS) Algorithm. *IEEE Transact. Knowledge Data Eng.*, 23(3). *Res. J. Appl. Sci. Eng. Technol.*, 4(9): 1198-1204, 2012 1204
- [18] S. Guillemot, 2011, Parameterized complexity and approximability of the Longest Compatible Sequence problem, *Discrete Optimization*, Volume 8, Issue 1, February 2011, Pages 50-60
- [19] Sankoff, D., 1972. Matching sequences under deletion/insertion constraints. *Proc. Natl. Acad. Sci. USA*, 69(1):4-6.
- [20] SayyedRasoulMousavi, FarzanehTabataba, 2012, An improved algorithm for the longest common subsequence problem, *Computers & Operations Research*, Volume 39, Issue 3, Pages 512-520
- [21] Sebastian Deorowicz, 2010, Bit-Parallel Algorithm for the Constrained Longest Common Subsequence Problem, *Fundamenta Informaticae*, Volume 99, Number 4 / 2010, p409-433
- [22] ShyongJianShyu and Chun-Yuan Tsai, 2009, Finding the longest common subsequence for multiple biological sequences by ant colony optimization, *Computers & Operations Research*, Volume 36, Issue 1, Pages 73–91
- [23] Temple, F.S. and M.S. Waterman, 1981. Identification of common molecular subsequences. *J. Molecul. Biol.*, 147(1):195-197.
- [24] Xu, X., L. Chen, Y. Pan and P. He, 2005. Fast Parallel Algorithms for the Longest Common Subsequence Problem Using an Optical Bus. *Lecture Notes in Computer Science*, Springer, pp: 338-348.
- [25] Yap, T.K., O. Frieder and R.L. Martino, 1998. Parallel computation in biological sequence analysis. *IEEE Trans. Parallel Distribut. Syst.*, 9(3): 283-294.

A Comparative Study of Different Techniques for Prime Testing in Implementation of RSA

Kumarjit Banerjee

ASE, Tata Consultancy Services,
kumarjit.banerjee@tcs.com

Satyendra Nath Mandal,

Dept. of I.T, Kalyani Govt. Engg
College, Kalyani, Nadia (W.B), India,
satyen_kgec@rediffmail.com³

Sanjoy Kumar Das

University of Kalyani, Nadia (W.B),
India, dassanjoy0810@hotmail.com

Abstract - The RSA cryptosystem, invented by Ron Rivest, Adi Shamir and Len Adleman was first publicized in the August 1977 issue of Scientific American. The security level of this algorithm very much depends on two large prime numbers. The large primes have been taken by BigInteger in Java. An algorithm has been proposed to calculate the exact square root of the given number. Three methods have been used to check whether a given number is prime or not. In trial division approach, a number has to be divided from 2 to the half the square root of the number. The number will be not prime if it gives any factor in trial division. A prime number can be represented by $6n \pm 1$ but all numbers which are of the form $6n \pm 1$ may not be prime. A set of linear equations like $30k+1$, $30k+7$, $30k+11$, $30k+13$, $30k+17$, $30k+19$, $30k+23$ and $30k+29$ also have been used to produce pseudo primes. In this paper, an effort has been made to implement all three methods in implementation of RSA algorithm with large integers. A comparison has been made based on their time complexity and number of pseudo primes. It has been observed that the set of linear equations, have given better results compared to other methods.

Keywords: RSA Algorithm, Trial Division, Pseudo Prime, BigInteger, $6n \pm 1$ approach, 30k approach

I. INTRODUCTION

The requirements of information security within an organization have undergone two major changes in the last few decades [14]. With the introduction of the computer the lead of automated tools for protecting files and other information stored on the computer became evident, especially the case for a shared system [5]. No one can deny the importance of security in data communication and networking [17]. Security [2] in networking is based on cryptography [8], the science and art of transforming messages to make them secure and immune to attack [18]. The RSA algorithm [1] is the most popular and proven asymmetric key cryptographic algorithm. The importance of asymmetric key cryptography is that, the private key does not to be shared on the network [12]. Only the public key is shared. A more formal definition of asymmetric cryptosystem may be given as [10]: A cryptosystem consisting of a set of enciphering transformations $\{E_e\}$ and a set of deciphering transformations $\{D_d\}$ is called a Public-key Cryptosystem or an Asymmetric Cryptosystem if, for each key pair (e, d) , the enciphering key e , called the public key, is made publicly available, while the deciphering key d , called the private key, is kept secret. The cryptosystem must satisfy the property that it is computationally infeasible to compute d from e . The RSA algorithm first requires

two sufficiently large primes to be chosen. For this purpose the primality tests of the numbers has to be computed. In the papers [4], [7] the trial division algorithm has been considered. In this regard, it must be mentioned that trial division can be used both for primality testing and factorization of numbers [13]. The modified trial division has been used along with the divisibility tests. In paper [4], some of the divisibility tests have been demonstrated. In the latter paper, it was shown that, taking numbers as string inputs requires algorithms to handle the strings and manipulate them as integers. It was also shown due the complexity involved in finding the square root of a large number, an upper bound is considered. This reduces the complexity of finding the square root. The time complexity was further reduced by considering only the odd numbers. In the next paper [7], the lower bound of the square root of n has been computed efficiently. The prime numbers are always of the form $6k+1$ and $6k+5$ apart from the number 2 and 3. Thus the original trial division for checking a prime number, the complexity was \sqrt{n} . The complexity was reduced to slightly higher than $\frac{1}{2}\sqrt{n}$ which is further reduced to $\frac{1}{3}\sqrt{n}$. Also the lower bound of the square root involves no unnecessary iteration. It is perfectly reasonable to use trial division as a primality test when n is not too large. Of course, "too large" is a subjective quality; such judgment depends on the speed of the computing equipment and how much time one is willing to allow a computer to run. On a modern workstation, and very roughly speaking, numbers that can be proved prime via trial division in one minute do not exceed 13 decimal digits [16]. However the time is reduced and for an 18 decimal digits number, it takes about 1 hour 5mins [7]. This is a considerable gain. Also the time is further reduced to 50mins (not considering the time for finding the multiplicative order [6]). However, in the paper [7] as mentioned finding the multiplicative order of a number is highly time consuming. As the number of digits increases, the complexity also increases exponentially which is even worse than computing modulo. The aim of this paper is to stick to computing the modulus as it is [4] and reduce the number of iterations of successive modulo computation. This reduction of the number of steps will eventually reduce the time. Thus a set of linear polynomials are considered. It seems that theoretically as well as practically, the time of computation reduces using the above mentioned technique. Another notable progress which is to be

Published Online, 2018 in AST Publishers
(<http://www.astpublishers.com/>) DOI: XX.XXXX/ajac.20XX.XX.XX

mentioned is that, the Java programming language provides the techniques for handling large integers using the java.lang.math.BigInteger API [9]. Earlier mentioned algorithms [4] is applicable and may be implemented in any programming language, however in order to focus on the performance of trial division BigInteger class has been used in this paper. But the Java API does not provide any means to find the square root. So an algorithm has been proposed to implement the square root using the Strings and compute the rest of the operations using the API. This has tremendously reduced the time required for computation.

This paper is divided into the following parts. Article 1 is the introduction. Article 2 describes the RSA algorithm. Article 3 is provides the necessary facts and data needed to compute efficiently. Article 4 describes the algorithm for finding square root of a large number. The Article No. 5 is the implementation. This part is the vital part of the project as it distinguishes the various algorithms used based on the time complexity. Article 6 is the comparison for modified trial division, 6k approach and 30k approach. Article 7 is the results obtained as a part of article 5&6. Finally the Article No 8 is the part for future works based on the conclusions from this paper. A list of references is provided at the end.

II. RSA ALGORITHM

The RSA algorithm involves three steps: key generation, encryption and decryption.

A. Key Generation

RSA involves a public key and a private key [11]. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

Choose two distinct prime numbers p and q . For security purposes, the integers p and q should be chosen uniformly at random and should be of similar bit-length. Prime integers can be efficiently found using a Primality test. Compute $n = p * q$. n is used as the modulus for both the public and private keys. Compute the totient [15]: $\phi(n) = (p-1)*(q-1)$. Choose an integer e such that $1 < e < \phi(n)$, and e and $\phi(n)$ are coprime. e is released as the public key exponent. Choosing e having a short addition chain results in more efficient encryption. Determine d (using modular arithmetic) which satisfies the congruence relation $d * e \equiv 1 \pmod{\phi(n)}$. d is kept as the private key exponent. The public key consists of the modulus n and the public (or encryption) exponent e . The private key consists of the modulus n and the private (or decryption) exponent d which must be kept secret.

B. Encryption

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice. He first turns M into an integer $0 < m < n$ by using an

agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to: $c \equiv m^e \pmod{n}$. This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

C. Decryption

Alice can recover m from c by using her private key exponent d by the following computation: $m \equiv c^d \pmod{n}$. Given m , she can recover the original message M by reversing the padding scheme.

The above decryption procedure works because: $m \equiv (m^e)^d \pmod{n} \equiv m^{ed} \pmod{n}$. Now, since $e * d = 1 + k * \phi(n)$, $m^{ed} \equiv m^{1+k*\phi(n)} \equiv m * (m^{\phi(n)})^k \equiv m \pmod{n}$

The last congruence directly follows from Euler's theorem when m is relatively prime to n . By using the Chinese remainder theorem it can be shown that the equations hold for all m . This shows that the original message is retrieved: $c^d \equiv m \pmod{n}$.

III. FACTS AND DATA

A. Any prime other than 2 and 3 is of the form $6 * p \pm 1$ for some integer p

Proof: From the division algorithm, any integer can be written as any one of the following forms:

$6 * k, 6 * k + 1, 6 * k + 2, 6 * k + 3, 6 * k + 4, 6 * k + 5$.

Now,

1. The numbers which are of the form $6 * k$ are not prime.
2. The numbers which are of the form $6 * k + 2$ are not prime, since $6 * k + 2 = 2 * (3 * k + 1)$.
3. The numbers which are of the form $6 * k + 3$ are not prime, since $6 * k + 3 = 3 * (2 * k + 1)$.
4. The numbers which are of the form $6 * k + 4$ are not prime, since $6 * k + 4 = 2 * (3 * k + 2)$.

So the only choices left are $6 * k + 1$ and $6 * k + 5$.

Again, $6 * k + 5 = 6 * (k - 1) + 1 = 6 * p - 1$ for some integer p ;

Hence, any prime other than 2 and 3 is of the form $6 * p \pm 1$ for some integer p .

(Proved).

B. Every prime number is any of the either forms $30k + 1, 30k + 7, 30k + 11, 30k + 13, 30k + 17, 30k + 19, 30k + 23, 30k + 29$ apart from 2, 3, 5.

Proof: From the division algorithm, any integer can be expressed in any of the forms,

$30k$, $30k+1$, $30k+2=2(15k+1)$, $30k+3=3(10k+1)$,
 $30k+4=2(15k+2)$, $30k+5=5(6k+1)$, $30k+6=6(5k+1)$,
 $30k+7$, $30k+8=2(15k+4)$, $30k+9=3(10k+3)$,
 $30k+10=10(3k+1)$, $30k+11$, $30k+12=6(5k+2)$, $30k+13$,
 $30k+14=2(15k+7)$, $30k+15=15(2k+1)$, $30k+16=2(15k+8)$,
 $30k+17$, $30k+18=6(5k+3)$, $30k+19$,
 $30k+20=10(3k+2)$, $30k+21=3(10k+7)$, $30k+22=2(15k+11)$,
 Published Online, 2018 in AST Publishers
 (http://www.astpublishers.com/) DOI: XX.XXXX/ajac.20XX.XX.XX

$30k+23$, $30k+24=6(5k+4)$, $30k+25=5(6k+5)$,
 $30k+26=2(15k+13)$, $30k+27=3(10k+9)$, $30k+28=2(15k+14)$,
 $30k+29$.

The set of numbers which cannot be expressed as an explicit product of two numbers among the above numbers, are the set of pseudo primes. The set of primes except 2, 3 and 5 is a subset of the pseudo primes. Hence proved.

C. Choosing the value for which set of pseudo primes are generated.

The number 30 is so chosen so that the ratio of the number of elements of the set of pseudo primes to the number is least. Another example of such a number is 12 for which the number of elements of psudo primes is 4. However $4/12 = 1/3 = 2/6$ which is the same as the primes expressed as $6k \pm 1$. Choosing 12 thus gives us no extra advantage. But choosing 30, the ratio is $8/30 = 4/15 < 1/3$. This advantage in turn reduces the time complexity shown in the results (Table 3). Table 1 shows the number of primes below x defined by the function $\pi(x)$. The data for the below table is taken from the internet and is assumed to be correct. Further calculations are made assuming the correctness of the data provided in the below table.

TABLE 1 Taken from <http://primes.utm.edu/howmany.shtml> on 12.05.2012 [19]

| SI No | x | $\pi(x)$ |
|-------|--------------------|-----------------------|
| 1 | 10 | 4 |
| 2 | 100 | 25 |
| 3 | 1000 | 168 |
| 4 | 10000 | 1,229 |
| 5 | 100000 | 9,592 |
| 6 | 1000000 | 78,498 |
| 7 | 10000000 | 664,579 |
| 8 | 100000000 | 5,761,455 |
| 9 | 1000000000 | 50,847,534 |
| 10 | 10000000000 | 455,052,511 |
| 11 | 100000000000 | 4,118,054,813 |
| 12 | 1000000000000 | 37,607,912,018 |
| 13 | 10000000000000 | 346,065,536,839 |
| 14 | 100000000000000 | 3,204,941,750,802 |
| 15 | 1000000000000000 | 29,844,570,422,669 |
| 16 | 10000000000000000 | 279,238,341,033,925 |
| 17 | 100000000000000000 | 2,623,557,157,654,233 |

| | | |
|----|---------------------------|--------------------------------|
| 18 | 1000000000000000000 | 24,739,954,287,740,860 |
| 19 | 10000000000000000000 | 234,057,667,276,344,607 |
| 20 | 100000000000000000000 | 2,220,819,602,560,918,840 |
| 21 | 1000000000000000000000 | 21,127,269,486,018,731,928 |
| 22 | 10000000000000000000000 | 201,467,286,689,315,906,290 |
| 23 | 100000000000000000000000 | 1,925,320,391,606,803,968,923 |
| 24 | 1000000000000000000000000 | 18,435,599,767,349,200,867,866 |

IV. ALGORITHM FOR SQUARE ROOT

1. Take the input number as String
2. Compute the length of the number.
3. If the length is odd goto step 5.
4. If the length is even go to step 6.
5. Compute the square root of the first digit using the available square root method and add $(\text{length}-1)/2$ zeros and next go to step 7.
6. Compute the square root of the first two digits using the available square root method and add $(\text{length}-2)/2$ zeros and next go to step 7.
7. From the second place from left 1 is added and multiplied by itself to check whether it is greater than the given number. Once it is greater the previous number is restored and manipulation is done for the next digit until the units' place arrives.

V. RSA IMPLEMENTATION

The following example demonstrates the RSA algorithm.

Keygeneration

Choose $p=738073442356770323$, $q=867053138428835569$
 such that
 $n=p*q=639948894586411968719277876466018787$, $\phi(n)=(p-1)*(q-1)=$
 $639948894586411967114151295680412896$
 Chose $e = 809$, then $d =$
 $442980693409630039040698053870248729$.

Encryption

Plain text

Modified Pseudoprime Technique in Trial Division for RSA Algorithm Implementation.

Encrypted text

577071194430732909898475103164735624
 129226543293407450964816274468043629
 423203089722537744153404239288585669
 415065772330596047365187423095134135
 601297229861452999384477227132609471
 415065772330596047365187423095134135
 176217843820718634704151973094933395

423203089722537744153404239288585669
 264183799975231368648364388789481394
 293503728936261141184848430630537556
 558983642432338038104835411624110678
 176217843820718634704151973094933395
 342903777217157119488014460258595276
 423203089722537744153404239288585669
 129226543293407450964816274468043629
 245423753596837702288451609473658256

107407424135020706888937120631977672
 415065772330596047365187423095134135
 303552951755819338372645731565618790
 176217843820718634704151973094933395
 264183799975231368648364388789481394
 356808713773152680760603572696977886
 176217843820718634704151973094933395
 325836166038945921612341638423936018

9887311103219652355004785572405134

304920493047471521283534080158340094 415065772330596047365187423095134135

300594192591881741332797243341618452

342903777217157119488014460258595276

176217843820718634704151973094933395

264183799975231368648364388789481394

415065772330596047365187423095134135

304920493047471521283534080158340094

264183799975231368648364388789481394

356808713773152680760603572696977886

107407424135020706888937120631977672

415065772330596047365187423095134135

425229117458901756389587234081524903

531062545253015243506345902871109135

264183799975231368648364388789481394

VI. COMPARISON

A. Number of composites for each Approach

This section details the results obtained by the use of the above mentioned technique [3]. The numbers of composites

are computed based the fact obtained from Table 1. Finally the time required for each are computed and depicted in the table 3. Comparisons are also made showing the efficiency of the above mentioned technique.

TABLE 2. Number of composites in the set of pseudo primes for 6k and 30k approach

| Sl No | 6k Method | | 30k Method | |
|-------|----------------------|---------------------|----------------------|---------------------|
| | No. of pseudo primes | No of Composites | No. of pseudo primes | No of Composites |
| 1 | 5 | 1 | 5 | 1 |
| 2 | 35 | 10 | 29 | 4 |
| 3 | 335 | 167 | 269 | 101 |
| 4 | 3335 | 2106 | 2669 | 1440 |
| 5 | 33335 | 23743 | 26669 | 17077 |
| 6 | 333335 | 254837 | 266669 | 188171 |
| 7 | 3333335 | 2668756 | 2666669 | 2002090 |
| 8 | 33333335 | 27571880 | 26666669 | 20905214 |
| 9 | 333333335 | 282485801 | 266666669 | 215819135 |
| 10 | 3333333335 | 2878280824 | 2666666669 | 2211614158 |
| 11 | 33333333335 | 29215278522 | 26666674669 | 22548619856 |
| 12 | 333333333335 | 295725421317 | 266666666669 | 229058754651 |
| 13 | 3333333333335 | 2987267796496 | 2666666666669 | 2320601129830 |
| 14 | 33333333333335 | 30128391582533 | 26666666666669 | 23461724915867 |
| 15 | 333333333333335 | 303488762910666 | 266666666666669 | 236822096244000 |
| 16 | 3333333333333335 | 3054094992299410 | 2666666666666669 | 2387428325632744 |
| 17 | 33333333333333335 | 30709776175679102 | 26666666666666669 | 24043109509012436 |
| 18 | 333333333333333335 | 308593379045592475 | 266666666666666669 | 241926712378925809 |
| 19 | 3333333333333333335 | 3099275666056988728 | 2666666666666666669 | 2432608999390322062 |

355788718088346271897476223336502171
 415065772330596047365187423095134135
 253585542471502083063853568498282725
 415065772330596047365187423095134135
 558983642432338038104835411624110678
 415065772330596047365187423095134135
 129226543293407450964816274468043629
 304920493047471521283534080158340094
 264183799975231368648364388789481394
 601297229861452999384477227132609471
 129226543293407450964816274468043629
 107407424135020706888937120631977672
 264183799975231368648364388789481394
 326618182158762361802553012280795598
 26160983530421612117563690350886144
 193914648167076844858826180155655583
 264183799975231368648364388789481394
 193914648167076844858826180155655583
 531062545253015243506345902871109135
 374494134193816962365257511998066265
 129226543293407450964816274468043629 107407424135020706888937120631977672 415065772330596047365187423095134135
 520905100285494589272246450259917094
 9887311103219652355004785572405134
 303552951755819338372645731565618790
 264183799975231368648364388789481394
 490861500368270558377320792680824056 303552951755819338372645731565618790 245423753596837702288451609473658256
 531062545253015243506345902871109135
 176217843820718634704151973094933395
 303552951755819338372645731565618790
 176217843820718634704151973094933395
 304920493047471521283534080158340094 520905100285494589272246450259917094 425229117458901756389587234081524903
 520905100285494589272246450259917094
 415065772330596047365187423095134135
 129226543293407450964816274468043629
 304920493047471521283534080158340094 46591653856392057844952957244229066

| | | | | |
|----|--------------------------------|----------------------------------|----------------------------|----------------------------------|
| 20 | 33333333 33333333 35 | 311125137 307724144 95 | 266666666666 6666669 | 24445847064 105747829 |
| 21 | 33333333 33333333 335 | 312206063 847314601 407 | 266666666666 6666669 | 24553939718 0647934741 |
| 22 | 33333333 33333333 3335 | 313186604 664401742 7045 | 266666666666 66666669 | 24651993799 77350760379 |
| 23 | 33333333 33333333 33335 | 314080129 417265293 64412 | 266666666666 666666669 | 24741346275 05986269774 6 |
| 24 | 33333333 33333333 333335 | 314897733 565984132 465469 | 266666666666 6666666669 | 24823106689 93174657988 03 |

Decrypted text is as follows:

Modified Pseudoprime Technique in Trial Division for RSA Algorithm Implementation.

B. Comparative study in primarily checking and encryption/decryption based on time

TABLE 3. Comparison between times for prime check of the previous algorithm and with this deterministic approach (Time very large mostly greater than 1 hour are not mentioned)

| Digits | Prime | First Algorithm [1] | 6k Approach | 30k Approach |
|--------|-----------|---------------------|-------------|--------------|
| 3 | 101 | < 1 sec | <1 sec | <1 sec |
| 3 | 751 | < 1 sec | <1 sec | <1 sec |
| 4 | 1201 | < 1 sec | <1 sec | <1 sec |
| 4 | 9091 | < 1 sec | <1 sec | <1 sec |
| 5 | 10753 | < 1 sec | <1 sec | <1 sec |
| 5 | 76801 | < 1 sec | <1 sec | <1 sec |
| 6 | 160001 | < 1 sec | <1 sec | <1 sec |
| 6 | 980801 | < 1 sec | <1 sec | <1 sec |
| 7 | 1146881 | < 1 sec | <1 sec | <1 sec |
| 7 | 9011201 | < 1 sec | <1 sec | <1 sec |
| 8 | 12600001 | < 1 sec | <1 sec | <1 sec |
| 8 | 99328001 | < 1 sec | <1 sec | <1 sec |
| 9 | 104857601 | < 1 sec | <1 sec | <1 sec |
| 9 | 756100001 | < 1 sec | <1 sec | <1 sec |

| | | | | |
|----|-------------------|-----------|---------|---------|
| 10 | 1027200001 | < 1 sec | <1 sec | <1 sec |
| 10 | 9524994049 | 1 sec | <1 sec | <1 sec |
| 11 | 10256250001 | 1 sec | <1 sec | <1 sec |
| 11 | 97656250001 | 2 secs | <1 sec | <1 sec |
| 12 | 100907200001 | 2 secs | <1 sec | <1 sec |
| 12 | 947147262401 | 3 secs | <1 sec | <1 sec |
| 13 | 1079916250001 | 5 secs | <1 sec | <1 sec |
| 13 | 9982699110401 | 8 secs | <1 sec | <1 sec |
| 14 | 12123750000001 | 10 secs | <1 sec | <1 sec |
| 14 | 87770788000001 | 25 secs | 1 sec | < 1 sec |
| 15 | 101702694862849 | 53 secs | 2 secs | 1 sec |
| 15 | 944377409044481 | 113 secs | 6 secs | 4 secs |
| 16 | 1136591040000001 | 127 secs | 6 secs | 4 secs |
| 16 | 9502720000000001 | 305 secs | 19 secs | 14 secs |
| 17 | 12136000000000001 | 702 secs | 22 secs | 16 secs |
| 17 | 95348273971200001 | 1410 secs | 63 secs | 47 secs |

| | | | | |
|----|---------------------------|-----------|-----------|-----------|
| 18 | 100663296000 000001 | 1630 secs | 65 secs | 48 secs |
| 18 | 908800000000 000001 | 3990 secs | 198 secs | 146 secs |
| 19 | 100000000000 0000003 | ~ | 208 secs | 153 secs |
| 19 | 999999999999 9999961 | ~ | 693 secs | 502 secs |
| 20 | 100000000000 00000051 | ~ | 728 secs | 506 secs |
| 20 | 999999999999 99999989 | ~ | 2861 secs | 2120 secs |
| 21 | 100000000000 000000039 | ~ | 2969 secs | 2139 secs |

pseudo primes. If the set of pseudo primes can intelligibly be reduced that can also be an improvement. Further considerations have to be made as to how to increase the number of digits of the primes but will fulfill the time bound.

