Novel Method for Embedded RISC Architecture to Reduce Memory Access Energy

T.Gnanasekaran¹, K.Duraiswamy², M. M. Arunprasath³

ABSTRACT

In embedded system, reducing program size is an important goal, because storing large size codes in a smaller memory is a problem. More than one application in one embedded system has large size codes. In order to program the device efficiently, the memory size must be large enough to accommodate the large size codes. For example, PIC16F84 microcontroller has lower memory than PIC16F877 microcontroller. So the PIC16F84 must be replaced with new PIC16F877, in order to get multiple applications. But replacing the existing device with new device or component is not so easy. In this proposed system, the existing system problems must be overcome to improve the efficiency. A way to achieve this is to restrict the size of the instructions. Shorter the instructions are obtained mainly by restricting the number of bits that encode registers. This is achieved by means of compressing the repeated instructions. While executing the codes the instructions are decompressed. This method increases the efficiency of the system and also reduces the energy.

Keywords: Instruction Compression, Instruction Decompression Table, RISC Processor.

1. INTRODUCTION

Here a new technique is introduced for reducing the energy spent by the memory-processor interface of an embedded system during the execution of firmware code. The method is based on the idea of compressing the most commonly executed instructions i.e., the instructions used by the embedded code with the highest execution probability so as to reduce the energy dissipated during memory access. This solution allows us to fix a priori the bit width of the compressed instructions. The two-fold advantage obtained from this choice is that the size of the instruction decompression table is fixed and limited, and the instruction fetching/decompression logic has reduced complexity. Here is an architecture for instruction decompression is introduced. In this architecture, the memory bandwidth and energy required to fetch the program from memory is reduced.

Here a new technique is introduced for reducing the energy spent by the memory-processor interface of an embedded system during the execution of firmware code. The method is based on the idea of compressing the most commonly executed instructions so as to reduce the energy dissipated during memory access. A major contributor to the system power budget is the memory-processor interface [5]. For this reason, several techniques focusing on memory-processor interface power optimization have been proposed. They can be classified into two broad classes: bus encoding techniques [3] and memory organization techniques.
These encoding schemes reduce interface power by changing the format of the information transmitted on the processor-memory bus. Memory organization methods change the way information is stored in memory and the address streams generated by the processor have relatively low transition activity. If the number of instructions used by the embedded code becomes large, it increases the number of bits of the compressed instructions. It is a major limitation. To overcome this limitation, increase the size of the instruction decompression table, this may excessively complicate the implementation of the controller that handles instruction fetching and decoding. Especially when bit-width of the compressed instructions is not compatible with the available memory addressing scheme. A solution is to compress only the instructions used by the embedded code with the highest execution probability. This solution allows us to fix a priori the bit width of the compressed instructions. The two fold advantage are size of the instruction decompression table is fixed and limited and the instruction fetching/decompression logic complexity has reduced. In this new architecture the memory bandwidth and energy required to fetch the program from memory is reduced.

2. ORIGINAL ARCHITECTURE

The existing system, the processor-memory architecture is shown in the figure 1. Here all the instructions are being fetched from memory. Executed with dynamic bit length k-bit size with bus architecture.

The instruction bits are being fetched by core program of the system according with instruction. The instruction code will be fetched from corresponding memory address. The instruction will be passed through desired bus architecture.

3. MODIFIED ARCHITECTURE

Another method which is shown in the figure 2, the system will be feed through compression system. The repeated instruction are passed through \( \log_2 N \). All the instructions are decompressed inverse of \( \log_2 N \). The decompression process is executed with referring Instruction Decompression Table (IDT) values. The width of uncompressed data is more than \( \log_2 N \). The modified architecture [6] has the following drawbacks.

1. The IDT may become very large so the area required is also more to store it.
2. The bit width of the compressed instructions is comparable to bit width of original instructions, thus making negligible reduction in memory bandwidth.
3. Two bytes are used to store compressed Values of \( \log_2 N \). It is not efficient.
4. TREE BASED COMPRESSION

Another method [2] proposes a Tree Based Compression (TBC). Instructions are forming an expression tree. The Huffman encoding is such an algorithm, but designing fast Huffman decoders is complicated.