TABLE 4. Time to encrypt and decrypt different types of files

| Size of file | Type of file | Time to encrypt in secs | Time to decrypt in secs |
|--------------|--------------|-------------------------|-------------------------|
| 1 KB | txt | 2 | 1 |
| 10 KB | txt | 12 | 6 |
| 14.5 KB | gif | 18 | 8 |
| 44.1 KB | mp3 | 51 | 22 |
| 100 KB | doc | 97 | 42 |
| 100 KB | txt | 123 | 54 |
| 121 KB | pdf | 150 | 65 |
| 427 KB | jpg | 590 | 226 |
| 1 MB | doc | 1054 | 450 |
| 1 MB | txt | 1225 | 595 |
| 47.7 MB | VOB | 6325 | 2787 |

VII. RESULTS

From table 1, a comparison for the number of composites for both 6k and 30k approach has been observed. We can see that the number of composites for both 6k and 30k increased as the given input number is increased, but the growth rate is less for 30k approach than 6k approach. This is the reason why the time taken for modified trial division for 6k approach is more than 30k approach. In table 2, the gradual increase in performance thereby reducing time for prime checking has been demonstrated. In table 3 the time taken for encryption and decryption for RSA implementation has also been recorded.

VII. CONCLUSIONS AND FUTURE WORKS

To check large number whether it is prime or not in personal computer is huge time consuming using trial division algorithm. In trial division approach, a number has to be divided from 2 to the half the square root of the number. The number will be not prime if it gives any factor in trial division. To improve the performance of trial division, a modification has been done by representing the number in form of $6n \pm 1$. In this case among all divisors, only those pseudo primes have been considered which can be represented by $6n \pm 1$. There are many numbers which can be represented in the form $6n \pm 1$ but not prime. A set of linear equations like $30k+1$, $30k+7$, $30k+11$, $30k+13$, $30k+17$, $30k+19$, $30k+23$ and $30k+29$ have been used to produce pseudo primes to minimize the number of composites produced by $6n \pm 1$. It has been observed that the set of linear equations, have given better results compared to other methods. Further works have to be made as to how to only choose the primes instead of choosing

REFERENCES

1. Ron L. Rivest, Adi Shamir, and Len Adleman, "A method for obtaining digital Signatures and public-key cryptosystems", Communications of the ACM 21 (1978), pp 120-126.
2. Boneh and Durfee, "Cryptanalysis of RSA with private key d less than $n^{0.292}$ ", IEEETIT: IEEE Transactions on Information Theory, Volume 46, Issue 4, Jul 2000 pp:1339 – 1349.
3. C. Pomerance, J. L. Selfridge and Wagstaff, Jr., S. S., The pseudoprimes to $25 \cdot 10^9$, Math. Comp., 35:151 (1980) 1003–1026.
4. Mandal N. Satyendra, Banerjee Kumarjit, Maiti Biswajit, Palchaudhury J. , Modified Trail division for Implementation of RSA Algorithm with Large Integers, Int. J. Advanced Networking and Applications Volume: 01, Issue: 04, Pages: 210-216 (2009).
5. M. Wiener, "Cryptanalysis of short rsa secret exponents", IEEE Transactions on Information Theory 36 (1990), pp.553-558.
6. William M. Fauceite Divisibility Rules for 7 and 13 April 2003. <http://160.10.56.251/DivBy7>. Date of access 05.04.2010.
7. Banerjee Kumarjit, Mandal N. Satyendra, Palchaudhury J, Banerjee Abhishek, A Deterministic Approach in Trial Division with Pseudoprimes for RSA Implementation with Large Numbers, IJCSSES International Journal of Computer Sciences and Engineering Systems, Vol.5, No.1, January 2011.
8. Mollin, Richard, An Introduction to Cryptography, Chapman & Hall, CRC, 2000.
9. Guicheng Shen, Bingwu Liu, Xuefeng Zheng, Research on Fast Implementation of RSA with Java, Nanchang, P. R. China, May 22-24, 2009, pp. 186-189.
10. Mollin, Richard, RSA and Public-Key Cryptography, Chapman & Hall, CRC, 2003.
11. David M. Burton, "Elementary Number Theory" (2nd ed), Universal Books Stall, New Delhi, 2004.
12. Douglas R. Stinson, "Cryptography, theory and practice", CRC Press, 1995.
13. Hans Riesel, "Prime numbers and computer methods for factorization" (2nd ed.), Birkhauser Verlag, Basel, Switzerland, Switzerland, 1994.
14. Steven Levy, "Crypto-secrecy and privacy in the new code war", Penguin Books, 2000.
15. William Dunham, "Euler – the master of us all", The Mathematical Association of America, 1999. ISBN: 0883853280.
16. Richard Crandall, Carl Pomerance, Prime Numbers A Computational, Second Edition.
17. Whitfield Diffie and Martin E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory IT-22, no. 6, 1976, pp644-654.
18. Tatsuki Okamoto and Shigenori Uchiyama, "A new public key cryptosystem as secure as factoring", Lecture notes in Computer Science 1403 (1998), 308-318. MR 1 729 059.
19. <http://primes.utm.edu/howmany.shtml> Last date of access 12.05.2012.

AUTHORS' INFORMATION



Satyendra Nath Mandal (23/10/1975) has received his B.Tech & M.Tech in Computer Science & Engineering from university of Calcutta, West Bengal India. He is now working as a lecturer in department of Information Technology at Kalyani Govt. Engg. College , Kalyani , Nadia, West Bengal, India. His field of research areas includes cryptography & network Security, fuzzy logic, Artificial Neural Network, Genetic Algorithm etc. He has about 35 research papers in national and International conferences. His six research papers have been published in International journal.



Kumarjit Banerjee (27/08/1986) has received his B. Tech degree in Computer Science and Engineering form West Bengal University of Technology, West Bengal, India. His field of interest includes Image Processing, Number Theory and Artificial Intelligence. He has published six papers in international Conference. His two research papers have been published in International journal.



Dr. Sanjoy Das is presently working as a Scientific Officer of Department of Engineering And Technological Studies at University of Kalyani, Kalyani, Nadia, West Bengal India-741235.

Coordinate based Routing Protocol for Mobile Networks: A Fuzzy Logic Approach

Paulami Dey

Department of Computer Science & Engineering
National Institute of Technology, Durgapur
Burdwan,
paulamidey77@gmail.com

India

Department of Computer Science & Engineering
National Institute of Technology, Durgapur
Burdwan, India parag.nitdgp@gmail.com

Abstract---An implementation of the co-ordinate based routing protocol for mobile networks is proposed in this paper with the Fuzzy logic concept. The rules for mapping between cell number and corresponding co-ordinates are defined. A flexible sense of membership function of elements supported by Fuzzy logic is used here. All possible routing paths can be enumerated in a simple way. The proposed method is one of the simpler than other techniques reported so far.

Keywords - Fuzzy logic; membership function; routing; mobile network;

I. INTRODUCTION

In the recent time, fast growth of cellular telephony is combined with a need for improved and efficient allocation strategies. A useful call selection procedure is also required at the same time. During congestion in the network, one of such strategies is used to give permission to limit number of users for using the network. This strategy is named as Call Admission Control (CAC). Remaining users are not allotted at any slot during this period [2]. Consequently, Quality of Service (QoS) can be established only for the admitted users. Hence, it is essential to consider two near-contradictory requirements – allocating resources as well as ensuring Quality of Service (QoS) when all users are trying to make a request at the same time.

It is known that in mobile cellular networks nodes communicate with each other using multi-hop links. This structure is stationary because there are base stations in every cell. Each node in the network has call forwarding capability to other nodes. Till date, various routing strategies have been designed to address the problem of finding routing path coupled with efficient congestion control technique. A new Dynamic pricing scheme called as Priority based Tree Generation for mobile networks (PGTM) has been proposed in

Parag Kumar Guha Thakurta

[1]. In [1], an effective call scheduling procedure has been proposed. The unique path sequence for each call requesting cell was also determined.

A new co-ordinate based dynamic routing protocol (CSTR) for mobile networks has been proposed in [3]. The depiction of the mapping scheme between the cellular structure described in [1] and the Cartesian co-ordinate system is given in [3], with the mobile terminal (MT) placed at the origin acting as a switching centre.

In this paper, a new coordinate based routing protocol for mobile networks is proposed. Here, the routing protocol is structured with the help of fuzzy logic concept. The membership function (μ) is defined in numeric manner. An inference rule is defined and subsequently, applied on the previous membership function to generate the possible routing paths for specific source and destination. Hence, this protocol would be able to determine the least cost routing path among the alternatives. The performance analysis for the proposed model has been described.

The rest of the paper is organized as follows. A brief description of mapping rules for the coordinate based routing protocol [3] is described for completeness of the work in section II. The proposed model based on fuzzy logic implementation is described in section III. The performance analysis and experimental results are shown in section IV and section V respectively. In section VI, we conclude with the advantages of the proposed model as well as the future scope of the work.

II. MAPPING RULES OF CSTR

[3]

In this model, Mobile Terminal (MT) is denoted by (0,0) and each other cell is having a coordinate of the form (x,y). The cells covered by the radius r (within transmission covering range [4]) of MT are mapped as (x,y+1), (x+1,y+1) and (x+1,y) such that $x+1 \leq r$ and $y+1 \leq r$. For example, in Fig. 1(a), cell numbers C_{13} , C_{12} and C_{11} of $r = 1$ in [1] are mapped into (1,0), (1,1) and (0,1) respectively. Therefore, cellular structure of mobile networks detected in [1] could be mapped into a coordinate based system as shown in Fig. 1(b)

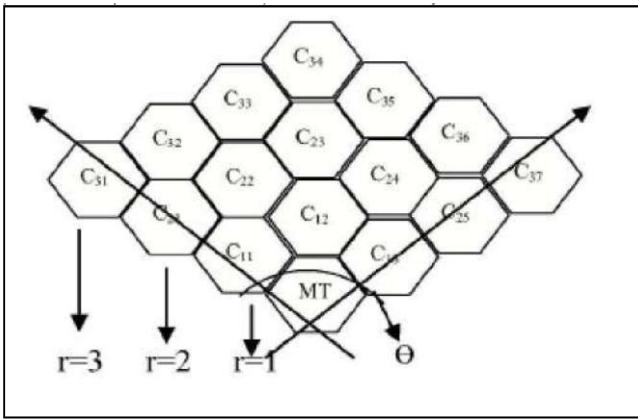


Fig. 1(a): Cellular structure for Mobile Networks for Mobile Networks for $r=3$

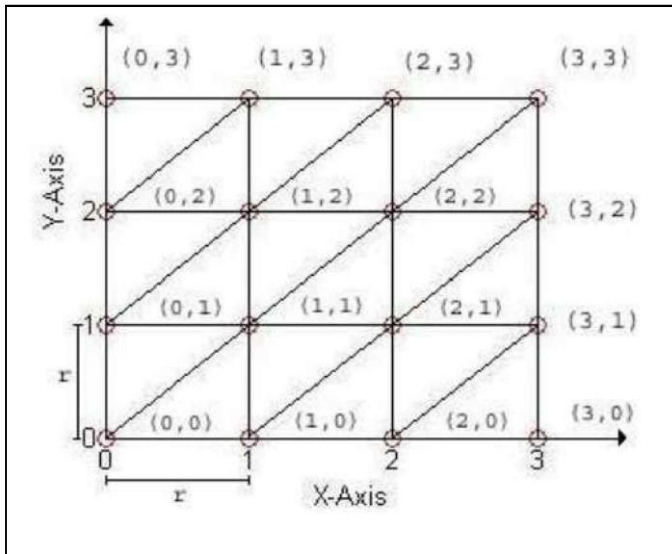


Fig. 1(b): Coordinate based representation of Fig. 1(a) (in [3])

III. PROPOSED MODEL

The proposed model in this paper is based on the fuzzy logic. This concept is used to provide the routing path for a call between different Base Stations (BSs) in various cells. The advantages of this work are listed in the following:-

- (i) All possible routing paths can be enumerated with this approach.
- (ii) The approach is supported with concrete mathematics.

Generally, it is well known that a simple fuzzy system [5] consists of three steps shown in the following Fig. 2.

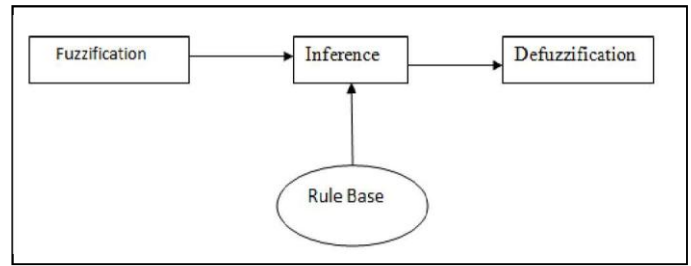


Fig. 2: Structure of fuzzy logic system

A. Fuzzification

The membership function (μ) is computed between the source node and the destination node in the network. For any fuzzy set A , Membership function μ on A is any function from A to the real unit interval $[0,1]$. Here, the fuzzy set for the proposed model is considered as the set of co-ordinates representing specific cells in the cellular structure of the network. For example, the source is represented as (x,y) and the coordinate representing another cell except the source is $(x1,y1)$, then the membership function between (x,y) and $(x1,y1)$ is represented as,

$$\mu = 1 / \{(x-x1)^2 + (y-y1)^2\} \quad (1)$$

Although each cell in the cellular structure can communicate with the cells in every direction of the network, but it is proposed that the system would consider call forwarding capability to only three cells of upper radius. The reason behind this is that – those excluded cells lie either on the lower radius or on the same radius. Hence, rejected calls for those excluded cells are being serviced by other cells in their lower radius. It is evident that this exclusion, in no way, would affect the efficiency of call routing. Moreover it reduces the redundant entries in the set of routing paths.

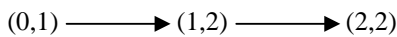
B. Inference

Fuzzy Rule is applied on the membership function μ to provide the routing paths. As the value of μ lies in the interval $[0,1]$, so the mean value 0.5 is kept as the deciding value. The higher than this value would indicate 'High signal' for routing as well as lower than this value would indicate 'No signal' for routing. The value equal to the mean value would indicate 'Normal signal' for routing

So here the rule for the proposed model is defined in the following.

Rule: IF μ is equal to or greater than the value 0.5 THEN the call is forwarded from the source cell to the corresponding cell or node.

For example, direct routing from (0,1) to (2,2) in Fig.1(b) is not possible because the corresponding value of the membership function $\mu = 0.2 < 0.5$. It indicates 'No signal' for routing from (0,1) to (2,2). So routing from (0,1) to (2,2) is possible via another node, like (1,2) and it can be represented as,



C. Defuzzification

It is the reverse process of Fuzzification. It means that conversion process of a fuzzy set into a crisp value. The requirement of defuzzification is beyond scope of this work.

The procedure for finding the routing paths is now described by the following algorithm. **Algorithm:**

Start with MT (x,y), where x=0, y=0
 // MT works as the switching centre
 RouteTo(x,y) begin while(x|y<=r)
 // r is the radius (within transmission covering range [4]) of
 //(MT)

$$\mu = 1 / \{ (x-x_1)^2 + (y-y_1)^2 \} \text{ where } x_1=x, y_1 = y+1 ;$$

```
if( $\mu \geq 0.5$ )
  return  $\mu$ ;
  RouteTo (x1,y1);
```

```
else
  return 0;
endif
```

$$\mu = 1 / \{ (x-x_1)^2 + (y-y_1)^2 \} \text{ where } x_1=x+1, y_1 = y+1 ;$$

```
if( $\mu \geq 0.5$ )
  return  $\mu$ ;
  RouteTo (x1,y1);
```

```
else
  return 0;
endif
```

$$\mu = 1 / \{ (x-x_1)^2 + (y-y_1)^2 \} \text{ where } x_1=x+1, y_1 = y ;$$

```
if( $\mu \geq 0.5$ ) return
   $\mu$ ;
  RouteTo (x1,y1);
else return 0;
endif
endwhile
end
```

IV. PERFORMANCE ANALYSIS

The proposed approach is analyzed with r=2 for simplicity, however it is applicable for any value of r within the transmission range [4] of MT. here, the model verifies with taking MT as (0,0) and subsequently it shows the various routing paths in the Fig. 3(a), Fig. 3(b), Fig. 3(c) and Fig. 3(d) respectively.

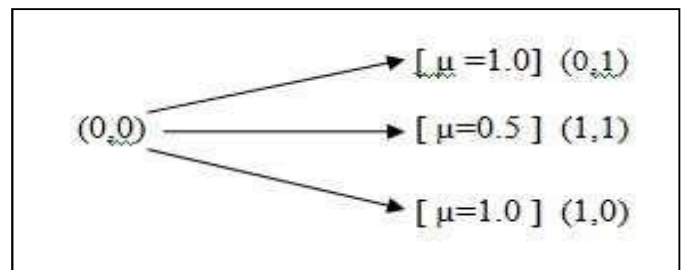


Fig. 3 (a): Routing directions for r=2 from (0,0)

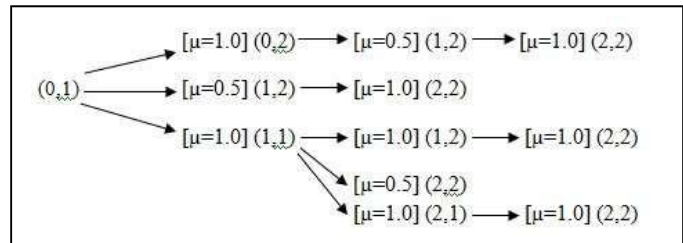


Fig. 3 (b): Routing directions from (0,1)

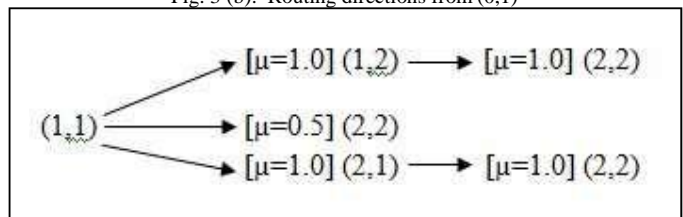


Fig. 3 (c): Routing directions from (1,1)

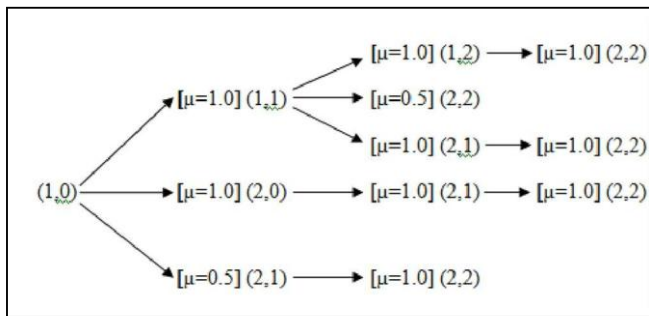


Fig. 3 (d): Routing directions from (1,0)

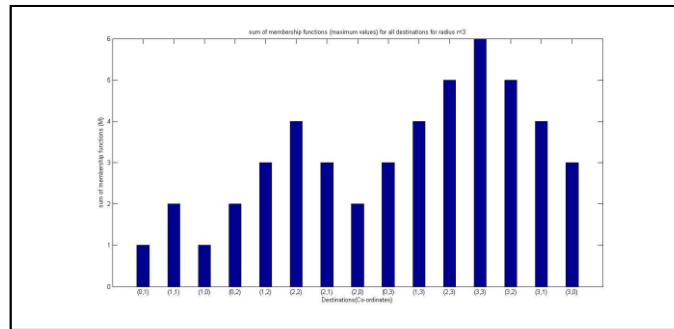


Fig. 4(a): Membership function analysis (for maximum value of M) for all destinations

Here the possible routing paths for a specific destination node (1,2) could represent in the following table 1. In this table, the sum of membership functions value ($M = \sum \mu$) is also computed accordingly. Now, there is a provision to provide the routing path with minimum value of M.

TABLE I. LIST OF POSSIBLE ROUTING PATHS FOR DESTINATION (1,2) WITH THE VALUE OF (M) FOR A PARTICULAR PATH

| Destination | Possible Routing Paths | M ($\sum \mu$) |
|-------------|---|------------------|
| (1,2) | $(0,0) \square (0,1) \square (0,2) \square (1,2)$ | 3 |
| | $(0,0) \square (0,1) \square (1,2)$ | 1.5 |
| | $(0,0) \square (0,1) \square (1,1) \square (1,2)$ | 3 |
| | $(0,0) \square (1,1) \square (1,2)$ | 1.5 |
| | $(0,0) \square (1,0) \square (1,1) \square (1,2)$ | 3 |

Now, the significance of performance analysis is reflected with the help of standard statistical approach. Here, the mean of M for different paths using frequency of occurrence is computed. For example, the mean of M values for the node (1,2) is 2.4 – the corresponding M values are 3 and 1.5 and the respective frequency values are 3 and 2.

V. EXPERIMENTAL RESULTS

This work is analyzed with MATLAB version 7.6.0.324 (R2008a). Here this results show the variation of different values of M which in turn indicates the variation of different paths lengths for all destinations up to radius $r=3$.

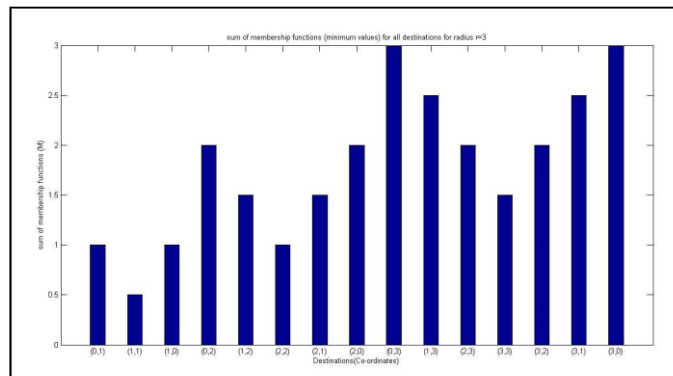


Fig. 4(b): Membership function analysis (for minimum value of M) for all destinations

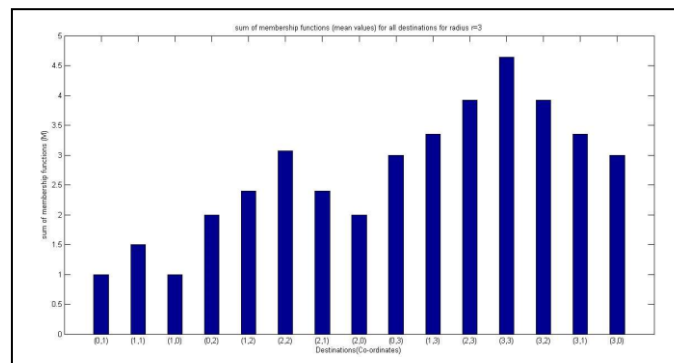


Fig. 4(c): Membership function analysis (for mean value of M) for all destinations

VI. CONCLUSIONS

The procedure for determining the routing paths using fuzzy logic is described in this work. It supports a mathematical basis support; thereby other efficient mathematical logics can be used in future to make this work more reliable. This working procedure increases the flexibility of dynamic call routing. Further study on extending this work to construct the routing table is in progress.

ACKNOWLEDGMENT

The authors would like to thank the Faculty of the department of Computer Science & Engineering of National Institute of Technology, Durgapur for their generous support.

REFERENCES

- [1] P.K.Guha Thakurta and Subhansu Bandyopadhyay, "A New Dynamic Pricing Scheme with Priority based Tree Generation and Scheduling for Mobile Networks", IEEE Advanced Computing Conference, March 2009 (available in IEEE Explore).
- [2] Xinbing Wang, Do Young Eun and Wenye Wang, "Dynamic TCP-Aware Call Admission Control Scheme For Generic Next generation PacketSwitched Wireless .
- [3] P.K.Guha Thakurta, Rajarshi Poddar and Subhansu Bandyopadhyay, "A New Approach on Co-ordinate based Routing Protocol for Mobile Networks", IEEE Advanced Computing Conference, February 2010 .
- [4] Wen-Hwa Liao, Jang-Ping Sheu and Yu-Chee Tseng, " GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks", Journal on Telecommunication Systems, Springer Netherlands, vol.18, No. 1-3, September 2001.
- [5] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, 2nd ed. New York: Springer-Verlag, 1996.

A Novel and Flexible Criterion to Improve Data Transmission in Clustering Protocols in WSNs

Mohsen Ataei

Departement of Computer Engineering, Islamic Azad University, Qazvin Branch (QIAU), Qazvin, Iran
m.ataei@qiau.ac.ir

Esmail Zeinali Kh.

Departement of Computer Engineering, Islamic Azad University, Qazvin Branch (QIAU), Qazvin, Iran
zeinali@qiau.ac.ir

Abstract— In this paper a new criterion called Energy-Cost Function is presented according to which in each round the energy cost for any individual node is calculated. When transmitting their data, the nodes make decisions based on this very cost function. In case the nodes decides that it will cost a lower amount of energy transmitting the data to the sink by itself rather than by the cluster-head to the sink, then the node transmits the data directly. In this way the cluster overload is reduced and both the network lifetime and instability period is boosted. Based on the conditions of the problem, cost function parameters are variable and since all the decisions are made locally, network scalability potential is retained. Simulation results show that the network lifetime or its instability period based on the cost function employed will be improved up to 40% in relation to the LEACH protocol.

Keywords - Wireless Sensor Networks (WSNs); Energy Cost Function; Lifetime; Instability Period;

I. INTRODUCTION

Wireless sensor networks (WSNs) are an especial type of computer-based networks which have had significant applications due to their great potential. Flexibility, self-organization, low-cost and rapid deployment are some of the key and ideal features of WSNs in many applications like data gathering, military areas, environment monitoring, intelligent surveillance, traffic management, medical operations and so on [1,2]. Such networks consist of a large number of tiny, high potential, low energy consumption and low cost sensor nodes which enables them to sense special attributes like damp, temperature, pressure etc. from the environment and send them to the Sink. So, on the whole, these networks are considered in two ways: sensing especial parameters from the environment and, communicating so as to pass their received packets to the Sink.

Sensor networks have many applications some of which include intelligent surveillance on expressways and areas which are difficult to reach, environmental monitoring and target tracking. These applications entail sensor networks to be deployed as wireless [3,4].

Typically these networks have nodes which are fixed or have a limited mobility and a Sink to which any node transmits its data either directly (single-hop) or indirectly (multi-hop). In a direct transmission, any sensor transmits its data to the sink directly hence consuming lots of energy for any transmission

due to long distance between the sensors and the Sink. On the contrary, those designs which shorten the communication distances could extend the duration of network lifetime. As a result, in these networks multi-hop communications are more fruitful and cost effective than that of single-hops.

The use of single-path routing is highly susceptible to security attacks that target to compromise the availability, reliability and resilience of the network [5].

The use of multipath routing can diminish the effect of security attacks that target the availability, reliability and resilience of the network [5].

This point is worth noting, however, that even in multi-hop communications the most energy of the nodes is dissipated as any sensor communicates with its neighbours which would in turn lead to faster energy depletion in sensor nodes. To solve this problem, we can put a limit on communications yet optimize it deploying clustering.

An example of this application is found in hierarchical routing in which nodes are divided into logical clusters. In any cluster one node is the cluster head and the other nodes are considered as ordinary node yet a member of the cluster.

The members of the cluster gather, from the environment, the expected information based on their application domain and then transmit the data to the cluster head. The cluster head after gathering this data transmits it to the Sink [6].

Clustering in sensor networks as an effective method for organizing the nodes of the network, enjoys a good many advantages some of which are as following: scalability, local routing which in turn leads to the decrease of the size of the forwarding table for each node, data aggregation in cluster head in order to minimize the energy dissipation of the nodes and finally reducing the overload of maintaining the network topology since it is localized.

Among the most important clustering algorithms is LEACH which endeavours to boost the network lifetime in a number of ways of which are evenly distributing the energy consumption in all of the nodes of the network and also decreasing the energy consumption in the nodes using the data aggregation strategy, thus reducing the number of control messages. This algorithm is one of the most considerable hierarchical policies for routing in WSN [7]. The LEACH protocol has two phases which are repeated periodically.

Each repetition begins with the set-up phase, where clusters are constructed. In the next phase, namely the steady-state phase, the nodes of each cluster in turn transmit the environment data which have been recorded by their sensors to the cluster head and here in the cluster head node, after the data aggregation, transmission to the Sink or the destination node is achieved [8].

In the set-up phase the first step is to determine the cluster heads. Based on the probabilistic model each node independently decides to be a cluster head. The main thing in the determination of the cluster heads is that no node is selected as cluster head more than the others; otherwise it loses its entire energy. To accomplish this goal, LEACH endeavours to utilize a mechanism for the determination of the cluster head in which any node in turn plays its role as a cluster head. After the cluster heads are determined, in the next step of the set-up phase, the cluster heads must advertise their being cluster head to the ordinary nodes. After the nodes take their turns the steady-state phase begins. The steady-state phase is divided into a number of frames in terms of time. In each frame a fixed amount of time is allocated to any of the nodes of the cluster during which the node transmits the data to the cluster head node. So the length of time for every frame and also the number of frames in every repetition depends on the number of the nodes of any cluster. At the end of each frame, the cluster head node aggregates the gathered data from the nodes and transmits it to the Sink. Accordingly, the number of transmissions by cluster heads to the Sink in any repetition depends on the time frame and the number of the nodes of any cluster and it differs in different clusters.

Non-cluster nodes can turn off their transmitter-receivers when no time is allocated to them thus save their energy. But cluster head nodes must keep their transmitter-receivers on across a repetition [8].

Different routing protocols suffer diverse constraints in design, but energy saving has always been the common ground and the most important aim when designing routing protocols [9]. As the residual energy is of great importance, when selecting a node as cluster head, the residual energy is taken into consideration in addition to the random probabilistic function utilized in LEACH [10]. This fact has an effective role in the better selection of cluster head hence boosts the network lifetime.

In this paper a function inspired by unit 5 of the reference [11] named Energy Cost Function is presented which acts based on the distance from the node to the Sink and also the residual energy of the node. Making the decision about whether to transmit the data directly to the Sink or transmit it to the cluster head is based on this cost function.

II. RADIO MODEL

The energy consumption of each sensor node consists of three components : sensing energy, communication energy and data processing energy. Sensing and data processing require much less energy than communication, so we consider only communication energy consumption [12].

The radio model utilized in this paper is the same as “Fig. 1”.

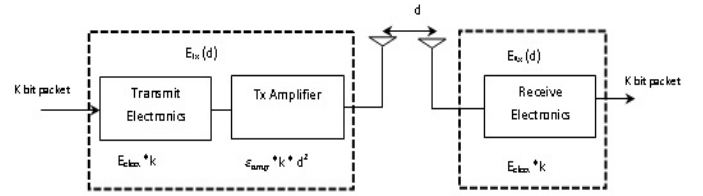


Figure 1. Models used for energy in the radio components, in accordance with LEACH [8]

Based on the radio model of energy consumption presented in “Fig. 1”, the energy consumption of the transmitter who sends a k-bit message can be calculated by the following equations:

$$E_{Tx}(k, d) = E_{Tx-elect}(k) + E_{Tx-amp}(k, d) = \begin{cases} kE_{elect} + k\epsilon_{fs}d^2, & d < d_0 \\ kE_{elect} + k\epsilon_{mp}d^4, & d \geq d_0 \end{cases} \quad (1)$$

E_{elect} is the energy consumption of the wireless transmitter-receiver circuitry for each bit. ϵ_{fs} and ϵ_{mp} depend on the amplifier model of transmitter circuitry while d stands for the distance between the transmitter and the receiver. Thus if we assume $d=d_0$ then we have $d_0 = \sqrt[4]{\epsilon_{fs}/\epsilon_{mp}}$ and to receive a k-bit message, the radio expends:

$$E_{Rx} = k \cdot E_{elect} \quad (2)$$

III. PROPOSED CLUSTERING PROTOCOL

In this section we introduce a criterion inspired by unit 5 of the reference [11] and reference [12], named Energy Cost Function which is the basis for designing the proposed protocol. This criterion is flexible and also adaptable in different situations with different clustering protocols. Simulation results show that employing this criterion when making decisions for transmitting the data by nodes, increases the network lifetime and the instability period¹. Since it is up to the node itself to decide based on this criterion whether or not to transmit the data without the Sink being involved, the network scalability potential is maintained.

A. Energy Cost Function

There are two parameters in WSN which have a considerable effect on the network lifetime. They are the distance from the Sink and the node’s residual energy. Obviously, the more the distance from a node to the Sink is, the more energy will be consumed for data transmission.

Considering the importance of these two parameters, we define a function called Energy Cost Function (equation 3) based on which the nodes transmit their data.

¹ Dead time between the first node to last node dies

$$EC_i = \frac{d_{i \text{ to BS}}}{e_i} \quad (3)$$

In the above equation, EC_i is the i node cost, $d_{i \text{ to BS}}$ is the distance from i node to the base station (Sink) and e_i is the residual energy of the node i .

According to this equation it is clear that the less the distance to the Sink and the more the residual energy of the node is, the less energy will be spent by the node to transmit its data. Parameters like cost function, distance (d) and energy are variable based on the conditions of the problem. In this paper the distances to the base station are considered as fixed and the experiments are carried out in three different situations ($EC = \frac{d}{e}$, $EC = \frac{d}{e^2}$, $EC = \frac{d}{e^4}$).

B. Proposed Clustering Protocol Based on the Energy Cost Function

Like the LEACH protocol our proposed protocol is made up of two phases; the set-up phase and the steady-state phase. Here, however, there is a subtle difference. As a matter of fact in the set-up phase in addition to the determination of the cluster heads based on a random probabilistic function (equation 4 [8]), the energy cost for each node is also calculated according to the calculation 1. Then the node recognizes its immediate cluster head; but for the membership in that cluster compares its cost function to that of the cluster. In case the node finds that the EC by the cluster head is less than that of itself, tries to become a member of that cluster and in the next phase, that is, the steady-state phase, transmits its data within the time frame which the cluster head will allocate to it based on the TDMA model. Now the cluster head gathers and aggregates the data and transmit them to the Sink. Conversely, if the node realizes that its EC will be less than the EC of the cluster head, refuses to become a member of the cluster head and transmits the data directly to the Sink.

This technique achieves a reduction in the operations such as the gathering and the aggregation of data in the cluster head hence reducing the overhead of the cluster head. Needless to say that this is not the case all the time, because deciding whether to transmit to the cluster head or to directly transmit to the Sink depends on the cost function.

$$T(n) = \begin{cases} \frac{p}{1-p \times (r \bmod \frac{1}{p})}, & \text{if } n \in G \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Applying local decision making with a control tool called the Cost Function, our proposed protocol on the one hand reduces the overhead of the cluster head, on the other hand it doesn't allow the nodes to save extra energy thus better energy balance is achieved and the network lifetime expands. It is clear that at the beginning the nodes near to the Sink do not become a member of the cluster head as far as possible; while the nodes in distance attempt to become a member. This is because there is a direct relationship between distance and the energy consumption of the nodes.

Our proposed protocol enjoys a good flexibility. Based on the type of the application and its importance we can increase and decrease parameters of the cost function until the desired result is reached.

For instance in case the network lifetime is of greater importance to us, we multiply the residual energy in the denominator of the fraction. In the next section the results of the experiments with different amounts of the residual energy will be displayed.

It needs to be said that in most proposed protocols, the Sink is involved in the selection of the cluster head; hence local deciding to become a cluster head is ignored thus the network scalability potential is also lost. But in our proposed protocol, this potential is maintained as making decisions are fully localized.

IV. SIMULATION RESULTS

To simulate the proposed protocol, we carried out two experiments with different nodes and with different areas. Each experiment was considered using three different cost functions ($EC = \frac{d}{e}$, $EC = \frac{d}{e^2}$, $EC = \frac{d}{e^4}$).

The results are obtained by averaging the performance of the program 10 times. In both experiments the Sink is in the centre of the sensor network. We ran the simulation utilizing the *Matlab* software.

A. The First Experiment

In this experiment, 100 nodes are uniform randomly distributed in an area of 100*100 meters (Fig.2 displays this area). The initial energy of the nodes is assumed to be 0.5J and the length of the packet 4000 bites. "Fig. 3" and "Fig. 4" illustrate the lifetime and the time frame of the instability period for this experiment.

Table 2 also displays the results obtained on average performing the program 10 times.

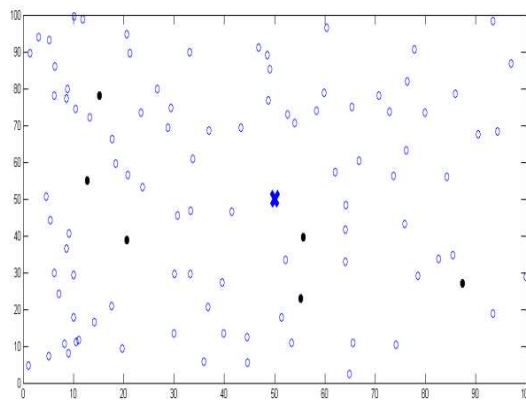


Figure 2. View from the simulation environment

As it is displayed in “Fig. 3” and ”Fig. 4” and the table of results 1, our proposed method with all three EC functions boosts the network lifetime and in the best state (where we apply the function $EC = d/e^4$) this increase reaches 40%.

As it could be understood from the results of the table, there is clearly a trade-off between the network lifetime and the instability of it. Considering the conditions of the problem we can select one of the EC functions and make use of it. For instance in case both the network lifetime and the instability period matters to us, we employ the function $EC = d/e$.

TABLE I. AVERAGE RESULTS OF 10 RUNS WITH DIFFERENT FUNCTIONS FOR THE FIRST EXPERIMENT

| Comparison criterion | LEACH | $EC = \frac{d}{e}$ | $EC = \frac{d}{e^2}$ | $EC = \frac{d}{e^4}$ |
|----------------------|-------|--------------------|----------------------|----------------------|
| Lifetime | 803 | 956 | 1011 | 1112 |
| Last node dies | 1160 | 1367 | 1284 | 1249 |
| Instability period | 357 | 411 | 273 | 137 |

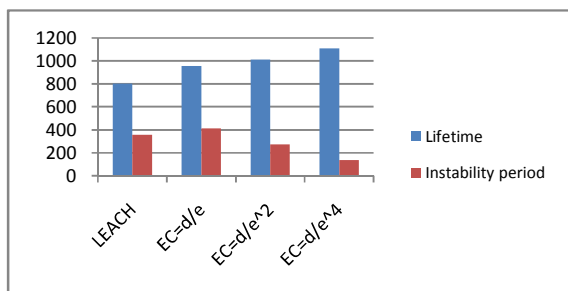


Figure 3. Table 1 compares the results

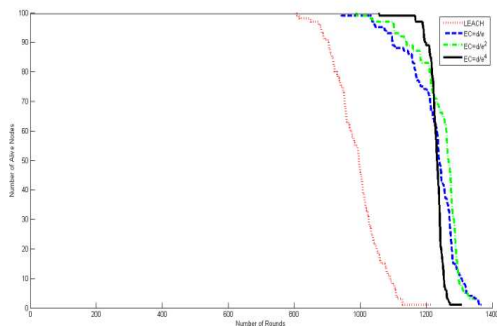


Figure 4. Number of alive nodes per round

B. The Second Experiment

In this experiment 200 nodes are randomly distributed in an area of 200*200 meters. The Sink is placed in the centre of the network (100,100). The initial energy of the nodes is assumed to be 0.5J and the length of the packet 4000 bites. “Fig. 5” and “Fig. 6” depict the results of the experiment. Table 2 also

shows the results obtained on average after the program was performed 10 times.

As it is clearly shown in the figures and the tables, even in this state utilizing the EC function in transmitting the data to the cluster head or directly to the Sink improves the network lifetime.

TABLE II. AVERAGE RESULTS OF 10 RUNS WITH DIFFERENT FUNCTIONS FOR THE SECOND EXPERIMENT

| Comparison criterion | LEACH | $EC = \frac{d}{e}$ | $EC = \frac{d}{e^2}$ | $EC = \frac{d}{e^4}$ |
|----------------------|-------|--------------------|----------------------|----------------------|
| Lifetime | 668 | 670 | 700 | 730 |
| Last node dies | 925 | 1054 | 1051 | 1030 |
| Instability period | 257 | 384 | 351 | 300 |

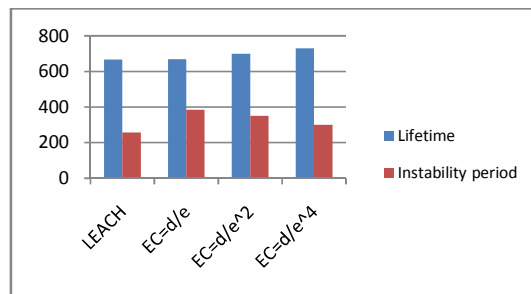


Figure 5. Table 2 compares the results

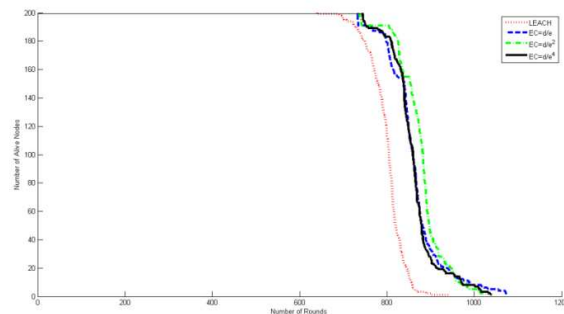


Figure 6. Number of alive nodes per round

V. CONCLUSION

In this paper a novel and flexible criterion called Energy-Cost function for clustering protocols in WSN was presented based on which the nodes can decide on the way they transmit their data to the cluster head or directly to the Sink node. Parameters of this function are variable based on the conditions of the problem. Since all decisions on the selection of the cluster head and the data transmission is made locally in this proposed protocol, the network scalability is maintained. The results show that utilizing this function improves the WSN lifetime or its instability period based on the function used, up to 40% in relation to LEACH protocol. The energy-cost function could be of use to other clustering protocols.

REFERENCES

- [1] H. Chao, Y. Q. Chen, and W. Ren, "A Study of Grouping Effect On Mobile Actuator Sensor Networks for Distributed Feedback Control of Diffusion Process Using Central Voronoi Tessellations", IEEE International Conference of Mechatronics and Automation, pp. 769-774, 2006.
- [2] Y. U. Ming, M. Aniket, and S. U. Wei, "An environment monitoring system architecture based on sensor networks", International Journal of Intelligent Control and Systems, pp. 201-209, 2005.
- [3] J. Yick, B. Mukherjee and D. Ghosal, "Wireless Sensor Network Survey", Journal of Elsevier on Computer Networks, Vol. 52, 2008.
- [4] F. Akyildiz, W. Su, Y. Sankarasubramanian and E. Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine, Vol. 40, pp. 102-114, 2002.
- [5] E. Stavrou, A. Pitsillides, "A survey on secure multipath routing protocols in WSNs", Elsevier Computer Networks, Vol. 54, pp. 2215-2238, 2010.
- [6] A. A. Abbasi, M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks", Elsevier Computer Communication, Vol. 30, 2007.
- [7] M. Younis, M. Youssef, and K. Arisha, "Energy-Awaremanagement in cluster-based sensor Networks", The International Journal on Computer Networks, Vol. 43, pp. 649-668, 2003.
- [8] W. R. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks", IEEE Transactions on Wireless Communications, Vol. 1, pp. 660-670, 2002.
- [9] J. N. Al-Karaki, A. E. Kamal, "Routing techniques in wireless sensor networks: A Survey", IEEE Wireless Communications, Vol. 11, pp. 6-28, 2004.
- [10] Y. M. Fan, J. J. Yu, "The communication protocol for wireless sensor network about LEACH", Proceedings of IEEE CISW'07, 2007.
- [11] Chang-Soo Ok, "Distributed Energy-Balanced Routing in Wireless Sensor Networks", A Dissertation in Industrial Engineering, Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, 2008.
- [12] J. Haifeng, Q. Jiansheng, S. Yanjing and Zh. Guoyong, "Energy optimal routing for long chain-type wireless sensor networks in underground mines", Elsevier Mining Science and Technology, 2011.

Robust Synchronization of Duffing System Using Integral Action in Backstepping Design

Shubhobrata Rudra, Dr. Ranjit kumar Barai, *Member, IEEE*, Dr. Madhubanti Maitra, *Member, IEEE*,
Rupam Kumar Dewan, Paramita Mandal, Dharmadas Mandal, KishoreKumar Kowdiki

Electrical Engineering Department
Jadavpur University
Kolkata, India

Abstract— Backstepping is a realistic nonlinear control design algorithm based on Lyapunov design approach; therefore, it automatically ensures the convergence of the regulated variable to zero. In this paper, it has been proposed for robust synchronization of Duffing chaotic systems. Integral action is being used to enhance the control action of the controller in steady state against the disturbances. The salient feature of the design is that the derived control doesn't contain any derivative terms; consequently it simplifies the controller realization. The effectiveness of the proposed controller has been demonstrated in simulation studies. The performance of the controller has been evaluated not only based on its synchronizing ability but also the disturbance rejection ability of the controller has been verified.

Keywords—component; Duffing Chaotic system, Synchronizing Problem, Backstepping, Integral Action.

I. INTRODUCTION

Synchronizing chaotic systems and circuits has received great interest in recent years since the seminal paper by Ott–Grebogi–Yorke [1]. Generally, the two chaotic systems in synchronization are called drive system and response system respectively. The idea of synchronization is to use the drive system output to control the response system output to track the output of drive system asymptotically. Some research activities on synchronization problem have been reported in recent articles [1-8]. However, synchronization problem for Duffing system triggered a new area of research, which is known as the issue of robust synchronization. Duffing system requires external excitation to exhibit synchronizing response and it is prone to external disturbances [8]. Hence, a robust controller which will able to ensure the guaranteed tracking performance of the system in presence of external disturbance becomes an impeccable choice for synchronizing control problem of Duffing system.

Backstepping is Lyapunov method based versatile robust control design approach for nonlinear systems and it ensures the convergence of the regulated variables to zero [9-11]. The main idea of the backstepping method is that the overall dynamic system is partitioned into two series cascaded subsystems.

Therefore, the states of the first subsystem are the control variables for the second. In backstepping approach, at first, the desired control input for the second subsystem is computed and then the control input for the first subsystem is computed to realize the desired state, which is the desired control input for

the second subsystem [10]. In this paper, a backstepping controller has been designed to address the synchronization of Duffing system and integral action is employed to enhance the robustness of the controller against unwanted disturbance signal [10]. The rest of the paper is organized as follows: In section II class of the state model of Duffing-system and synchronization problem has been presented. In section, III the design of Backstepping controller (with integral action) has been described in detail and a proof of stability of the controller has been given. Simulation results are presented in section IV. Section V concludes the work.