5. PROPOSED SYSTEM: INSTRUCTION BASED COMPRESSION (IBC)

In order to simplify the decompression a compression algorithm called "Instruction Based Compression" is suggested, which is based on fixed-length code words. In a embedded program most of the instruction are repetitive. Taking this advantage instruction compression is proposed. i.e. Compress the instructions that are executed more often, less probable instructions are left unchanged and stored as it in the memory. This option guarantees a fixed and limited size of Instruction Dictionary Table (IDT) for all most used 256 instructions. This requires previous knowledge of a controller to handle the instruction. Because the program stored in memory is, a mix of more compressed and few uncompressed instruction set. The new proposed system for compression is based upon instruction.

Instruction based compression is motivated by the large percentage of expression trees that are composed of single instructions, the most frequent being single instruction trees [1]. Rare trees are also fairly small, while medium frequency trees are larger. So, all instruction in a program is replica of only 18.3% of its instructions, and a similar exponential behavior was again observed for single instructions. The resulting final compression ratio is on average 31.5%, and again it is achieved using only four classes. The reason for best compression is that although two entries in the TBC dictionary stores distinct trees, the trees can have similar instructions. On the other hand, entries in the IBC dictionary are unique instructions.

6. ALGORITHM

1. The algorithm divides the set of distinct trees into nc classes, each class having nk trees.

2. The number of classes (nc) is determined exhaustively by exploring all possible partitions from two to eight classes. Almost all programs, the minimum compression ratio is achieved when the partition is performed using four classes.
3. Each partition of a given number of classes, determine all possible combinations of class sizes and measure their compression ratio.

4. Smallest compression ratio is then selected as the best partition for that program.

5. Fixed-length code words of size \([\log_2 nk]\) are then assigned to trees of class \(k\).

6. For each codeword append a prefix of size \([\log_2 nc]\), that is used by the decoder to identify the class.

7. So the compression algorithm substitutes each expression tree in the program by its corresponding prefix and codeword.

The Firmware running on a given embedded processor normally uses a small subset of the instructions supported by the processor. By replacing such instructions with binary patterns of limited width. Memory bandwidth can be reduced, thus decreasing the total energy. Binary patterns are like \([\log_2 N]\), where \(N\) is the number of distinct instructions appearing in the code. Here those two \(k\)-bit instructions are said to be distinct if they differ by at least one-bit.

This solution does not require adhoc compilers. The original instructions can be automatically replaced by \([\log_2 N]\)-bit instructions by means of a script after the subset of instructions actually used by the program is identified through instruction level simulation, and the number of \([\log_2 N]\) is determined. The original machine code can thus be compressed to reduce the memory bandwidth that is needed to execute the program. The so-called instruction decompression table and the related control circuitry can be designed and placed between the processor and memory. Hence, the architecture of the core processor is left unchanged.

**Example:**

Initialize

```plaintext
clrf porta
clrf portb
bcf status.6
bsf status.5
movlw b'11011011'
*movwf porta
movlw b'00000000'
*movwf portb
*movwf adcon1
movlw b'1000100'
*movwf adcon1
bcf status.5
retlw 00
```

```plaintext
;-------------------------------------------
setdata andlw 0Fh
*movwf temp1
btfss temp1,0
bcf portb,0
btfsc temp1,0
bsf portb,0
btfss temp1,1
bcf portb,1
btfsc temp1,1
bsf portb,1
btfss temp1,2
bcf portb,2
btfsc temp1,2
bsf portb,2
btfss temp1,3
bcf portb,3
btfsc temp1,3
bsf portb,3
retlw 00
;---------------------------------------------
```
Novel Method for Embedded RISC Architecture to Reduce Memory Access Energy

*movwf temp
swapf temp,w
call setdata
LCDEN
LCDDIS
movf temp,w
call setdata
LCDEN
LCDDIS
movlw 02
call msecond
retlw 0

Instructions with prefix '*' indicates the same instruction which are repeated more than once in a program. These instructions are to be compressed by using the above-mentioned algorithm.