II. PROBLEM FORMULATION

The state model of drive System and response system are shown below:

$$x_1 = x_2 \quad (1)$$

$$\begin{pmatrix} \dot{x}_2 \\ \dot{y}_1 \end{pmatrix} = \begin{pmatrix} -ax_2 \\ -ay_1 \end{pmatrix} + \begin{pmatrix} bx_2 \\ by_1 \end{pmatrix} + \begin{pmatrix} c \cos(x_2) \\ c \cos(y_1) \end{pmatrix} + \begin{pmatrix} u \\ u \end{pmatrix} \quad (2)$$

By properly choose u , synchronization between response system (2) and drive system (1) can be achieved.

The state error between the states can be defined as follows:

$$e_1 = x_1 - y_1, \quad e_2 = x_2 - y_2 \quad (3)$$

Differentiation of equation (3) yields:

$$\dot{e}_1 = \dot{e}_2$$

$$e_2 = -ae_1 - be_2 - e_1(12 + 3x_1 e_1 + 3x_1^2) + c_1 \cos(t) - \cos(0.4t) + u$$

The problem to realize the synchronization between two chaotic systems now transforms to another problem on how to choose a control law u to ensure the asymptotic regulation of error variables (e_1 and e_2). The Backstepping control is designed along with integral action to enhance the steady state robustness of the system against unwanted external disturbance signal.

III. INTEGRAL ACTION BACKSTEPPING CONTROL DESIGN

The primary objective of the control system is to ensure the asymptotic regulation of the tracking error between drive system and response system. A Backstepping control along with integral action has been derived to ensure the proper synchronization of the Duffing chaotic system. The exploitation of integral action also ensures the robust disturbance rejection property of the controller.

A. Controller Design

For the regulation of first error variable e_1 , e_2 act as a virtual control input, to find an estimative stabilizing function

$\alpha_1(e_1)$ a control Lyapunov function $V_1 = \frac{1}{2}e_1^2$ has been selected.

$$V_1 = \frac{1}{2}e_1^2 \quad (5)$$

The choice of stabilizing function which will render the negative definiteness of the derivative of V_1 is $\alpha_1(e_1) = -c_1 e_1$, where c_1 is a positive design constant.

To introduce an integral action in steady state a suitable choice of virtual control law is given below in equation (5)

$$\alpha_1(e_1) = -c_1 e_1 - \lambda \int_0^t e_1 dt \quad (6)$$

where c_1 and λ_1 are positive design constants, and

$X_1 = \int_0^t e_1 dt$ which enhance the robustness of the controller against the external disturbance signal.

The error variable for second integrator can be defined as

$$z = -e_2 \alpha_1(e_1) = -e_2 - c_1 e_1 - \lambda \int_0^t e_1 dt \quad (7)$$

The state model of the system in (e_1, z) coordinate can be expressed as

$$\begin{aligned} \dot{e}_1 &= -z - c_1 e_1 - \lambda \int_0^t e_1 dt = -(b - c_1)z \\ &+ (bc - \lambda) \int_0^t e_1 dt - e_1(12 + 3x_1 e_1 + 3x_1^2) \\ &+ c_1 \cos(t) - \cos(0.4t) + u \end{aligned} \quad (8)$$

$$\dot{z} = -(bc - \lambda) \int_0^t e_1 dt - e_1(12 + 3x_1 e_1 + 3x_1^2) + c_1 \cos(t) - \cos(0.4t) + u$$

Now to ensure the stability of the overall system u should

choose such a way that it renders $\dot{z} = -c_2 z$ which will ensure the asymptotic stability of the synchronizing system.

A choice of such u is given in equation (8). The values of the controller gain are as follows: $c_1=1, c_2=1, \lambda=0.1, a=1.8, b=-0.1, c=-1.1$.

$$\begin{aligned} u &= -(b - c_1)z - (bc - \lambda) \int_0^t e_1 dt - e_1(12 + 3x_1 e_1 + 3x_1^2) \\ &+ (bc - \lambda) \int_0^t e_1 dt - e_1(12 + 3x_1 e_1 + 3x_1^2) \\ &- c_1 \cos(t) + \cos(0.4t) + u \end{aligned} \quad (9)$$

The final control Lyapunov function for the system is expressed in equation (9) below:

$$V_c = \frac{\lambda}{2} \int_0^t e_1^2 dt + \frac{1}{2} z^2 \quad (10)$$

The derivative of the V_c can be obtained from equation (8) and equation (9) is shown in following equation

$$\begin{aligned} \dot{V}_c &= \lambda \int_0^t e_1 \dot{e}_1 dt + z \dot{z} \\ &= -c_1 e_1^2 - c_2 z^2 \end{aligned} \quad (11)$$

B. Stability Analysis

The fact that $\dot{V} < 0$ from (11) implies that $V(t) \leq V(0)$, and therefore, that e_1 and z are bounded. Now, the following new function has been defined:

$$N(t) = c e_1^2 + c z^2 \quad (12)$$

Now, integrating (12) gives

$$\begin{aligned} V_c(t) &= V_c(0) + \int_0^t V_c(\tau) \tau d\tau \\ &= V_c(0) - \int_0^t N(\tau) d\tau \end{aligned} \quad (13)$$

Thus

$$\int_0^t N(\tau) \tau d\tau = V_c(0) - V_c(t) \quad (14)$$

Considering $\dot{V} < 0$ and $V(t) > 0$ the following results can be easily derived:

$$\lim_{t \rightarrow \infty} \int_0^t N(\tau) \tau d\tau < \infty \quad (15)$$

To use Barbalat's lemma, let us check the uniform continuity of $V_c(t)$. The derivative of $V_c(t)$ is

$$\dot{V}_c(t) = 2[c e e_{11} + c z z_2] \quad (16)$$

This shows that $\dot{V}_c(t)$ is bounded, because e & z

are bounded. Therefore $V_c(t)$ is uniformly continuous.

Through Barbalat's lemma, it can be shown that e_1 and z converge to zero as $t \rightarrow \infty$.

IV. SIMULATION RESULTS

The synchronization of the states of drive system and response system has been shown in Fig. 1(a) and Fig 1.(b). Initial value of the states are chosen as follows: $x_1(0)=1$, $x_2(0)=2$, $y_1(0)=1$ and $y_2(0)=2$.

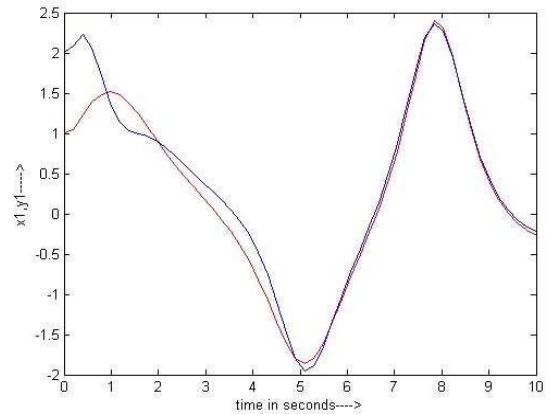


Fig. 1.(a) Variation of States x_1 & y_1

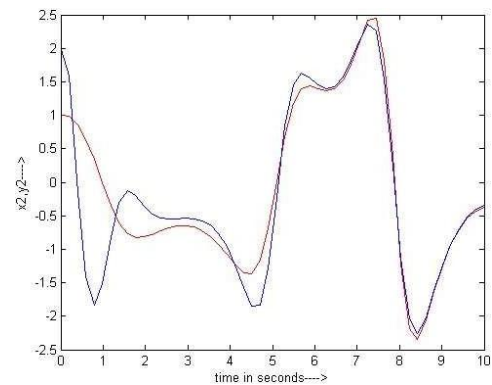


Fig. 1.(b) Variation of States x_2 & y_2

To evaluate the robustness of the controller in another run a white band noise has been added with the response system input.

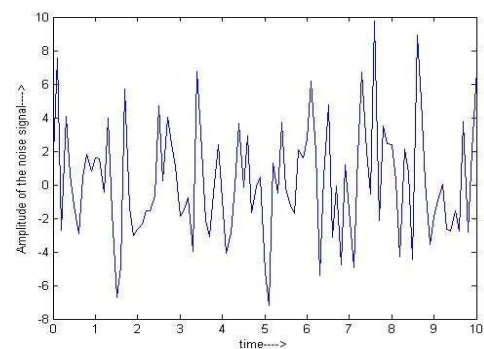


Fig. 2. Noise Signal

The responses of the states are shown in figure 3(a) and 3(b).

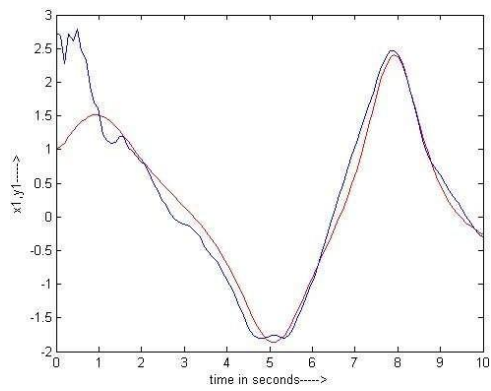


Fig.3.a. State Response of x_1 & y_1

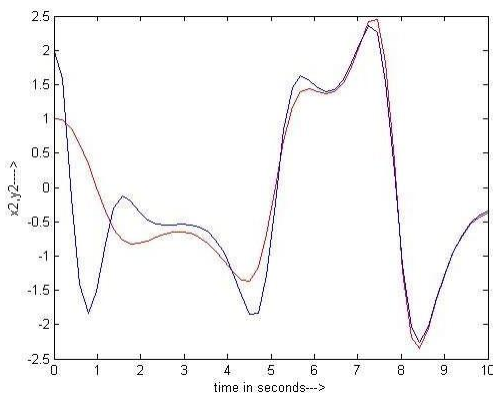


Fig.3.b. State Response of x_2 & y_2

- [6] Liao T-L, Tsai S-H. Adaptive synchronization of chaotic systems and its application to secure communications. *Chaos, Solutions & Fractals* 2000;11:1387-96.
- [7] Bai E-W, Lonngren KE. Synchronization and control of chaotic systems. *Chaos, Solutions & Fractals* 1999;10:1571-5.
- [8] X.Tan, J Jhang and Y. Yang, Synchronizing chaotic systems using backstepping design. *Chaos, Solutions & Fractals* 2003 37-45
- [9] P.V. Kokotovic and M. Arcaak, "Constructive nonlinear control: a historical perspective," *Automatica*, vol.37, pp. 637-662, 2001.
- [10] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*, New York; Wiley Interscience, 1995. [11] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, 1996.

Hence from the above experimental results it can be concluded that the controller is able to exhibit stable performance in face of bounded disturbance added to the response system.

V. CONCLUSION

In this paper, backstepping design has been used to synchronize chaotic systems. Integral action has been explored to enhance the robustness of the controller. The advantages of this method can be summarized as follows: (a) it is a systematic procedure for synchronizing Duffing system; (b) it is able to offer a robust performance in presence of external disturbance signal. The simulation results reveal that the proposed controller exhibits an excellent synchronizing ability for the Duffing system when it is subjected to some external disturbances. Finally, to conclude, authors take the liberty to claim that the control law design proposition for Duffing system is Robust enough to ensure the proper synchronization in presence of a large disturbance.

REFERENCE

- [1] Ott E, Grebogi C, Yorke JA. Controlling chaos. *Phys Rev Lett* 1990;64:1196-9.
- [2] Carroll TL, Pecora LM. Synchronizing chaotic circuits. *IEEE Trans Circ Syst I* 1991;38:453-6.
- [3] Bai EW, Lonngren EE. Synchronization of two Lorenz systems using active control. *Chaos, Solutions & Fractals* 1997;8:51-8.
- [4] Liao TL. Adaptive synchronization of two Lorenz systems. *Chaos, Solutions & Fractals* 1998;9:1555-61.
- [5] Cuomo KM, Oppenheim AV, Strogatz SH. Synchronization of Lorenzbased chaotic circuits with applications to communications. *IEEE Trans Circ Syst I* 1993;40:626-33.

Detection of Intruders and Flooding in VoIP using IDS, Jacobson Fast and Hellinger Distance Algorithms

A. Rahul,
Assistant Professor,
Department of C.S.E,
Vardhaman College of Engineering
Hyderabad-501218, A.P., India
arirahul539@gmail.com

B.Suresh kumar
M.Tech (C.S.E)
Department of C.S.E
Vardhaman College of Engineering
Hyderabad-501218, A.P., India
sureshkumargoud2006@gmail.com

S.K.Prashanth
Associate Professor
Department of C.S.E
Vardhaman College of Engineering
Hyderabad-501218, A.P., India
sk_p21@yahoo.co.in

Abstract -- VoIP services are becoming increasingly a big competition to existing telephony services (PSTN). Hence, the need arises to protect VoIP services from all kinds of attacks that target network bandwidth, server capacity or server architectural constrains. SIP Protocol is used for VoIP connection establishment. It works based on either TCP or UDP Protocols. This protocol structure is almost as same as HTTP Protocol, i.e. for every request there will be some response, even though the request is invalid. HTTP Protocol is prone to flooding attacks, like SYN-Flood attack. Because of Session Initiation Protocol (SIP) is same as HTTP, SIP is also prone to Flooding attacks. The proposed Intrusion Detection System (IDS) is used to detect the intruders in telephony system. Genetic algorithm is used to recognize the authorized user. VoIP Flood Detection System (VFDS) is aimed to detect TCP Flooding attacks and SIP Flooding attacks on SIP devices using Jacobian Fast and Hellinger distance algorithms. The Jacobian Fast Algorithm fixes the threshold limit and Hellinger distance calculation is a statistical anomaly based algorithm uses to detect deviation in traffic

Key words: VoIP, attacks in VoIP, flood attacks, IP telephony, Jacobian Fast, Hellinger Distance, Intrusion Detection System, Genetic algorithm.

I. INTRODUCTION

At its simplest, Voice over Internet Protocol (VoIP) is the transport of voice using the Internet Protocol (IP), however this broad term hides a multitude of deployments and functionality and it is useful to look in more detail at what VoIP is being used for today. Currently the following types of VoIP applications are in use:

Private users who are using voice over IP for end to end phone calls over the public internet. These users typically trade quality, features and reliability for the fact

that the service is very low cost and are generally happy with the service.

Business users on private networks provided by telecom and datacom providers. These services offer relatively high quality and reliability and are feature rich but come at a price.

IP trunking solutions used by long haul voice providers. Typically these offerings use private IP networks to connect islands of the PSTN together, e.g. a low cost way of calling the USA from the UK. Customers access these services using traditional black phones but the voice is carried over an IP network.

The actual problem lying in the VoIP scenario is that the VoIP servers are vulnerable to Intrusion, DoS attacks. We construct the IDS to avoid intruders based on Genetic Algorithm and VFDS (VoIP Flooding Detection System) that uses the Hellinger Distance algorithm to conclude whether there is any flooding or not. Detection of flood is based on the threshold level which is calculated using Jacobson's fast algorithm.

II. INTRODUCTION TO IDS

Intrusion is the formal term describing the act of compromising a system. Intruder is an unauthorized user who is going to steal the information. Intruder may be insider or outsider .80% of security breaches are committed by insider. Outsiders attempt to go around firewall to attack machines on internet network. He may come from internet, dial-up lines, physical break-ins. IDS [1] is used to detect the intruders which reduce the unauthorized traffic in the network based on Genetic Algorithm [1]. Chromosome conversion [1] is used to convert packets in to chroms and it performs two basic operations as crossover (rotation) and mutation (interchange). Genetic algorithm identifies the authorized user based on the Data Set. Data Set is a Set just acts like a Database which contains the record of authorized users. The below Fig: 1 illustrates the above information. IDS is followed by VFDS is to detect the flooding is discussed in the further topics.

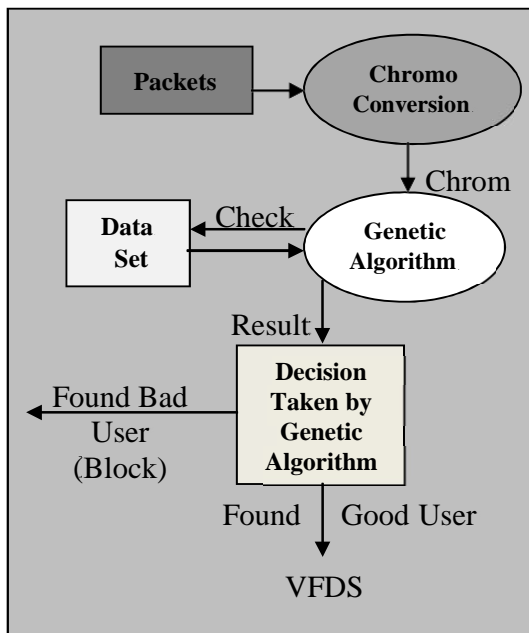


Fig.1. Intrusion Detection System Architecture

III. INTRODUCTION TO SIP NETWORKS

The SIP [4] has established itself as the de-facto standard for VoIP services. Several providers are already offering Internet Tele-phony services based on SIP. The popularity of SIP will likely rise even more with the advent of the IP Multimedia System (IMS), the nextgeneration telephony network which is also based on SIP.

SIP is deployed in a client-server infrastructure where SIP clients (User Agent clients, UAC) contact a central SIP Proxy (i.e. a server) to manage ongoing sessions. It is a text based protocol designed to establish or terminate a session among two or more partners. The message format is derived from the HTTP protocol, with message headers and corresponding values, e.g. —From: user@sip.org| to denote the sender of a message.

As it is generally deployed in the open internet, SIP infrastructures can easily become a target for different attacks from the outside world, including Denial-of-Service attacks, Message Tampering, Call Hijacking and other threats.

Here it is presented a VFDS [2], an open security architecture that is designed to monitor the traffic flow between SIP servers and external users and proxies. The goal is to detect attacks directed at the

protected SIP servers and provide a framework for attack prevention / mitigation.

Our focus lies especially on high traffic flooding attacks that can easily overwhelm a proxy's resources in terms of CPU processing power, memory requirements or bandwidth capacity. Hence, VFDS was designed with scalability in mind. As such, a layered solution with dedicated tasks (traffic monitoring, analysis, decision) with each layer optimized for scalability also presented here. The SIP traffic is forwarded to individual intelligent extension modules which provide monitoring, attack detection.

The proposed architecture is described in section I, II, IV. The approach is discussed in Section V, VI and results are shown in Section VII. Conclusion and Future work is presented in section VIII followed by References in Section IX.

The rapid evolution of voice and data technology is significantly changing the business environment with such services such as instant messaging, integrated voice and e-mail, and follow-me services—all offering an environment where people can communicate much more efficiently. To meet the demands of the changing business environment businesses are beginning to deploy converged voice-and-data networks based on SIP.

SIP was originally defined in 1999, by the Internet Engineering Task Force (IETF) in RFC 2543. The definition was the culmination of years of work in the IETF MMUSIC Working Group to provide a mechanism to allow voice, video, and data to be integrated over the same network. SIP provides the foundation for building converged networks that support seamless integration with traditional voice networks, e-mail, the World Wide Web, and next-generation technologies such as instant messaging.

As businesses continue to increase their use and reliance on converged services, reliability and availability becomes increasingly important. This paper introduces the key elements of a SIP network, further define the concept of high availability in SIP networks, and explore various techniques to increase the availability of SIPbased VoIP networks.

As shown in Fig.2 a SIP-based network consists of:

- 1.) **SIP User Agent**—any network endpoint that can originate or terminate a SIP session. This may include a SIP-enabled telephone, a SIP PC client (known as a “soft phone”), or a SIP-enabled gateway.
- 2.) **SIP Proxy Server**— A call-control device, such as the Cisco SIP Proxy Server, that provides many services such as routing of SIP messages between SIP user agents
- 3.) **SIP Redirect Server**—A call-control device that provides routing information to user agents when requested, giving the user agent an alternate uniform resource identifier (URI) or destination user agent server (UAS).

4.) SIP Registrar or Location Server—A device that

stores the logical location of user agents within that domain or sub-domain. A SIP registrar server stores the location of user agents and dynamically updates its data via REGISTER messages.

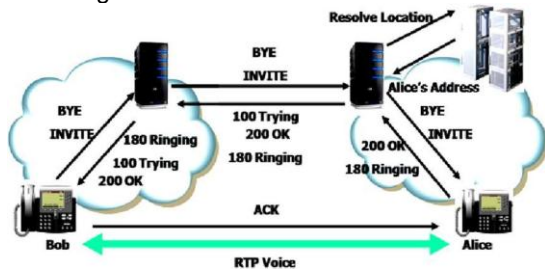


Fig.2. SIP Message flow during Connection Establishment

The recent attacks on popular web sites like Yahoo, eBay and E*Trade, and their consequent disruption of services have exposed the vulnerability of the Internet to Distributed Denial of Service (DDoS) attacks [3]. It has been shown that more than 90% of the DoS attacks use TCP. The TCP SYN flooding is the most commonly-used attack. It consists of a stream of spoofed TCP SYN packets directed to a listening TCP port of the victim. Not only the Web servers but also any system connected to the Internet providing TCP-based network services, such as FTP servers or Mail servers, are susceptible to the TCP SYN flooding attacks. The SYN flooding attacks exploit the TCP's three-way handshake [5] mechanism and its limitation in maintaining half-open connections. When a server receives a SYN request, it returns a SYN/ACK packet to the client. Until the SYN/ACK packet is acknowledged by the client, the connection remains in halfopen state for a period of up to the TCP connection timeout, which is typically set to 75 seconds. The server has built in its system memory a backlog queue to maintain all half-open connections. Since this backlog queue is of finite size, once the backlog queue limit is reached, all connection requests will be dropped.

If a SYN request is spoofed, the victim server will never receive the final ACK packet to complete the three-way handshake. Flooding spoofed SYN requests can easily exhaust the victim server's backlog queue, causing all the incoming SYN requests to be dropped. The stateless and destination-based nature of Internet routing infrastructure cannot differentiate a legitimate SYN from a spoofed one, and TCP does not offer strong authentication on SYN packets. Therefore, under SYN flooding attacks, the victim server cannot single out, and respond only to, legitimate connection requests while ignoring the spoofed.

To counter SYN flooding attacks, several defense mechanisms have been proposed. All of these defense mechanisms are installed at the firewall of the victim server or inside the victim server, thereby providing no hints about the sources of the SYN flooding. They have

to rely on the expensive IP trace back to locate the flooding sources. Because the defense line is at, or close to, the victim, the network resources are also wasted by transmitting the flooding packets.

Therefore, a simple stateless mechanism to detect SYN flooding attacks, which is immune to the SYN flooding attacks. Also, it is preferred to detect an attack early near its source, so that one can easily trace the flooding source without resorting to expensive IP trace back.

IV. THE VFDS DESIGN

The VFDS consists of training phase and testing phase where packets are captured and HD is calculated as shown in Fig: 3. Generally VFDS detects anomalies in collections of packet streams, going through a cyclic behavior consisting of two phases: the training and testing phases.

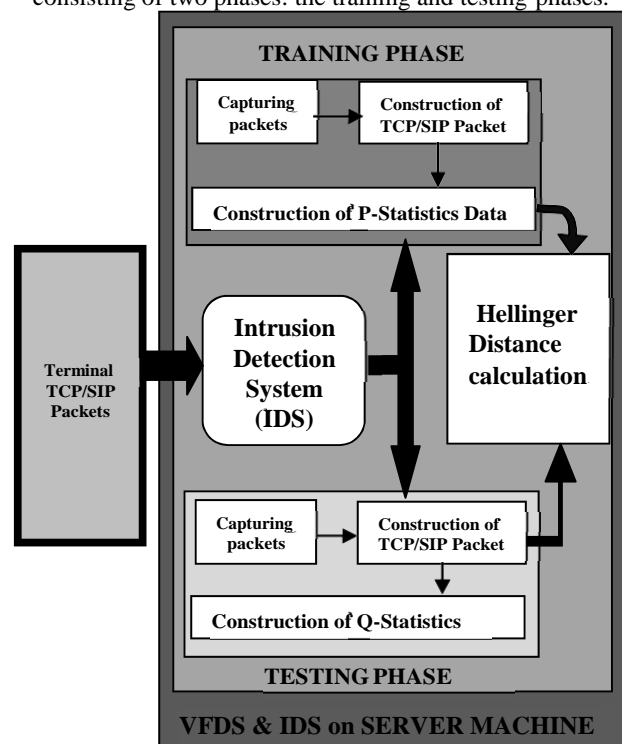


Fig.3. VFDS Design Architecture

As shown in Fig. 4, during the training phase, the training data set consisting of the attribute set is collected over n sampling periods of duration Δt over a normal traffic stream. This initial training data set is assumed to be devoid of any attacks and acts as a base for comparing with the next $(n+1)$ th periods of the testing data set. Using the soon-to-be-described HD, we measure the distance between these two data sets. If the measured distance exceeds a threshold, an alarm is raised; otherwise, the testing data set is included

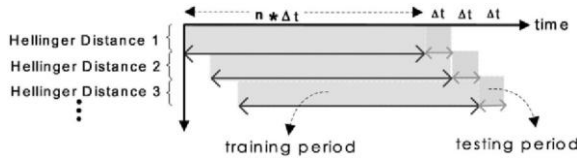


Fig.4. Relationship between training and testing periods

In the immediately (n-1) sampled traffic data to derive a new training data set. This moving window mechanism helps the training data set to adapt with the dynamics of network traffic. In order for this design to work, the following three parameters are computed online:

Using HD (soon to be described) we calculate the following:

1. The probabilistic distribution for training data. This computed as the ratio of packets that satisfy the feature to the total number of packets received during the training phase.

2. The probabilistic distribution for testing data. This computed as averages during the time window immediately following the training period, again as a ratio of packets satisfying the chosen feature to the total number of packets, whereas the deviation of the two probability distributions are computed using the (soon-to-be-described) HD.

3. The threshold of deviation to distinguish normal behavior from the abnormal behavior. This is used to compute a dynamic threshold as the computation progresses through cycles of training and testing phases, using Jacobson's fast deviation computing algorithm [5].

V. HELLINGER DISTANCE

Hellinger distance presents an intrinsic way to estimate the distances between probability measures independent of the parameters. It is closely related to the total variation distance [6] but with several advantages. To explain this, let P and Q be two probability distributions on a finite sample space Ω , where P and Q on Ω are N-tuples $(p_1, p_2, p_3, \dots, p_N)$ and $(q_1, q_2, q_3, \dots, q_N)$ respectively, satisfying (in)equalities $p_\alpha \geq 0, q_\alpha \geq 0, \sum_\alpha p_\alpha = 1$. Then, the HD between P and Q is defined as

$$d_H^2(P, Q) = \frac{1}{2} \sum_{\alpha=1}^N (\sqrt{P_\alpha} + \sqrt{Q_\alpha})^2 \quad (1)$$

The HD satisfies the inequality $d_H \leq 1$ when $P=Q$. The distance between P and Q shows the maximum distance of one. Sometimes, the factor $\frac{1}{2}$ is not used in the above equation. A related notion is the affinity between probability measures, which is defined as

$$A(P, Q) = 1 - d_H^2(P, Q) = \sum_{\alpha=1}^N (\sqrt{P_\alpha Q_\alpha}) \quad (2)$$

The affinity between two probability measures P and Q is one (that is, $A = 1$) if they are equal and zero if the measures are totally different. Further details on HD can be found in (5).

A. Measuring Protocol Deviations Using the Hellinger Distance

In order to detect protocol violations, depending upon the protocol to be observed and a collection of potential attacks that can be launched against it, we select and track the distribution of a (small) set of attributes. Suppose we choose N attributes of a protocol, which satisfy $p_\alpha, q_\alpha \geq 0, \sum_\alpha p_\alpha = 1, \sum_\alpha q_\alpha = 1$ and Here, α represents an attribute in the chosen set of N attributes. Probability measure P is defined over the training data set, whereas probability measure Q is defined over the testing data set. Both P and Q are hypothesized to be an array of the normalized frequencies of all N attributes.

B. Detection Threshold

Normal attribute behaviors also change with time, although the strong attribute correlation makes the fluctuation of its dynamics much less than that of traffic behaviors. To accurately keep track of the normal attribute behaviors, we use a dynamic threshold for detection. Such a dynamic setting of threshold will make an attack harder to evade. We employ the stochastic gradient algorithm to compute the dynamic threshold based on the HD observed during the previous training period. Our threshold is an instance of Jacobson's Fast algorithm for RTT mean and variation [6]. Fast estimators for average a and mean deviation v , given measurement m , can be computed as

$$\text{Err} = m_n - a_{n-1}, \quad (3) \quad a_n \leftarrow a_{n-1} + g \cdot \text{Err}, \quad (4)$$

$$v_n \leftarrow v_{n-1} + h \cdot (|\text{Err}| - v_{n-1}), \quad (5)$$

Where m_n is the current sample of the HD, a_{n-1} and a_n are the previous and current smoothed Hellinger distances, respectively, and v_{n-1} and v_n represent the previous and current mean deviations. To make the computation efficient, g and h are chosen to be negative exponents of two. Here, we use the values $g=1/2^3$ and $h=1/2^3$, as previous research suggested [7&8]. Although

the original g and h are used in the context of RTT measurement, the underlying principles of both cases are the same: based on the past and present values, we attempt to predict the future values. The smoothed HD is based on the observed HD m , which is measured between the probability measures P and Q. During the testing periods, we derive the estimated threshold HD (HD^{thres}) using the smoothed HD (2) and the mean deviation (3):

$$HD_{n+1}^{thres} = X * a_n + \eta * v_n \quad (6)$$

The purpose of the multiplication factors X and η is to get a safe margin for the setting of the threshold value, so that VFDS avoids any false alarms without degrading its

detection sensitivity. The first factor in (4), which largely depends upon the observed HDs, should be large enough to make the first part of (4) higher than the maximum observed HD, whereas the second factor is tied with the variations of these observed Hellinger values. These two factors are adjustable parameters and can be properly tuned during the training period.

VI. DEPICTING PROTOCOL BEHAVIORS

We used SIP Packet generating tools which is capable of generating SIP, TCP, UDP packets, to experimentally profile normal protocol behaviors.

A. Observation of TCP Packets

In order to study the attribute behaviors of VoIP traffic, we build a testbed. The testbed consists of a PC(Intel P-IV 2.4GHz, 1GB RAM) in which a virtual LAN is configured with VMWARE in which Linux operating system is installed out of which one System is acting as Proxy Server and the other one is acting as SIP clients. Enterprise networks A and B are simulated within same PCs equipped with SIP traffic generators in client machine and VFDS is installed in Proxy Server playing the role of multiple systems.

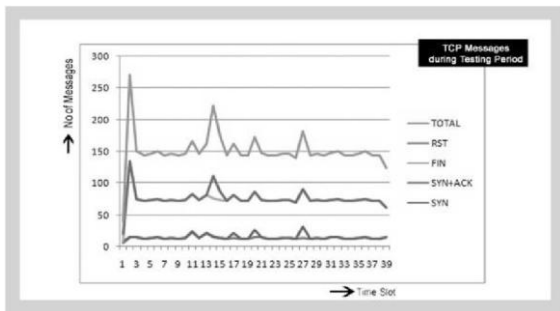


Fig.5. TCP attribute behavior during training

On the server we install our VFDS in which we will capture the incoming packets generated by sip tool and find out how many no of packets arrived in testing period. During experiment for the duration of one minute a total of 2225 SYN and 12050 FIN packets were arrived in testing period. The statistics during training period for the duration of one minute a total of 519 SYN Packets and 2326 FIN packets were arrived as shown in fig 5 and fig 6 shows the statistics during testing period.

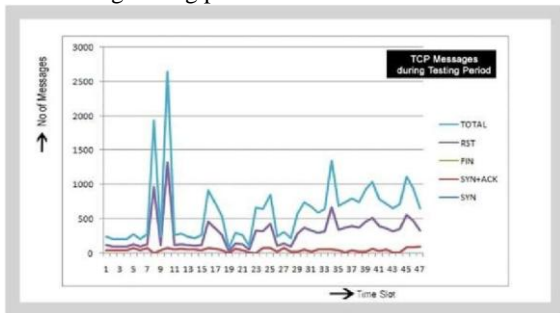


Fig.6. TCP attributes behavior during testing periods.

B. Observation of SIP Packets

To study the traffic of the SIP packets we have selected Sip packet generator tool. During the training phase we collected a data of 2323 INVITE packets and 4650 no of 200-OK packets in testing period a total number of 27583 INVITE packets and 32925 no of 200OK packets were arrived. The SIP traffic during training and testing periods are shown the graphs in fig 7 and fig 8.

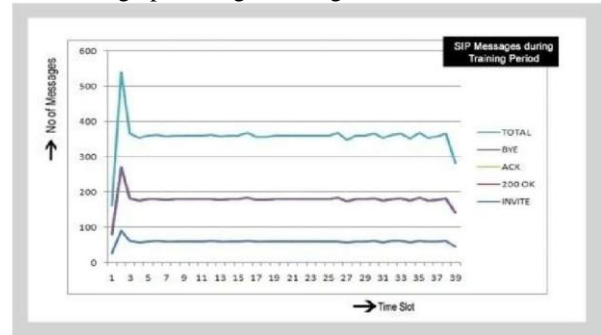


Fig. 7. SIP-attribute behavior during Training Period

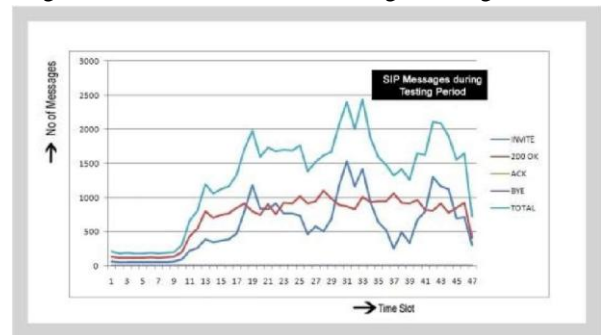


Fig.8. SIP-attribute behavior during Training Period

VII. COMPUTING THE HELLINGER DISTANCE FOR TCP

In this experiment, we choose four attributes SYN, SYN-ACK, FIN for the calculation of the TCP HD values as shown in fig 8.

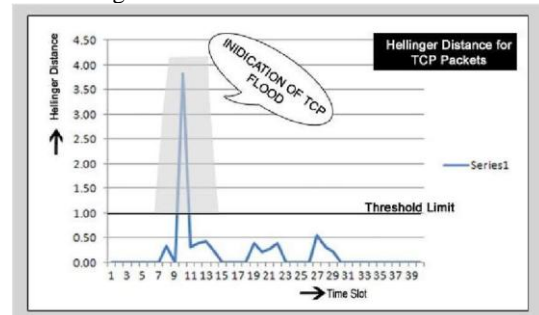


Fig.9. HD of TCP attributes.

Now, suppose that there are NSYN, NSYN_ACK, NFIN, and NRST packets during the training period (that is, in $n \cdot \Delta t$ time). P is an array of the normalized frequencies of pSYN, pSYN_ACK, pFIN, and pRST over the training data set, and Q is an array of the normalized frequencies of qSYN, qSYN_ACK, qFIN, and qRST of the same attributes observed over the testing period (that is, at the $(n + 1)$ th sampling duration), defined as follows:

$$p_\alpha = N_\alpha / N_{Total} \quad (7)$$

Where $\alpha \in \{SYN, SYN - ACK, FIN, RST\}$, and

$$N_{Total} = (NSYN + NSYN_ACK + NFIN + NRST) \quad (8)$$

$$p_\alpha = N_{I\alpha} / N_{ITotal} \quad (9)$$

Where $\alpha \in \{SYN, SYN - ACK, FIN, RST\}$, and $N^{I}_{Total} = (N^I_{SYN} + N^I_{SYN_ACK} + N^I_{FIN} + N^I_{RST}) \quad (10)$

The HD between IP and QQ at the end of on p 1Pth sampling period is computed as follows:

$$HD = (\sqrt{P_{SYN}} - \sqrt{Q_{SYN}})^2 + (\sqrt{P_{SYN_ACK}} - \sqrt{Q_{SYN_ACK}})^2 + (\sqrt{P_{FIN}} - \sqrt{Q_{FIN}})^2 + (\sqrt{P_{RST}} - \sqrt{Q_{RST}})^2 \quad (11)$$

A. Computing the Hellinger Distance for SIP

We choose to experiment with SIP attributes INVITE, 200 OK, ACK, and BYE. Here, the probability measure P is an array of the normalized frequencies of PINVITE, P200 OK, PACK, and PBYE over the training data set. Similarly, Q is an array of QINVITE, Q200 OK, QACK, and QBYE during the chosen testing period. All other details are similar to the previous example. To calculate the HD between P and Q, we use

$$HD = (\sqrt{P_{INVITE}} - \sqrt{Q_{INVITE}})^2 + (\sqrt{P_{200\ OK}} - \sqrt{Q_{200\ OK}})^2 + (\sqrt{P_{ACK}} - \sqrt{Q_{ACK}})^2 + (\sqrt{P_{BYE}} - \sqrt{Q_{BYE}})^2 \quad (12)$$

Fig. 10 shows the HD for the SIP attribute set of {INVITE, 200 OK, ACK, BYE}.

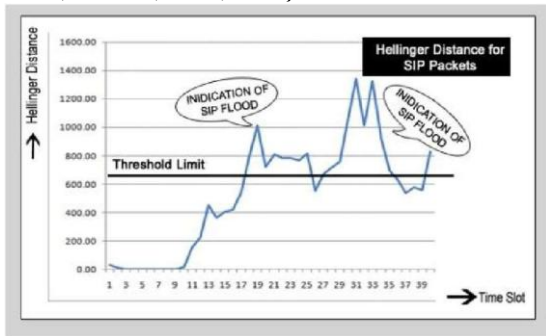


Fig.10. HD of SIP attributes.

Fig. 10 illustrates the dynamics of the SIP data estimated threshold HD and the observed HD. Because the spikes of the observed HD are much higher than those of the estimated threshold distance, no false alarm is raised.

VIII. CONCLUSIONS AND FUTURE WORK

SYN, INVITE, and TCP packet floods pose a serious threat to the IP telephony infrastructure. The multiprotocol-based VoIP service needs a fast and generic detection mechanism working across different protocol layers. Here it is investigated, the protocol attribute behaviors and characterize the network traffic with respect to the intrinsic correlation among protocol attributes. Utilizing HD, it is presented an online statistical flooding detection mechanism, called VFDS, in which we measure the similarity (or dissimilarity) of the correlation among protocol attributes at different times. The rationale behind our approach is that a deviation from normal protocol behaviors can be measured and quantified.

In this paper it was reviewed the concepts of the reference [2]. A trial to simulate and develop a new framework to prove that, this mechanism will also work for low traffic and highly congested traffic and to avoid false alarm.

IX. REFERENCES

- [1] Mohammad Sazzadul hoque, Md. Abdul mukit2 and Md. Abu Naser bikas —An implementation of intrusion detection system using genetic algorithm — International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012
- [2] Hemant Sengar, Haining Wang, Duminda Wijesekera, and Sushil Jajodia, —Detecting VoIP Floods Using the Hellinger Distancel, IEEE Transactions On Parallel And Distributed Systems, Vol. 19, No. 6, June 2008.
- [3] .R Jones, J Cruz, "Carrier Class Voice over IP", August 1999, 9 pages..
- [4] M. Handley, ACIRIH .Schulzrinne Columbia U E. Schooler Cal Tech J. Rosenberg, Bell Labs, March 1999 "SIP: Session Initiation Protocol", March 1999, 132 pages, ftp://ftp.isi.edu/in-notes/rfc2543.txt
- [5] Rakesh Arora, Voice Over IP: Protocols and Standards http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols/index.html ,
- [6] V. Jacobson and M.J. Karels, —Congestion Avoidance and Control, Proc. ACM SIGCOMM '88, pp. 314-329, Aug. 1988.
- [7] W. Stevens, TCP/IP Illustrated Volume-1, first ed. Addison-Wesley, 1994.
- [8] D. Pollard, Asymptopia, first ed., book in progress, <http://www.stat.yale.edu/pollard/>, 2000.

A Mathematical Model of Blood Flow in a Catheterized artery with multiple stenoses

¹B.Basu Mallik & ²Saktipada Nanda

Department of Basic Science & Humanities

Institute of Engineering & Management, Salt Lake Electronics Complex
Kolkata - 700091, West Bengal, India.

e-mail: ¹b.basumallik@gmail.com ²saktipada_nanda@yahoo.com

Abstract— A mathematical model is developed in this investigation for studying the axi-symmetric flow of blood through a catheterized artery with multiple stenoses. Consideration of Newtonian character of blood is described following the report of Young (1968) and Srivastava (2009) with the appropriate constitutive equation governing the flow. The boundary conditions appropriate to the problem under study are the standard no slip conditions at the artery and the catheter wall. Analytical expressions for impedance (flow resistance), the wall stress distribution in the stenotic region and the shear stress at the stenosis throat in their non dimensional form are derived by using the model. The derived expressions are computed numerically and the results are presented graphically for different values of the rheological and other parameters. The study provides an insight into the effects of catheter radius and stenosis height on impedance, wall stress distribution in the stenotic region and the shear stress at the stenotic throat.

Keywords- Stenosis, Catheter, Shear stress, Impedance, stenotic throat, Newtonian flow, ischemia, thrombosis.

I. INTRODUCTION

The understanding of anatomy and physiology of an organic system depends much on the knowledge of blood flow through arteries. The cause and development of many arterial diseases are related to the flow characteristics of blood and the mechanical behavior of the blood vessel walls. The abnormal and unnatural growth in the arterial wall thickness at various locations of the cardio vascular system is medically termed ‘stenosis’. Its presence in one or more locations restricts the flow of blood through the lumen of the coronary arteries into the heart leading to cardiac ischemia. Once the constriction develops, it brings about significant alterations in the blood flow, pressure distribution, wall shear stress and the impedance (flow resistance). The fact that haemodynamic factors play a commendable role in the genesis and the growth of the disease has attracted many researchers to explore modern approach and more and more sophisticated mathematical models for investigation on flow through stenotic arteries.

To illuminate the effects of stenosis present in the arterial lumen, intensive experimental and theoretical investigations have been carried out world wide for both normal and stenotic arteries. In most of the investigations relevant to the domain under discussion, the Newtonian behavior of blood (single phase homogeneous viscous fluid) was accepted. This model of blood is acceptable for high shear rate in case of a blood flow through narrow arteries of diameter $\leq 1000 \mu\text{m}$.

In modern medicinal science, catheters may play an important role in the diagnosis and treatment of the disease-termed medically as ‘Stenosis’. The technique is almost same as employed for measuring blood pressure and other mechanical properties of the arteries. Due to the insertion of catheter in a stenosed artery, the frictional resistance to flow will remarkably change the velocity distribution. An improved model to treat arteriosclerosis is balloon angioplasty where a tiny balloon is attached to the tip of a catheter of appropriate size and carefully guided to the location where the stenosis (single or multiple) has developed. Finally the balloon is inflated by giving air from outside leading to fracture of the undesirable deposits and widening of the narrow path.

Srivastava and Srivastava [11] recently presented a review of the particulate suspension of blood flow through narrow catheterized artery. Kanai et al. [6] studied the problem in the measurement of blood pressure by artery catheterization. Gunj et al. [5], Anderson et al. [1] and Wilson et al. [14] observed the measurement of transstenotic pressure gradient during coronary angioplasty. Leimgraber et al.[7] have measured high mean pressure gradient across stenosis in coronary angioplasty. Back [2] and Back et al. [3] studied the increase of average flow resistance during artery catheterization in normal (without stenosis) as well as in stenosed arteries. Sarkar and Jayaraman [9] observed the change in flow pattern of pulsatile blood flow due to steady streaming effect in a catheterized artery during balloon angioplasty. Dash and Jayaraman [4] studied the problem of blood flow in a catheterized curved stenosed artery. Sankar and Hemlatha [8] discussed the pulsatile flow of blood in a stenosed catheterized

artery by modeling it as Herschel- Bulkley fluid. Srivastava and Rastogi [12, 13] studied the flow of blood by considering it to be a macroscopic two phase model in a stenosed catheterized artery.

In most of the investigations mentioned earlier, single stenosis (symmetric or non-symmetric) models were considered. On the basis of experimental observations it is established that stenosis may develop in series or may also overlap.

In this analytical study an effort is made to present some new characteristics of blood flow in stenosed arteries which were not given due attention by previous researchers though they may have a significant role in the diagnosis and treatment of this fatal disease. The Newtonian fluid model of blood is accepted for the present dissertation. Also the length of the artery is taken large compared to its diameter so that some special wall effects may be neglected.

II. MODEL DESCRIPTION

Let us consider the axi-symmetric flow of blood through a catheterized artery with a composite stenosis. The geometry of the stenosis is assumed to be manifested in the arterial segment given by is described in Fig.1 as:

$$\frac{R(z)}{R_0} = 1 - \frac{2\delta}{R_0 L_0} (z - d); d \leq z \leq d + \frac{L_0}{2} \quad (1)$$

$$= 1 - \frac{\delta}{2R_0} \left\{ 1 + \cos \frac{2\pi}{L_0} \left(z - d - \frac{L_0}{2} \right) \right\}; d + \frac{L_0}{2} \leq z \leq d + L_0 \quad (2)$$

$$= 1; \quad \text{otherwise} \quad (3)$$

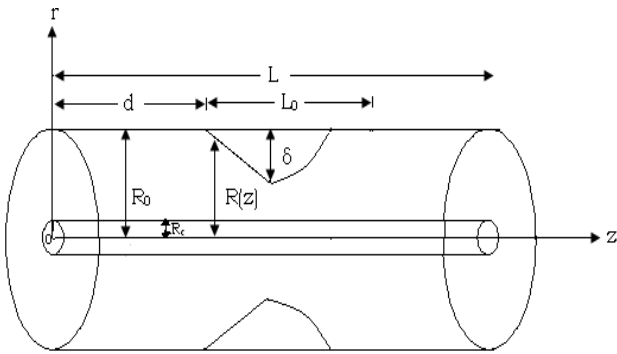


Fig1.Geometry of a composite stenosis in a catheterized artery

where $R(z)$ and R_0 are the radius of the artery with and without stenosis respectively, R_c is the radius of the catheter. L_0 is the length of the stenosis and d indicates its location. δ is the maximum height of stenotic growth at $z = d + L_0/2$.

Here blood is assumed to be represented by a Newtonian fluid. Considering the axisymmetric, laminar, steady; one

dimensional flow of blood in an artery, the governing equation in a mild stenosis may be stated as (Young [15]; Srivastava and Rastogi [12]):

$$\frac{dp}{dz} = \frac{\mu}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) u \quad (4)$$

where (r,z) are cylindrical polar coordinate system where z is measured along the axis of the tube, p is the pressure, u and μ are the fluid velocity and viscosity.

The boundary conditions are

$$u = 0 \quad \text{at } r = R(z) \quad (5)$$

$$u = 0 \quad \text{at } r = R_c \quad (6)$$

Integrating equation (4) and using the boundary conditions (5) and (6), the expression for the velocity is given by

$$u = -\frac{R_0^2}{4\mu} \frac{dp}{dz} \left[\left(\frac{R}{R_0} \right)^2 - \left(\frac{r}{R_0} \right)^2 + \frac{\left\{ \left(\frac{R}{R_0} \right)^2 - \left(\frac{R_c}{R_0} \right)^2 \right\}}{\log \left(\frac{R}{R_c} \right)} \log \left(\frac{r}{R} \right) \right] \quad (7)$$

The flow flux, Q is obtained as

$$Q = 2\pi \int_{R_c}^R r u dr$$

$$= -\frac{\pi R_0^4 \left\{ \left(\frac{R}{R_0} \right)^2 - \varepsilon^2 \right\}}{8\mu} \frac{dp}{dz} \left[\left(\frac{R}{R_0} \right)^2 + \varepsilon^2 - \frac{\left\{ \left(\frac{R}{R_0} \right)^2 - \varepsilon^2 \right\}}{\log \left\{ \left(\frac{R}{R_0} \right) / \varepsilon \right\}} \right] \quad (8)$$

$$\text{where } \varepsilon = \frac{R_c}{R_0}$$

From equation (8) we obtain

$$-\frac{dp}{dz} = \frac{8\mu Q}{\pi R_0^4} \phi(z) \quad (9)$$

$$\text{where } \phi(z) = \frac{1}{F(z)}$$

$$F(z) = \left\{ \left(\frac{R}{R_0} \right)^2 - \varepsilon^2 \right\} \left[\left(\frac{R}{R_0} \right)^2 + \varepsilon^2 - \frac{\left\{ \left(\frac{R}{R_0} \right)^2 - \varepsilon^2 \right\}}{\log \left\{ \left(\frac{R}{R_0} \right) / \varepsilon \right\}} \right]$$

The pressure drop, Δp ($= p$ at $z = 0$, $-p$ at $z = L$) across the stenosis in the tube of length, L is obtained as

$$\begin{aligned} \Delta p &= \int_0^L \left(-\frac{dp}{dz}\right) dz \\ &= \frac{8\mu Q}{\pi R_0^4} \psi \end{aligned} \quad (10)$$

where

$$\begin{aligned} \psi &= \int_0^d [\phi(z)]_{\frac{R}{R_0}=1} dz + \int_d^{d+\frac{L_0}{2}} [\phi(z)]_{\frac{R}{R_0} \text{ from (1)}} dz + \\ &\int_{d+\frac{L_0}{2}}^{d+L_0} [\phi(z)]_{\frac{R}{R_0} \text{ from (2)}} dz + \int_{d+L_0}^L [\phi(z)]_{\frac{R}{R_0}=1} dz \end{aligned}$$

The second and third integrals being non-integrable in analytic form, are evaluated numerically by developing computer codes.

The expressions for impedance (λ), the wall shear stress in the stenotic region (τ_w) and the shear stress at the stenosis throat (τ_s) in their non dimensional form are given by

$$\begin{aligned} \lambda &= \frac{1 - \frac{L_0}{L}}{\eta} + \\ &\frac{1}{L} \int_d^{d+\frac{L_0}{2}} \frac{dz}{(a^2 - \varepsilon^2)[a^2 + \varepsilon^2 - (a^2 - \varepsilon^2) / \log(\frac{a}{\varepsilon})]} + \\ &\frac{L_0}{2\pi L} \int_0^\pi \frac{d\alpha}{(\theta^2 - \varepsilon^2)[\theta^2 + \varepsilon^2 - (\theta^2 - \varepsilon^2) / \log(\frac{\theta}{\varepsilon})]} \end{aligned} \quad (11)$$

$$\tau_w = \frac{\frac{R}{R_0}}{\left\{ \left(\frac{R}{R_0}\right)^2 - \varepsilon^2 \right\} \left[\left(\frac{R}{R_0}\right)^2 + \varepsilon^2 - \left\{ \left(\frac{R}{R_0}\right)^2 - \varepsilon^2 \right\} / \log\left(\frac{R}{R_0}\right) / \varepsilon \right]} \quad (12)$$

$$\tau_s = \frac{b}{\{b^2 - \varepsilon^2\} [b^2 + \varepsilon^2 - \{b^2 - \varepsilon^2\} / \log(\frac{b}{\varepsilon})]} \quad (13)$$

where,

$$a \approx a(z) = 1 - 2\left(\frac{\delta}{R_0}\right)(Z - d) / L_0, \quad b = 1 - \frac{\delta}{2R_0},$$

$$c = \frac{\delta}{2R_0}, \quad \theta \approx \theta(\alpha) = b + c \cos \alpha,$$

$$\alpha = \pi - \left(\frac{2\pi}{L_0}\right)(z - d - \frac{L_0}{2})$$

$$\eta = (1 - \varepsilon^2) \{1 + \varepsilon^2 + (1 - \varepsilon^2) / \log \varepsilon\}$$

$$\lambda = \frac{\bar{\lambda}}{\lambda_0}, \quad \tau_w = \frac{\bar{\tau}_w}{\tau_0}, \quad \tau_s = \frac{\bar{\tau}_s}{\tau_0}$$

$$\bar{\lambda} = \frac{\Delta p}{Q}, \quad \bar{\tau}_w = -\frac{R}{2} \left(\frac{dp}{dz}\right), \quad \bar{\tau}_s = \left[-\frac{R}{2} \left(\frac{dp}{dz}\right)\right]_{\frac{R}{R_0}=b},$$

$$\lambda_0 = \frac{8\mu L}{\pi R_0^4}, \quad \tau_0 = \frac{4\mu Q}{\pi R_0^3}$$

λ_0 and τ_0 are the flow resistance and shear stress for a normal artery (without stenosis) in Newtonian fluid.

III. NUMERICAL RESULTS & DISCUSSION

To discuss the results of the study quantitatively, algorithms and computer codes are developed for the numerical evaluations of the analytical results obtained in equations (11)-(13). The various parameter values are taken as (Young [15]):

L_0 (cm) = 0.5; L (cm) = 1; ε (non-dimensional catheter radius) = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 ; $\frac{\delta}{R_0}$ (non-dimensional stenosis

height) = 0, 0.05, 0.10, 0.15. In the present study we have also observed the flow of blood in uncatheterized and normal (without stenosis) artery for parameter values $\varepsilon = 0$ and $\frac{\delta}{R_0} = 0$ respectively.

The impedance, λ increases with the catheter size, ε for any given stenosis height $\frac{\delta}{R_0}$ and also increases with stenosis

height, $\frac{\delta}{R_0}$ for any given catheter size, ε . It is further observed

that for any given stenosis height, a significant increase in the magnitude of impedance, λ occurs for any small increase in the catheter size, ε . (Fig2.)

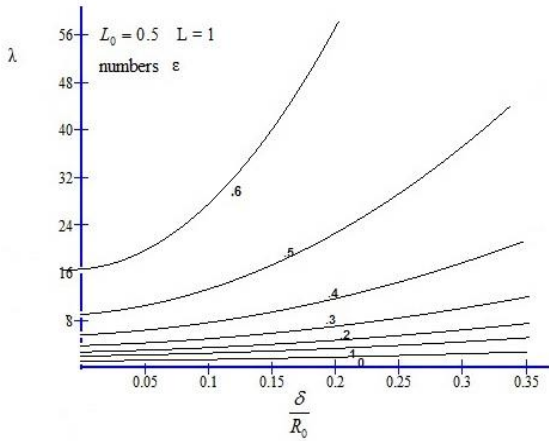


Fig. 2 Variation of impedance, λ with δ/R_0 for different ϵ .

The flow resistance, λ steeply increases with the catheter size, ϵ upto 0.3 but then increases rapidly with increasing catheter size ϵ and attains a very high asymptotic magnitude (Fig.3).

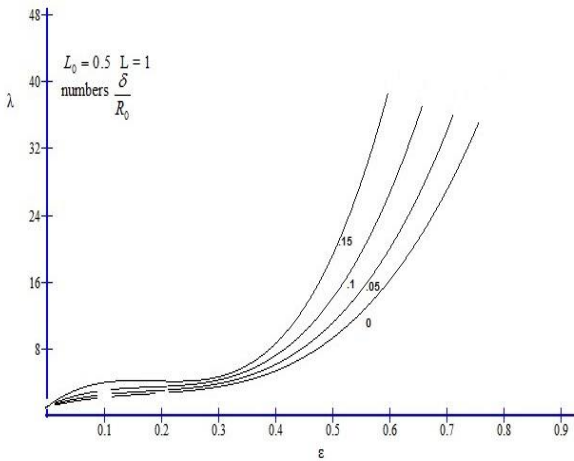


Fig.3 Variation of impedance, λ with ϵ for different δ/R_0 .

The wall shear stress in the stenotic region τ_w increases with the axial distance z / L_0 and attains reasonably higher magnitude at the end point of the constriction profile (i.e. at $z / L_0 = 1$) than its approached value at $z = 0$ (Fig.4).

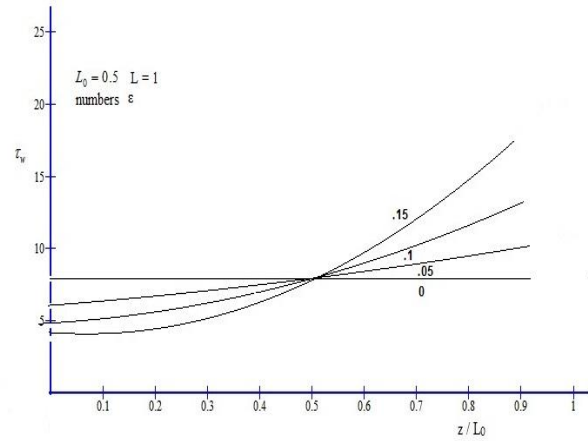


Fig.4. Wall shear stress distribution in the stenotic region for different δ/R_0 .

The flow characteristics, τ_w increases with the catheter size, ϵ at any axial distance in the stenotic region. For small catheter size upto 0.3 the variations in the magnitude of τ_w seems to be steeply, however, for larger values of ϵ , the magnitude of τ_w varies rapidly (Fig.5).

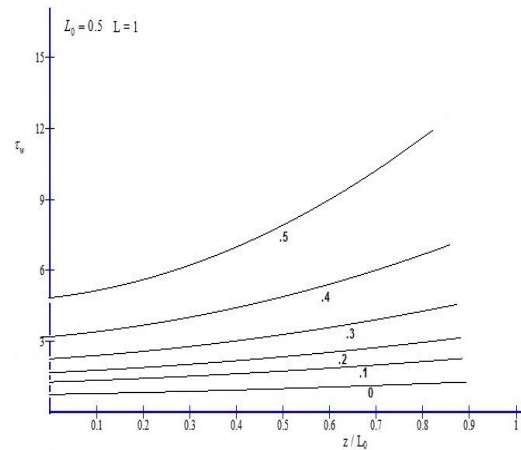


Fig.5 Wall shear stress distribution in the stenotic region for different ϵ

The shear stress at the stenosis throat, τ_s (at $z / L_0 = 1/2$) possess similar characteristics to that of the flow resistance, λ with respect to any parameter. However, the magnitude of the shear stress, τ_s is noted to be reasonably higher than the corresponding magnitude of the impedance, λ for any given set of parameters (Fig.6 and Fig.7).

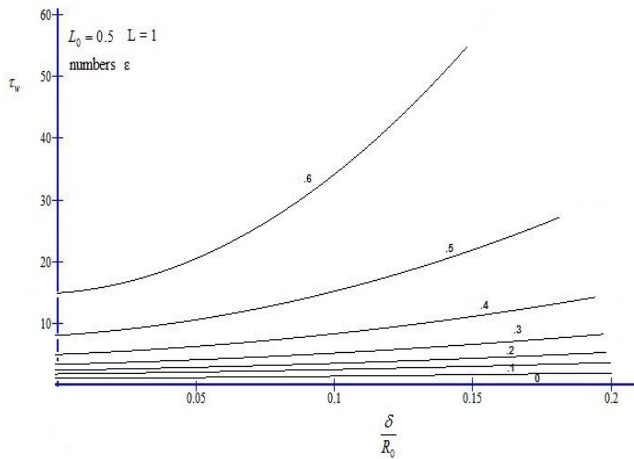


Fig. 6 Variation of shear stress at stenosis throat, τ_s with δ/R_0 for different ϵ .

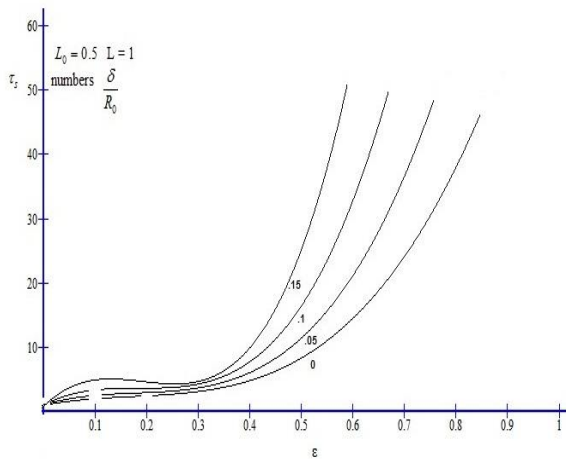


Fig.7 Variation of shear stress at stenosis throat, τ_s with ϵ for different δ/R_0 .

IV. CONCLUSION

The motivation behind this theoretical analysis is to estimate the increased impedance and shear stress in a stenosed artery during artery catheterization. It is observed that under the conditions of the present study, computational results are more or less in the expected range. It is further noted that the catheter radius (ϵ) and stenosis height ($\frac{\delta}{R_0}$) are the strong parameters influencing the flow qualitatively and quantitatively. So the size of the catheter should be cautiously chosen keeping in view of the stenosis height during medical treatment. The asymptotic nature of the impedance at $\epsilon = 0.3$ for different values of the stenosis height is an important observation of the investigation. Thus the model developed in this paper will throw light on the clinical treatment of the

obstruction of fluid movement due to formation of multiple stenoses in the arterial system and may reduce some of the major complications for the development of ischemia and coronary thrombosis.

REFERENCES

- [1] Anderson, H.V., Roubin, G.S., Leimgruber, P.P., Cox, W.R., Douglas, Jr. J.S., King III, S.B. and Gruentzig, A.R., " Measurement of transstenotic pressure gradient during percutaneous transluminal coronary angioplasty", *Circulation*, vol. 73, pp.1223-1230, 1986.
- [2] Back, L.H, " Estimated mean flow resistance increases during coronary artery catheterization", *J. Biomech*, vol. 27, pp.169-175, 1994.
- [3] Back, L.H., Kwack, E.Y. and Back, M.R., " Flow rate-pressure drop relation in coronary angioplasty: catheter obstruction effect", *J. Biomed. Engng*, pp. 118, 83-89, 1996.
- [4] Dash, R.K., Jayaraman, G. and Mehta, K.N. , " Flow in a catheterized curved artery with stenosis", *J. Biomech* ,pp.49-61, 1999.
- [5] Gunj, P., Abben, R., Friedman, P.L., Granic, J.D., Barry, W.H. and Levin, D.C., "Usefulness of transstenotic coronary pressure gradient measurements during diagnostic catheterization", *Am. J. Cardiology* ,vol 55,pp. 910-914,1985.
- [6] Kanai, H., Lizuka, M. and Sakamotos, K., "One of the problem in the measurement of blood pressure by catheterization: wave reflection at the tip of catheter", *Med. Biol. Engng* ,vol.28,pp. 483-496, 1970.
- [7] Leimgruber, P.P., Roubin, G.S., Anderson, H.V., Bredlau, C.E., Whitworth, H. B., Douglas Jr., J.S., King III, S. B. and Gruentzig, A. R. "Influence of intimal dissection on restenosis after successful coronary angioplasty", *Circulation*,vol. 72, pp. 530-535,1985.
- [8] Sankar, D.S. and Hemlatha, K. , "Pulsatile flow of Herschel-Bulkley fluid through catheterized arteries- a mathematical model", *Appl. Math Modelling* ,vol. 31,pp. 1497-1517, 2007.
- [9] Sarkar, A. and Jayaraman, G., " Correction to flow rate-pressure drop in coronary angioplasty: steady streaming effect", *J. Biomech*,vol. 31, 781-791,1998.
- [10] Srivastava, V.P., "Arterial blood flow through a non-symmetrical stenosis with applications", *Jpn. J. Appl. Phys*, vol. . 34,pp- 6539-6545, 1995.
- [11] Srivastava, V. P. and Srivastava, Rashmi., " Particulate suspension blood flow through narrow catheterized artery", *Comput. Math. Applc* , vol .58, pp. 227-234,2009.
- [12] Srivastava, V. P. and Rastogi, R., "Effects of hematocrit on impedance and shear stress during stenosed artery catheterization", *Applications and Applied Mathematics*, vol. 4, pp. 98-113, 2009.
- [13] Srivastava, V. P. and Rastogi, R., " Blood flow through stenosed catheterized artery: effects of hematocrit and stenosis shape", *Comput. Math. Applc* ,vol.59, pp.1377-1385, 2010.
- [14] Wilson, R. F., Johnson, M. R., Marcus, M.L., Aylward, P. E. G., Skorton, D., Collins, S. and White, C. W., "The effect of coronary angioplasty on coronary flow reserve", *Circulation*, vol. 77, pp. 873-885, 1988.
- [15] Young, D. F., "Effects of a time-dependent stenosis on flow through a tube", *J. Eng. Ind.*, vol. 90, pp. 248-254, 1968.
- [16] Basu Mallik, B. and Nanda, S.P., "A Mathematical Analysis Of Blood Flow Through Stenosed Arteries: A Non-Newtonian Model", *IEM International Journal of Management & Technology(IEMIJMT)*, Vol-1(2),pp. 41-45,2012.
- [17] Nanda, S.P. and Bose, R.K., "Mathematical Analysis on Blood Flow through a Flexible Stenosed Artery", *IJCME*, vol. 2(1), pp. 17-30,2012.
- [18] Nanda, S.P. and Bose, R.K., "Blood Flow Through A Flexible Artery In Presence Of Stenosis- A Mathematical Study", *JMCMS*, vol. 6(2), pp 859-874,2012.

A Computer Vision Framework for Automated Shape Retrieval

Sourav Saha¹, Sahibjot Kaur¹, Jayanta Basak¹, Priya Ranjan Sinha Mahapatra²

¹Institute of Engineering & Management, Dept. of Computer Science & Engg., Kolkata, India.
{souravsaha1977, sahibjotkaur92, lettertojayanta}@gmail.com

²University of Kalyani, Dept. Computer Sc. & Engg., W.B., India
priya_cskly@yahoo.co.in

ABSTRACT—With the increasing number of images generated every day, textual annotation of images for image mining becomes impractical and inefficient. Thus, computer vision based image retrieval has received considerable interest in recent years. One of the fundamental characteristics of any image representation of an object is its shape which plays a vital role to recognize the object at primitive level. Keeping this view as the primary motivational focus, we propose a shape descriptive framework using a multi-level tree structured representation called Hierarchical Convex Polygonal Decomposition (HCPD). Such a framework explores different degrees of convexity of an object's contour-segments in the course of its construction. The convex and non-convex segments of an object's contour are discovered at every level of the HCPD-tree generation by repetitive convex-polygonal approximation of contour segments. We have also presented a novel shape-string-encoding scheme for representing the HCPD-tree which allows us to use the popular concept of string-edit distance to compute shape similarity score between two objects. The proposed framework when deployed for similar shape retrieval task demonstrates reasonably good performance in comparison with other popular shape-retrieval algorithms.

Index Terms—Convex Polygon; Content Based Shape Retrieval; Shape Representation

I. INTRODUCTION

Due to the recent developments in digital imaging technologies, an increasing number of images are generated every day which propels today's tech-savvy people to demand an automated system for retrieving images of their interest from large data-pool. Searching images using their textual annotations is a subjective process and is also not practical for large databases. In recent years, shape is considered as one of the most promising criteria for automated identification of an object. In general, shape is a very common perceived concept which is widely understood yet difficult to

define formally. The human perception of shape is a high-level intuitive concept whereas mathematical definitions tend to describe shape with low-level features. Many researchers like Marshall [1][28] tried to define shape as a function of position and direction of simply connected curves within the two-dimensional field. Regardless of how shape gets defined, it has always been important visual information attracting attention of researchers over the past few decades specifically dealing with the problems of pattern recognition and computer vision. With the ever-growing demand of automated emulation of human vision system, research interest among today's scientists in computer vision domain has been gradually driven towards obtaining maturity in automated shape analysis from both theoretical and practical point of view. A user survey in [1] indicated that 71% of the users were interested in retrieval by shape. The main motivation of this research work is to focus on computationally extracting and exploring geometric properties of shape for content-based shape retrieval.

In general, the shape of a binary image object is understood as being formed by the boundary of the object. In this sense, the trivial representation of an object's shape is the sequence of points identified along the boundary of the original image. It is often the case that such representation is not useful to describe identifying characteristics of an object's shape under plausible deformations. But human visual system somehow is able to recognize an object even when it undergoes various deformations. Scientists argue that it is hard to emulate human vision ability computationally as the biological functionalities involved in visual perception is quite different in comparison with computational functionality of today's computers. Computers are efficient in terms of carrying out complex, repetitive mathematical or logical operations with great accuracy but perceptual task like object recognition for them still remains a hard challenge. In order to equip computers with such naïve recognition ability, efforts have been continuing to develop an effective mathematical framework for obtaining an accurate shape-descriptive computational model

formulating unique identity of an object [1]. A popular criterion for characterizing shape representation techniques involves their classification which uses boundary or interior structure of an object as identifying features. Unfortunately, most of the existing approaches based on either strategy have not extensively explored boundary or structural decomposition framework at ontological level applying computational geometric analysis approach. In this paper, we have made an effort to generate a multi-level tree structured framework termed as Hierarchical Convex Polygonal Decomposition (HCPD) tree based on computational geometric analysis. The hierarchy of the HCPD reflects the convexity based inclusion relationship among various segments of the object along the boundary. To facilitate shape-based matching, a new shape encoding strategy for HCPD-tree has been developed. In addition, an effective shape-code matching algorithm, based on Dynamic Programming strategy has been used to quantitatively measure the shape-similarity between two objects. The key objective of this work is to demonstrate efficiency of the proposed framework by deploying it for the task of retrieving similar shapes with reference to a query object.

Rest of the paper is organized as following. In the background section we have discussed about commonly used object recognition model along with previous notable researches on various shape representation frameworks. In section III, the proposed technique is detailed focussing on HCPD tree generation, HCPD-tree encoding, and shape similarity score evaluation strategy. The subsequent section to the detailed framework-description has aimed to produce a report on how efficiently our framework has performed on shape-retrieval task by substantiating results. The limitations of our framework and future scope of the work have also been discussed in concluding section.

II. BACKGROUND

Content-based image retrieval (CBIR) has emerged as a promising mean for retrieving images and browsing large images databases [1]. CBIR has always been a topic of intensive research in recent years. It is the process of retrieving images from a collection based on automatically extracted features. Shape of an object as compared to other features, like texture and colour, is much simpler and also effective as identifying feature in semantically characterizing the image of an object.

A. Basic Object Recognition Concepts

The object recognition problem can be defined as a labelling problem based on models of known objects. Formally, given an image containing one or more objects of interest and a set of labels corresponding to a set of known models, the system's task is to assign correct label to an object.

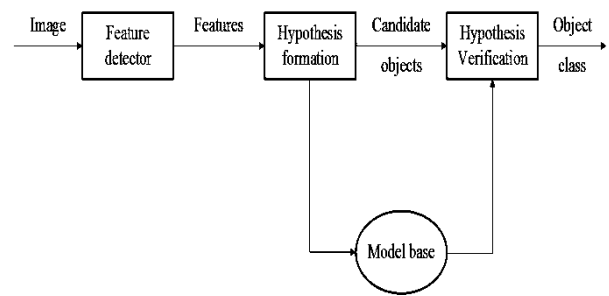


Figure 1 Object Recognition Framework

The central issues that should be considered in designing an object recognition system are discussed below.

- **Object or model representation:** For some objects, geometric descriptions formulated in terms of computational model are efficient. The model must process all relevant information without any redundancies.
- **Shape Descriptor:** In general, the shape of an object is understood as being formed by the set of points belonging to that object. Shape representation of an object very often depends on either boundary or interior structure. It plays a vital role in acting as unique identifying feature.
- **Feature-model matching:** In most object recognition tasks, there are many features and numerous objects. Effectiveness of features and efficiency of a matching technique must be considered during developing a matching approach.
- **Shape Matching:** Shape similarity refers to establishing criteria that estimates how much two shapes are similar (or different) to each other, including issues such as when a given shape **A** can be considered more similar to another shape **B** than to **C**. To facilitate shape-based matching, an effective encoding scheme needs to be developed. Besides encoding schemes, we also need to explore various distance metrics like *String based Edit Distance*, *Syntactic Deviation Measure*, *Graph based Edit Distance* to quantitatively measure shape similarity.
- **Hypotheses formation:** The hypothesis formation step is basically a heuristic to reduce the size of the search space.
- **Object verification:** The presence of each likely object can be verified by using their models.

B. Related Literature Review

In the past, contour and skeleton were usually used to analyse and represent the shape of objects. Due to the importance of shape information in image understanding, many shape comparison algorithms have been proposed [1]–[19], which can be roughly divided into two categories: contour-based and region-based. The contour-based methods are much more popular than the region-based ones in the past decade, and they can be further classified into two subcategories: global

and local. Contour-based is an important aspect of human visual perception. In global shape comparison, a shape is usually represented by a feature vector extracted from the whole contour, and shape comparison is conducted by comparing such representation vectors. A classic global shape representation is the curvature scale space (CSS) [3], which has been recommended by the MPEG-7 community as one of the standards. In CSS, the zero-crossings of the contour curvature function are located at different scales. These zero-crossings form a CCS image, and the maxima of such CCS image contours are used for shape matching. Another example of global method is the polygonal multi-resolution and elastic matching (PMEM) [9], in which three primitives of each contour segment are extracted at different scales. Then, the sum of absolute differences (SAD), improved by the elastic matching, and is used to measure the similarity between shapes. Another recent global method is the contour point's distribution histogram (CPDH) [19], which represents a shape by the spatial distribution of contour points in the polar coordinate system and compares such distributions using the Earth Mover's Distance (EMD) [19].

In local shape comparison, a shape is typically represented by a set of local descriptors such that each descriptor captures only local shape information. Shape similarities are then derived by two layers of comparison: the low-level similarities between local descriptors and the high-level comparison on top of such similarities. A classic example in this category is the shape context (SC) [1], which uses 2-D histograms to capture the spatial context around each landmark point, and compares two shapes by matching two sets of such histograms. SC becomes very popular because of its powerful descriptive ability and is extended in various ways. One such extension is the inner distance shape contexts (IDSC) [5], which replaces the Euclidean distance with the inner distance to achieve robustness against articulation. In addition, in [5] dynamic programming (DP) is used to utilize the continuity constraint on contour points. In [10], hierarchical Procrustes matching (HPM) is proposed to capture shape information across different hierarchies. The shape tree (ST) in [2] also captures hierarchical geometric propensities of a shape, and it utilizes a tree matching method for shape comparison. In [16], a descriptor named contour flexibility (CF) is proposed which describes each contour point by its deformable potential. CF also uses DP for shape matching. Recently, a novel shape descriptor using height functions (HF) is proposed in [23] and DP is again used for matching such descriptors.

Local shape descriptors usually outperform global ones in terms of shape matching accuracy. However, such superiority is typically at a cost of reduced efficiency in terms of computation time. By contrast, global methods achieve better run time efficiency thanks to the simplicity in their shape representation and associated shape matching algorithms. Modern CBIR systems often deal with large image datasets, and are therefore in favor

of fast query response. Consequently, both effectiveness and efficiency are desired by such systems [20]. In fact, MPEG-7 has set several criteria [21] to evaluate a shape descriptor, including high retrieval accuracy, feature compactness and low computational complexity. Notice that in recent years, the multi-scale framework [2], [9], [10] introduced in many methods has been proved to be able to enhance the performance of the original single-scale method. It is worth to mention that most of the existing methods rely on scale normalization to achieve scale invariance and circular shifting to achieve the rotation insensitivity, both of which request extra computational time. Instead of designing new shape descriptors and/or shape comparison methods, some recent studies improve shape retrieval by exploring retrieval techniques, such as in [24-26]. These methods can be viewed as post-shape matching enhancement and can be divided into three categories: context-based, knowledge-based and fusion-based. More details about these methods can be found in [20].

Motivated by the above non-satisfying research-explorations attempted towards recognizing shapes, in this paper we propose a novel computational shape representation framework that has experimentally demonstrated reasonable efficiency for shape retrieval task. The key idea in this proposed framework is to exploit different degrees of convex properties along the contour of the object using multi-level tree structured representation called Hierarchical Convex Polygonal Decomposition (HCPD) [40]. We have also presented a novel HCPD tree encoding scheme and a dynamic programming strategy based shape-code matching algorithm in order to measure similarity between two shapes.

III. PROPOSED FRAMEWORK

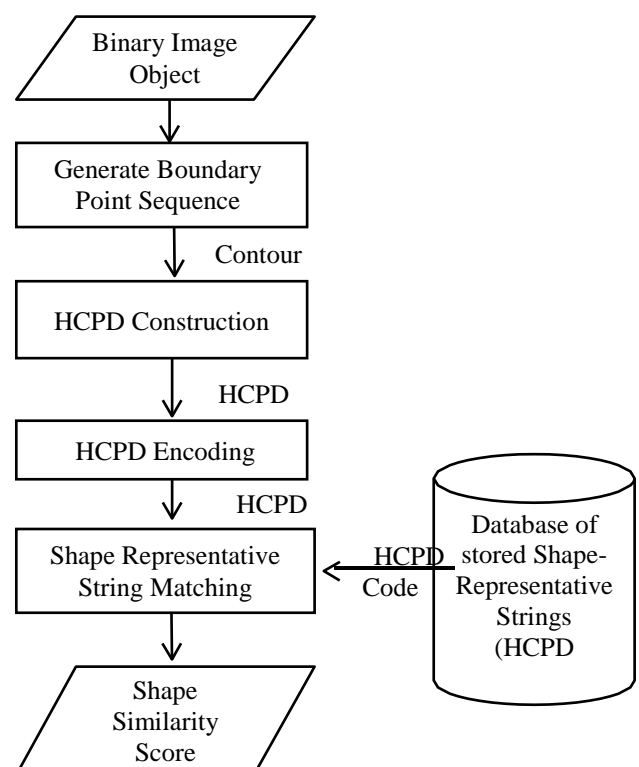


Figure 2 Overview of proposed framework

We model shape of image objects using a tree structured representation called Hierarchical Convex Polygonal Decomposition (HCPD) which utilizes convexity of the boundary segments to discover dominant contour points at multiple levels of the tree. An overview of the proposed framework has been presented in Figure 2 to understand the sequence of operations needed to perform similar shape retrieval task. In this section, we have illustrated how our HCPD framework has been constructed in the course of discovering dominant contour points at multiple levels, how the HCPD is encoded and how we have deployed a shape-code matching scheme inspired by dynamic programming strategy to measure shape-similarity quantitatively between two objects.

A. Hierarchical Convex Polygonal Decomposition (HCPD)

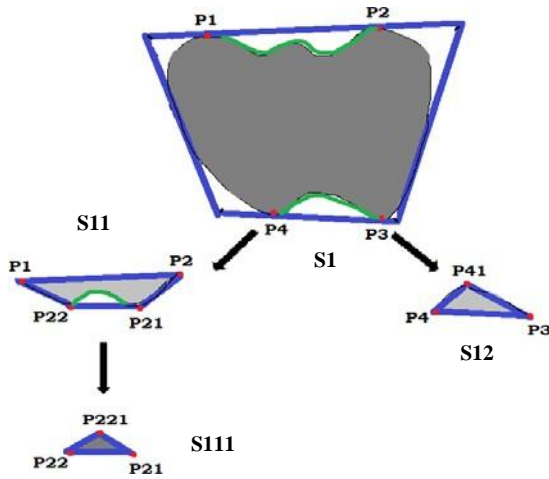


Figure 3 HCPD of Apple-1

This section illustrates our HCPD framework construction strategy with reference to a binary image object as shown in Figure 3. Our proposed method hierarchically decomposes the boundary into several connected convex and non-convex segments by generating a circumscribing convex polygon of the input shape. The hierarchical decomposing discovers various dominant points on the contour as shown in figure 3. The shaded regions represent decomposed sub-shapes which are further decomposed at next lower levels. At first level, the circumscribing convex polygon results in two non-convex segments namely S_{11} : $\langle P_1, P_2 \rangle$, and S_{12} : $\langle P_3, P_4 \rangle$. At subsequent level, S_{11} is further decomposed based on its circumscribing convex polygon and results in identifying a non-convex segment S_{111} : $\langle P_{21}, P_{22} \rangle$. Further S_{111} is circumscribed by a convex polygon and it is found that no more non-convex segment is present in S_{111} leading to the termination of decomposition at its representative node. Notably, at every node starting from the root repetitively non-convex parts of shape-contour are extracted by circumscribing convex-polygon around boundary-segment at a parent node and subsequently

each non-convex segment is treated in similar manner at lower tree-levels as long as convex-polygonal approximation results in significant non-convex regions. It implies that the decomposition operation continues down the tree levels until no more non-convex parts are generated out of a segment. Such hierarchical decomposition discovers various dominant points along the contour of the object as shown in figure 4 with reference to HCPD of object--Apple-1. The formal algorithm for HCPD generation is presented below.

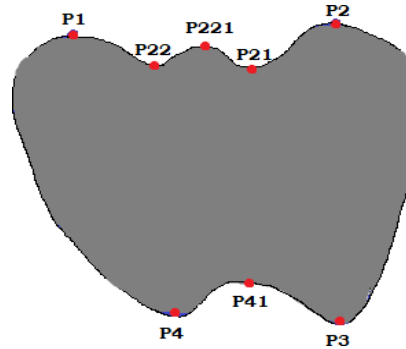


Figure 4 Dominant Points of Apple-1

Algorithm:

```

Algorithm: DecomposeShape(Shape S){
Input: Binary Object Shape S
Output: Hierarchical Tree of convex-
polygons (HCPD)
1 L ← FindBoundaryPoint(S);
2 FindConvexPolygon(L);
3 Determine & save shape-code of the
polygon for respective tree-node.
4 Determine non-convex regions;
5 for(each non-convex region: Si) do{
6 DecomposeShape(Si) }
7 }

```

B. Boundary Point Tracing

Boundary point tracing is one of important pre-processing techniques performed on digital images in order to extract information about their general shape. In our proposed work, we have adopted Moore's neighbourhood [34] contour tracing strategy to extract boundary points in a specific order (counter-clockwise). During HCPD generation phase, at every node-level decomposition, we decide which of them form vertices of a convex-polygonal covering corresponding to the boundary segment represented by the node. Thus boundary points sequence is important before we can move forward to HCPD generation phase. In order to discover a new boundary point, the exploration requires examining eight-neighbourhood of an already found boundary point, P, namely locations P1, P2, P3, P4, P5, P6, P7 and P8 as shown in Fig. 2 in counter-clockwise direction. This repetitive 8-neighbourhood exploration of every newly discovered boundary pixel results in a sequence of points along the actual contour of the object. This recursive process starts with any

extreme corner along the boundary and terminates when the starting corner pixel is visited for a second time.

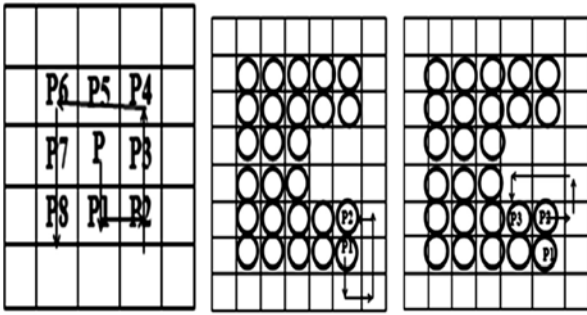


Figure 5 Contour Tracing

In principle, the general idea of Moore's neighbourhood tracing [4] as shown in figure 5 is that every time the counter clockwise scanning hits an object pixel-point, P, we trace back to the neighbour pixel (P4) from where the object pixel-point P's location was entered. We go around pixel to start exploration of its 8-neighborhood in counter-clockwise fashion starting from P4 until a new boundary pixel-point is encountered. The algorithm based on the above discussed principle is presented below.

Algorithm: BoundaryPointSequenceGeneration()

Input: Binary Object Shape S

Output: An ordered set of Border Pixels

```

1 Seed ← RightBottomMost Pixel
2 Order 8-neighborhood pixels of the
seed starting from direct bottom-
neighbor in counter-clockwise
direction as P1 to P8.
3 do {
4   for P1 to P8 in 8-neighborhood do{
5     if (Pi ∈ BorderPixel & not yet part
of the boundary sequence){
6       Add Pi to the Boundary Sequence.
7       Set Pi as next seed.
8       Order 8-neighborhood of the new
seed counter-clockwise starting from
Pi-1
9       break from the current loop. }
10    } // end for loop
11} while (seed ≠ RightBottomMost Pixel)

```

C. Approximating Convex Polygonal Cover of a Boundary Segment

Our convex polygonal covering of a segment is primarily inspired by Graham's scan algorithm [35] which is popularly used for determining convex hull of an object. Given a segment, the idea is to detect three successive boundary points in counter-clockwise direction to form an ordered triplet-points $\langle p1, p2, p3 \rangle$ as a candidate and attempt to single out a point from the ordered triplet that needs either to be discarded or picked up for the convex-polygon. The point selection

or elimination is based on whether these three points make a left turn or right at position $p2$. The equation as stated below helps us in deciding the turn-direction as it yields non-negative value for left turn but a right turn produces negative value. In case $p1, p2, p3$ forms a left turn, we may consider boundary point $p1$ as convex-polygonal point and the remaining points $p2, p3$ are set as first and second element for the next candidate triplet-points. On the other hand, a right turn implies that $p2$ cannot be on convex-polygon of the object and in that case $p1, p3$ are set as first and second element for our next candidate triplet-points. Subsequently, another boundary point ($P4$) in counter-clockwise direction is detected and added as the last point of our next candidate triplet-points. Once again we repeat the same procedure to find out the point from the ordered triplet that needs either to be discarded or picked up based on the above mentioned convexity analysis of point $p2$. Our approach as presented below deviates from Gram's scan algorithm with regards to that fact that it considers only boundary points in counter-clockwise direction instead of every object points and discards inclusion of a point if the triangular area generated by the triplet $\langle p1, p2, p3 \rangle$ becomes insignificantly small. Thus the modified algorithm basically results in an approximated convex-polygon covering the boundary of the input segment. Each of these convex polygons is encoded in a special manner as detailed in next section to form a binary pattern and a depth-first-search (DFS) of the HCPD tree generates a shape-code for the input object.

$$\text{Area}(p1, p2, p3) = ((p2.x - p1.x) * (p3.y - p1.y) - (p3.x - p1.x) * (p2.y - p1.y)) [1]$$

Algorithm:

```

FindApproximateConvexPolygon() {
Input: Boundary point-list: L
Output: Convex Polygonal Cover
1 p1 ← first point in Boundary
point-list: L;
2 p2 ← second point in Boundary
point-list: L;
3 p3 ← third point in Boundary
point-list: L;
4 while (p3 ≠ first point of Boundary
point-list: L) {
5   if ( p1, p2, p3 form a left
turn) {
6     if (triangular area generated
with p1, p2, p3
> AreaThreshold) {
7   add p1 to convex polygon point
list.}
8     p1 ← p2;
9     p2 ← p3;
10    p3 ← next point in Boundary
point-list: L; }
11  else {
12    p2 ← p3;
13    p3 ← next point in Boundary
point-list: L; }

```

D. Hierarchical Convex Polygonal Decomposition Tree (HCPD) Encoding

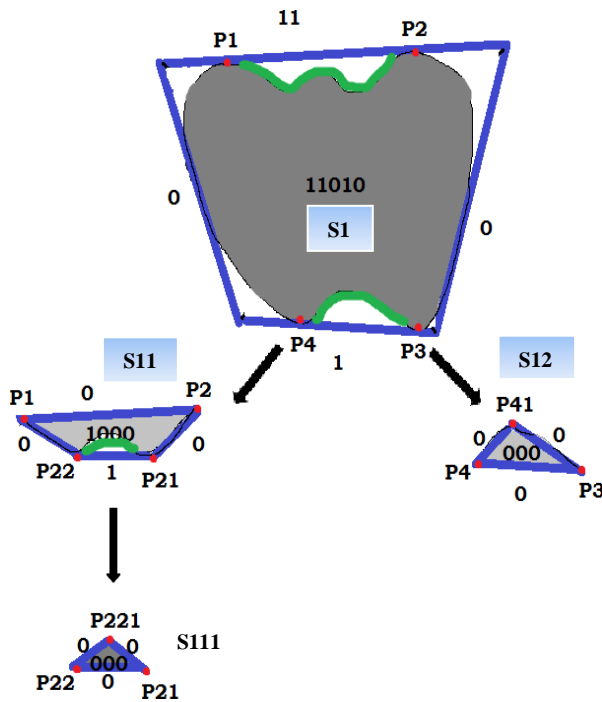


Figure 6 Encoding of HCPD (Apple-1)

The convex polygonal cover generated at each node during HCPD tree construction is encoded using binary symbols based on convexity of contour segments and the concatenation of these binary codes in the course of depth-first-search (DFS) traversal of the HCPD tree leads to a special shape-code for the input object. Such an encoding scheme facilitates utilization of popular string matching algorithms. Below, the encoding policy for a shape-attributed node of HCPD tree is described.

Encoding Policy for HCPD:

At each HCPD node, Convex Polygonal Covering method partitions the input contour into several segments as shown in figure 6. Every such segment is encoded based on following principle.

- If it is found that 'n' number of dominant points is discovered for an encoding candidate segment of the current node at its immediate next level, then the candidate segment is encoded as n number of repeated ones.
- If it is found that no dominant points are discovered for a candidate segment of the current node at its immediate next level, then the candidate segment is encoded as zero.

The above mentioned encoding scheme is further illustrated with reference to the figure 6. As we look into the HCPD, it is observed that two new dominant points namely P21 and P22 are discovered corresponding to the segment <P1, P2> at its immediate

next level. Therefore, the segment <P1, P2> must be encoded as 11.

- After encoding each partitioned segment at a node based on the discovery of dominant points at next level, a largest binary number is formed by making a clockwise trip along the contour-segments. The largest possible binary sequence in terms of its value is treated as the special shape code for the HCPD node.

For example, the root node of the HCPD as shown in figure 6 is encoded as 11010 considering <P1, P2>, <P2, P3>, <P3, P4>, <P4, P1> as a most likely partitioned segment-sequence in order to form largest binary number by making a clockwise trip along the contour.

- Once each node gets encoded following above mentioned binary encoding scheme, we traverse HCPD-tree using DFS exploration to form a special string considering all node-representative shape-codes. The special string preserves parent-children relationship among the nodes of HCPD-tree so that the tree can be constructed from the shape-code. The following example explains the strategy with reference to figure 6.

For the HCPD-tree as shown in figure 6, the special shape-encoded-string representing the tree is "[11010 [1000 [000]], [000]]" i.e. an implicit DFS traversal string which can be represented in terms of shape-segments as [S1 [S11 [S111]], [S12]]. The algorithm for generating the expression is presented below.

Algorithm: getDFSString(HCPD Tree-Node)

Input: HCPD tree-node with shape code
Output: dfsString: Shape Code for HCPD Tree

```

1 String dfsString = "["
+ShapeCode(current node);
2 String str = "";
3 for (each child of current Node) {
4 str += getDFSString(child);
5 }
6 if(current HCPD node has children){
7     dfsString += "[ "+ str + " ]";
8 dfsString += " ]";
9 returndfsString;

```

E. Shape Similarity Matching

We have developed an effective shape-code matching algorithm, based on string-edit distance [33] to measure shape-similarity quantitatively between two objects. Edit distance is a way of quantifying how dissimilar two strings are to one another by counting the minimum number of operations required to transform one string into the other. Several definitions of edit distance exist, using different sets of string operations. One of the most common variants is called Levenshtein distance [33],

named after the Soviet Russian computer scientist Vladimir Levenshtein. We have applied similar metric for computing shape-dissimilarity score with some modification to suit our shape representative HCPD-Tree comparison. Below, we have discussed some of the basic concepts of Levenshtein edit distance aligning the idea with our shape-string-context.

Formal Definitions:

Given two strings *a* and *b* on an alphabet $\Sigma : \{0, 1\}$, the edit distance *d* (*a*, *b*) is the minimum-weight series of edit operations that transforms *a* into *b*.

Insertion: If *a* = *uv*, then inserting the symbol *x* produces *uxv*.

Deletion: Deletion of a single symbol changes *uxv* to *uv*

Substitution/Inversion: Substitution of a single symbol *x* for a symbol *y* ≠ *x* changes *uxv* to *uyv* (*x*→*y*). It can be considered as a sequence of deletion followed by insertion. In case of binary symbol, substitution can also be interpreted as inversion operation.

In Levenshtein's original definition, each of these insertion or deletion operation has unit cost. So the cost for substitution is basically two except the fact that substitution of a symbol by itself has zero cost. Levenshtein's edit distance is equal to the minimum number of operations required to transform *a* to *b*. Based on Levenshtein edit distance, the shape dissimilarity measure between shape codes "11010" and "1011" is 3 as we can obtain shape code "1011" by deleting first symbol and inverting last symbol of shape code "11010". Mathematically, the recurrence relation for computing shape distance between two strings *m*, *n* based on Levenshtein's strategy is given by:

$$d[m, n] = \begin{cases} \max(d[m, n], \min(d[m, n-1], d[m-1, n]) + 1) & \text{if } s[m] = t[n] \\ \min(d[m, n-1], d[m-1, n]) + 1 & \text{if } s[m] \neq t[n] \end{cases}$$

The algorithm for evaluating shape distance is presented below. Note that, the distance matrix *d* has (m+1)*(n+1) value and *d*[*i*,*j*] for all *i* and *j* holds the shape distance between the first *i* symbols of *s* and the first *j* symbols of *t*.

Algorithm:

```
ShapeDistance(char s[1..m], char t[1..n]):
1 set each element in d[0..m, 0..n] to zero
2 for i <- 1 to m do
3 d[i, 0] <- i
4 for j <- 1 to n do
5 d[0, j] <- j
6 for j <- 1 to n do
7     for i <- 1 to m do
8         if s[i] = t[j] then
9 d[i, j] <- d[i-1, j-1]
```

```
10     else
d[i, j] <-
    minimum(d[i-1, j] + 1,
            d[i, j-1] + 1, d[i-1, j-1]
            + 1);
11 return d[m, n]
```

Illustration: Here we illustrate how edit distance is used to estimate similarity between two shapes. Figure 7 shows HCPD-tree of a shape similar to Figure 3 whereas Figure 8 presents HCPD-tree of another shape distinctively different from both Figure 3 and Figure 7. Let assume that *EditDist*(HCPD_A, HCPD_B) denotes shape dissimilarity score between shape-A and shape-B as discussed previously and also assume that *L_{max}* refers to the number of levels of one of the HCPD-Tree having maximum depth i.e. *L_{max}* = max{number of levels of HCPD_A, number of levels of HCPD_B}. Shape distance or dissimilarity denoted as *EditDist*(HCPD_A, HCPD_B) is calculated based on cumulative edit distances at every tree-level as explained below.

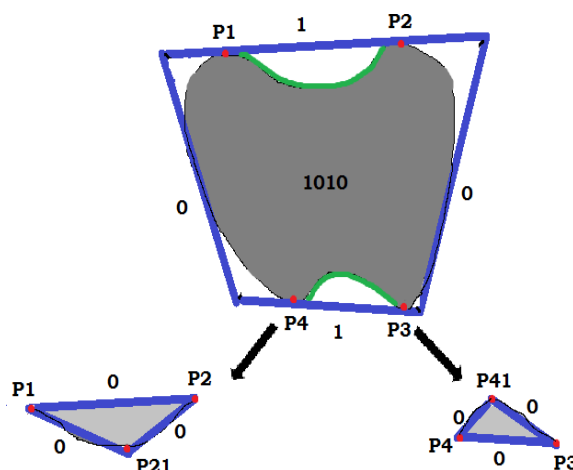


Figure 7 HCPD of Apple-2

Shape Dissimilarity Score Between Apple-1 and Apple-2: *L_{max}* = 3

$$\text{EditDist}(\text{HCPD_Figure 3}, \text{HCPD_Figure 7}) := \text{EditDist}(11010, 1010) * L_{max} + \text{EditDist}(1000000, 000000) * (L_{max} - 1) + \text{EditDist}(000, \text{null}) * (L_{max} - 2) = 3 + 2 + 3 = 8$$

Shape Dissimilarity Score Between Apple-1 and Papaya-1: *L_{max}* = 3

```

EditDist(HCPD_Figure 3, HCPD_Figure
8): EditDist(11010, 10100) * L_max
+ EditDist(1000000, 000000) * (L_max -
1)+ EditDist(000, ---) * (L_max - 2)
= 9 + 2 + 3 = 14

```

It is obvious from the computed *Shape Dissimilarity Scores* obtained by comparing the query shape's (Apple-1) HCPD-tree with HCPD-trees representing shapes Apple-2, and Papaya-1 respective of objects shapes: Apple-1, Apple-2, and Papaya-1, that the shape of Apple-1 is closer to the shape of Apple-2 in comparison with the shape of Papaya-1.

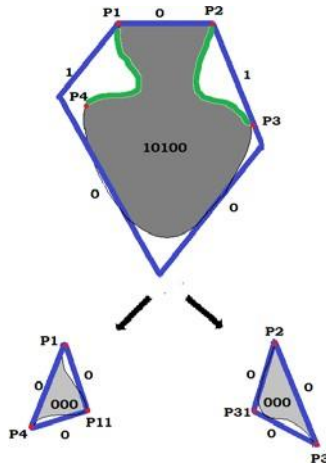


Figure 8 HCPD of Papaya-1

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

To evaluate the performance of the proposed shape retrieval system, experiments have been conducted based on the MPEG-7 test database [32]. The dataset consists of 1400 shapes grouped into 70 classes, each class containing 20 similar objects. Some of the shapes have experienced a number of transformations, such as scales, cuts and rotations and also the image resolution is not constant among them. The tables listed next present a set of sample images from MPEG-7 test database and their respective HPCDs along with shape code.

B. Performance Evaluation Metric

Evaluation of retrieval performance is a crucial problem in content-based image retrieval, mainly due to the subjectivity of the human similarity judgment. The evaluation of a shape retrieval system depends on the application domain. However, many different methods for measuring the performance of a system have been created and used by researchers. Perhaps the most widely used measure, for retrieval effectiveness; in the literature is the "Bull's eyes test" [32]. This frequently used test in shape retrieval enables the comparison of our approach against other performing shape retrieval techniques. Every shape in the dataset is compared to all other shapes, and the number of shapes retrieved from the same class among the top 40 retrieved similar shapes based on the applied algorithm is reported. Ideally the bull's eye retrieval rate for a query image is the ratio of the total number of retrieved shapes from the same class to the highest possible number which is 20 on MPEG-7. Thus the overall Bulls Eye Percentage (BEP) can be calculated taking average over individual BEP score for every query image from the data set.

















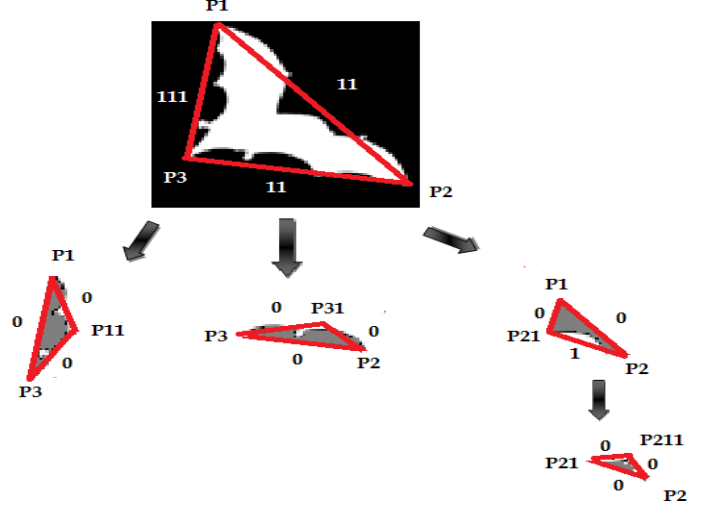
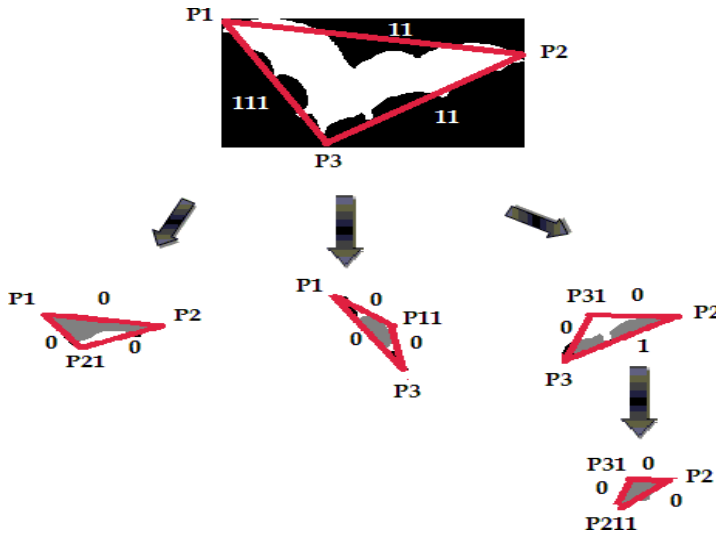
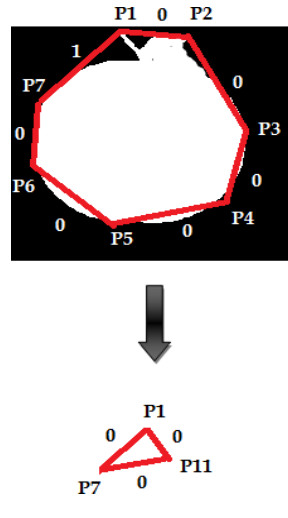
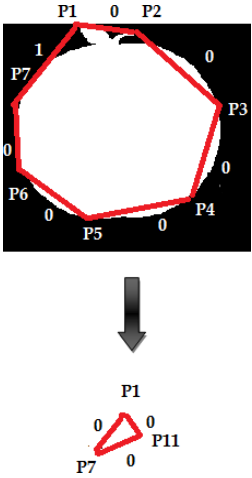
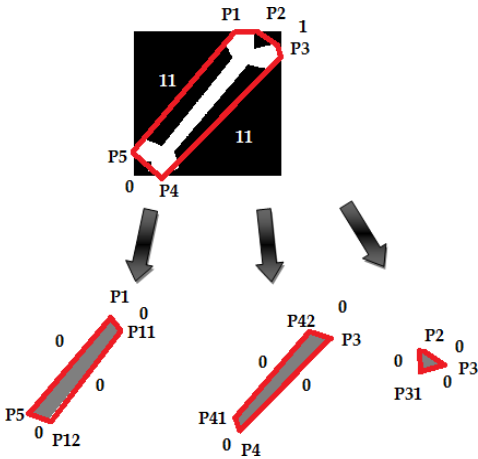
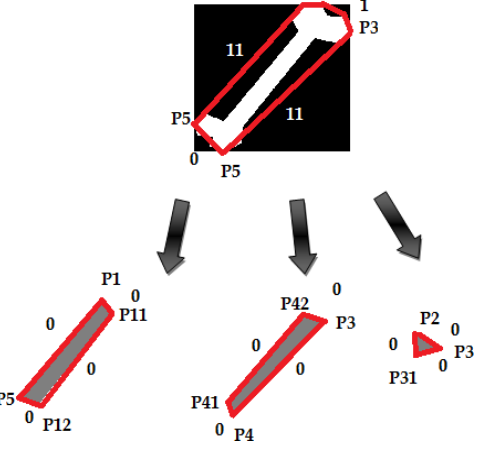
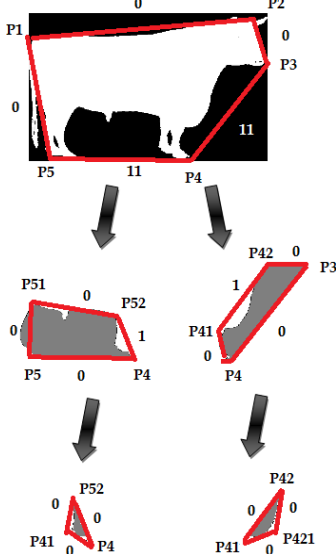
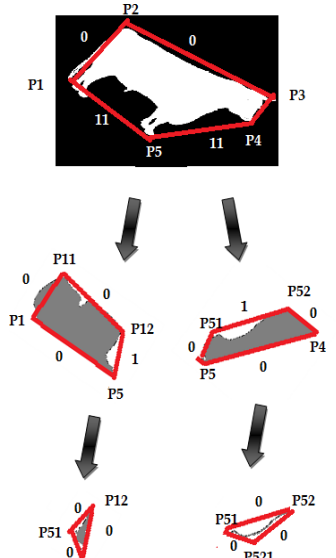
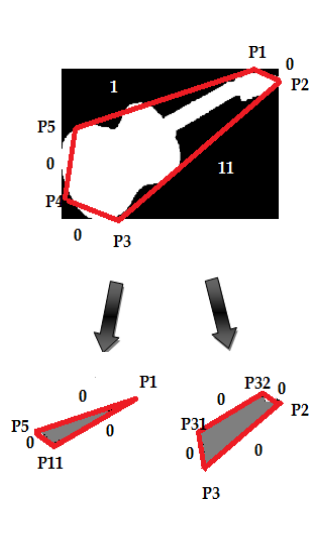
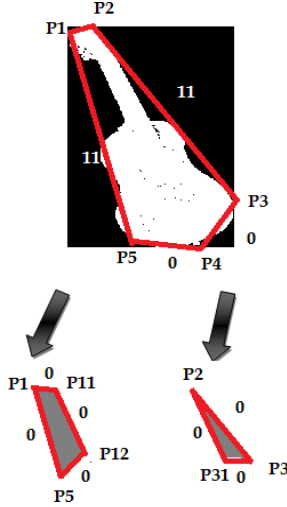
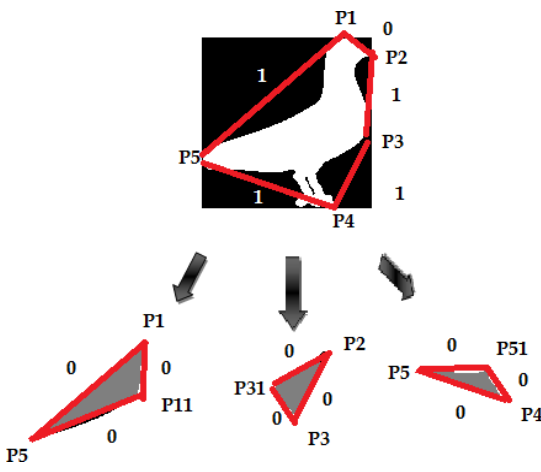
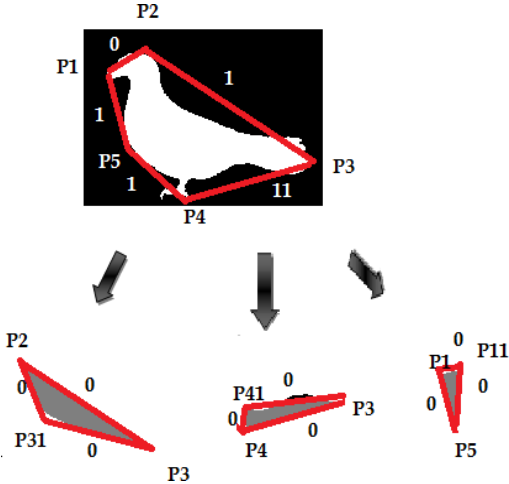
| | | | |
|---|---|---|---|
|  1 |  2 |  3 |  4 |
|  5 |  6 |  7 |  8 |
|  9 |  10 |  11 |  12 |
|  13 |  14 |  15 |  16 |

Table 1 Sample Objects from MPEG-7

| Name | Hierarchical Convex Polygonal Decomposition | Shape Code |
|---------|---|---------------------|
| bat-1 |  | 1111111000000010000 |
| bat-16 |  | 1111111000000010000 |
| apple-1 |  | 0000001000 |

| | | |
|----------------|---|-------------------------|
| <p>apple-2</p> |  | <p>0000001000</p> |
| <p>Bone-1</p> |  | <p>1110110000000000</p> |
| <p>Bone-2</p> |  | <p>1110110000000000</p> |

| | | |
|-------------------------|---|------------------------------------|
| <p>cattle-16</p> |  | <p>00111100000001000000</p> |
| <p>cattle-17</p> |  | <p>00111100000001000000</p> |
| <p>guitar-2</p> |  | <p>1011000000000</p> |

| | | |
|-------------------------|---|--------------------------------|
| <p>guitar-10</p> |  | <p>01100110000000</p> |
| <p>bird-13</p> |  | <p>01111000000000</p> |
| <p>bird-17</p> |  | <p>0111110000000000</p> |

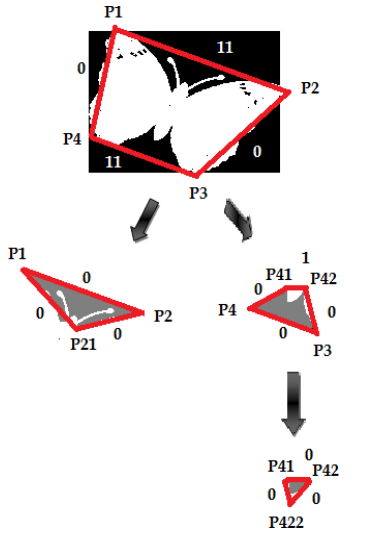
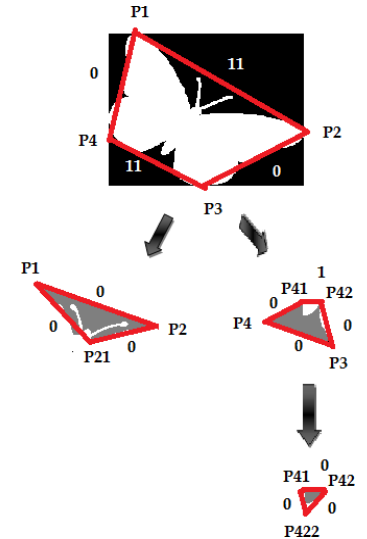
| | | |
|----------------------------|--|--------------------------------|
| <p>butterfly-12</p> |  | <p>1101100001000000</p> |
| <p>butterfly-14</p> |  | <p>1101100001000000</p> |

Table 2 Performance Score

C. Results

The following table presents the performance of our proposed algorithm for the sample data set listed in above table as compared to popular Rammer's Polygonal Shape Chain-Code [29]. As described in previous section, every class of image data set contains 20 samples and relevant retrievals are the images belonging to the class to which the query image is ideally included. One of the interesting observations during experimentation is that as the number of sides in the polygonal shape-representation of an image increases, the performance of retrieval rate falls down. However, on the average the new proposed algorithm resulted in 83.75% on Bull's eyes test, which seems reasonably good and comparable with existing state-of-the-art algorithms.

| Sample No. | Rammer's Algorithm | | Proposed Algorithm | |
|------------|--------------------|-------------|--------------------|-------------|
| | Retrieval | BEP Score % | Retrieval | BEP Score % |
| 1 | 17 | 85 | 17 | 85 |
| 2 | 14 | 70 | 15 | 75 |
| 3 | 15 | 75 | 16 | 80 |
| 4 | 14 | 70 | 16 | 80 |
| 5 | 11 | 55 | 14 | 70 |
| 6 | 11 | 55 | 15 | 75 |
| 7 | 12 | 60 | 15 | 75 |
| 8 | 13 | 65 | 15 | 75 |
| 9 | 17 | 85 | 18 | 90 |
| 10 | 16 | 80 | 19 | 95 |
| 11 | 15 | 75 | 17 | 85 |
| 12 | 14 | 70 | 18 | 90 |
| 13 | 17 | 85 | 19 | 95 |
| 14 | 17 | 85 | 19 | 95 |
| 15 | 13 | 65 | 18 | 90 |
| 16 | 14 | 70 | 17 | 85 |

V. CONCLUSIONS

A. Summary

A novel shape descriptive framework called HCPD is proposed for similar shape retrieval which exploits different degrees of convexity of an object's contour using a multi-level tree structured representation and a special shape-code to encode the HCPD-tree has also been developed to utilize the concept of popular string matching algorithms. In order to measure shape similarity between two objects, a dynamic programming strategy based algorithm has been employed which computes dissimilarity score like Levenshtein's string edit distance computation scheme. The performance of the proposed scheme is reasonably good and comparable with existing state-of-the-art algorithms. However, we need to further analyse the performance issues related to complexity of contour curvature under plausible deformations as well as we also need to investigate suitable algorithms for matching two HCPD-trees.

B. Limitations & Future Work

This paper presents a relatively new direction for object retrieval and is still in its initial stage. Extensive investigations and analysis of various stages are vital in the formulation of an accurate assessment of the limitations, and requirements of the employed shape modelling. Future work will involve refinement of various modules with special emphasis on encoding scheme to form a shape representative pattern without loss of generic but unique shape-characteristics of an object.

VI. References

- [1] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intel.* vol. 24, no. 4, pp. 509–522, Apr. 2002.
- [2] P. F. Felzenszwalb and J. D. Schwartz, "Hierarchical matching of deformable shapes," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.
- [3] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," in *Proc. IDMS*, 1997, pp. 51–58.
- [4] L. J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 22, no. 10, pp. 1185–1190, Oct. 2000.
- [5] H. Ling and D. W. Jacobs, "Shape classification using the inner distance," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.
- [6] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 25, no. 1, pp. 116–125, Jan. 2003.
- [7] T. Adamek and N. E. O'Connor, "A multiscale representation method for nonrigid shapes with a single closed contour," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 742–753, May 2004.
- [8] B. J. Super, "Learning chance probability functions for shape retrieval or classification," in *Proc. IEEE Conf. CVPR*, Jun. 2004, pp. 93–96.
- [9] E. Attalia and P. Siy, "Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching," *Pattern Recognition*, vol. 38, no. 12, pp. 2229–2241, Dec. 2005.
- [10] G. McNeill and S. Vijayakumar, "Hierarchical procrustes matching for shape retrieval," in *Proc. IEEE Conf. CVPR*, Jun. 2006, pp. 885–894.
- [11] C. Scott and R. Nowak, "Robust contour matching via the order preserving assignment problem," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1831–1838, Jul. 2006.
- [12] B. J. Super, "Retrieval from shape databases using chance probability functions and fixed correspondence," *Pattern Recognition. Artif. Intel.* vol. 20, no. 8, pp. 1117–1138, 2006.
- [13] N. Alajlan, M. S. Kamel, and G. H. Freeman, "Geometry-based image retrieval in binary image databases," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 30, no. 6, pp. 1003–1013, Jun. 2008.
- [14] M. R. Daliri and V. Torre, "Robust symbolic representation for shape recognition and retrieval," *Pattern Recognit.*, vol. 41, no. 5, pp. 208–220, May 2008.
- [15] L. Lin, K. Zeng, X. Liu, and S. C. Zhu, "Layered graph matching by composite cluster sampling with collaborative and competitive interactions," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1351–1358.
- [16] C. Xu, J. Liu, and X. Tang, "2D shape matching by contour flexibility," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 31, no. 1, pp. 180–186, Jan. 2009.
- [17] H. Ling, X. Yang, and L. J. Latecki, "Balancing deformability and discriminability for shape matching," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 411–424.
- [18] K. Nasreddine, A. Benzinou, and R. Fablet, "Variational shape matching for shape classification and retrieval," *Pattern Recognit. Lett.*, vol. 31, no. 12, pp. 1650–1657, Sep. 2010.
- [19] X. Shu and X.-J. Wu, "A novel contour descriptor for 2D shape matching and its application to image retrieval," *Image Vis. Comput.*, vol. 29, no. 4, pp. 286–294, Mar. 2011.
- [20] D. Zhang, "Review of shape representation and description techniques," *Pattern Recognit.*, vol. 34, no. 1, pp. 1–19, Jan. 2004.
- [21] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for nonrigid shapes with a single closed contour," in *Proc. IEEE Conf. CVPR*, Jun. 2000, pp. 424–429.
- [22] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- [23] J. Wang, X. Bai, X. You, W. Liu, and L. J. Latecki, "Shape matching and classification using height functions," *Pattern Recognit. Lett.*, vol. 33, no. 2, pp. 134–143, Jan. 2012.
- [24] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 357–364.
- [25] X. Bai, X. Yang, L. J. Latecki, W. Liu, and Z. Tu, "Learning context sensitive shape similarity by graph transduction," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 32, no. 5, pp. 861–874, May 2010.
- [26] X. Bai, B. Wang, and X. Wang, "Co-transduction for shape retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 328–341.
- [27] R. Hu, W. Jia, Y. Zhao, and J. Gui, "Perceptually motivated morphological strategies for shape retrieval," *Pattern Recognit.*, vol. 45, no. 9, pp. 3222–3230, Sep. 2012.

- [28] Marshall S., 1989, "Review of shape coding techniques", *Image and Vision Computing* 7(4), pp. 281–294.
- [29] Ramer U., , 1972, "An iterative procedure for the Polygonal approximation of plane curves", *Computer Graphics and Image Processing*, Academic Press, Volume 1, Issue 3, pp. 244-256.
- [30] Latecki L. J. and Lakamper R., 1999, "Convexity Rule for shape Decomposition Based on Discrete Contour Evolution," *Computer Vision and Image Understanding*, 73(3), pp. 441-454.
- [31] Ruberto C. D., 2004, "Recognition of shape by attributed skeletal graphs," *Pattern Recognition*, 37, pp. 21-31.
- [32] <http://www.dabi.temple.edu/~shape/MPEG7/dataset.html>
- [33] Navarro, Gonzalo, 2001, "A guided tour to approximate string matching". *ACM Computing Surveys* 33 (1), pp. 31–88
- [34]http://www.imageprocessingplace.com/downloads_V3/rot_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/moore.html
- [35] Graham, R.L., 1972, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", *Information Processing Letters* 1, pp. 132-133
- [36] Muller H., Muller W., Squire D. M., Marchand-Maillet S., and Pun T., 2001, "Performance evaluation in content-based image retrieval: overview and proposals", *Pattern Recognition. Letter*, 22(5), pp. 593–601.
- [37] Loncaric S., 1998, "A survey of shape analysis techniques". *Pattern Recognition*, 31(8), pp. 983–1001.
- [38] Milios E. and Petrakis E.G.M., 2000, "Shape retrieval based on dynamic programming". *IEEE Transactions on Image Processing*, 9(1), pp. 141–147.
- [39] Mokhtarian F., 1995, "Silhouette-based isolated object recognition through curvature scale space". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5), pp. 539–544.
- [40] SouravSaha, JayantaBasak and PriyaranjanSinhaMahapatra, A Hierarchical Convex Polygonal Decomposition Framework for Automated Shape Retrieval, Second International Conference on Information systems Design and Intelligent Applications (INDIA - 2015) Springer Proceedings of Second International Conference INDIA 2015, Volume 1.

Editorial & Administrative Address

SMART

SOCIETY FOR MAKERS, ARTISTS, RESEARCHERS AND TECHNOLOGISTS

6408 ELIZABETH AVENUE SE, AUBURN,
WA 98092, USA

U.S. ISSN CENTRE APPROVED