7. DECOMPRESSION

Figure 4. Decompressor Interface

The block diagram of the de-compressor interface with processor and instruction memory is shown in Figure 4. The core communicates with the de-compressor using the static instruction memory interface based on a four-phase process.

Figure 5. De-Compressor Block Diagram

The de-compressor block diagram is shown in Fig.5. The IDT contains 255 32-bit words. The address each word is the compressed code of the instruction stored in the word.

Decompression is performed by reading the instruction the content of IDT at the specified address by the byte of compressed instruction. The compressed instruction buffer having 32-bit register that can be accessed byte-by-byte from the memory. The results coming from memory are stored in the CIB and are decompressed one byte at a time. Final stage the control logic block gets the instruction interface signal, IDT lookup and CIB read/write images that direct transfer from memory to processor of compressed instruction.

The key function of logic control block is address generation in refer with the IDT values. If the processor reads the address in sequence the controller generates new memory address every four processor fetch cycles. The remaining three cycles, compressed instructions are executed from CIB. On the contrary if either the processor is the destination address of a branch/jump or a mark is read a cycle to instruction memory is initiated. The IDT is an asynchronous SRAM read with macro generator provided within the STM CAD Design kit. A small
amount of logic has been added between the processor and de-compressor to the MIPS/ DLX interface signals into the standard described above.

The de-compressor is a single clock edge triggered design while the processor uses a two-phase non-overlapping clock. The area and energy consumption of the de-compressor as well as the critical path delay are dominated by the IDT. IDT is a 1-Kbyte SRAM. When the de-compressor is processing compressed instructions, it performs one memory bus cycle every four fetch cycles. When it does not involve CIB refill fetch time for compressed instructions reduces to IDT read time. This is the most common case. In the remaining cases fetch latency is longer. Memory access time plus IDT read time is the time required for fetching a compressed instruction immediately after a CIB refill. The worst fetch time is experienced when the first instruction after a CIB refill is not compressed here two instruction memories reads to fetch an instruction.

Results:

Data Validity:

<table>
<thead>
<tr>
<th>S.No.</th>
<th>File size in KB</th>
<th>Compressed File Size in KB</th>
<th>Compressed Ratio in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>136</td>
<td>57</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>104</td>
<td>38</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>81</td>
<td>36</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>33</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>19</td>
<td>8</td>
<td>42</td>
</tr>
</tbody>
</table>

8. Conclusion

By using the selective instruction compression technique, the memory access is reduced and the energy spent during the memory access is also reduced. Therefore, the efficiency will be improved. The compression technique can be adopted for data also. However, data compression is not as efficient as instruction compression because the total number of repeated data is less than the total number of repeated instructions. So the compression technique.

References

Novel Method for Embedded RISC Architecture to Reduce Memory Access Energy


Author’s Biography

T.Gnanasekaran graduated from University of Madras in 1989 and completed his post graduate from Anna University, India. He is presently with BannariAmman Institute of Technology, Sathyamangalam, Tamilnadu, India, as Assistant Professor and working towards his Ph.D, in the area of Error Control Coding. His other interests include signal design for WiMax and modulation technique.

Dr.K.DuraiSwamy is working as Dean / Academic, K.S.Rangasamy College of Technology, Tiruchengode-637 209, Tamilnadu, India. His areas of interests are Computer communication and Networking.
A Novel Technique for Web Page Informative Content Extraction

M.Karthikeyan1, Krishnan Nallaperumal2, K.Senthamarai Kannan3, T.Rajesh4

ABSTRACT
The web has established itself as the dominant medium for doing electronic commerce. Consequently the number of service providers, large and small, advertising their services on the web continues to proliferate. In this paper, new extraction algorithms for mining information from web pages are proposed. Search engines crawl the World Wide Web to collect web pages. These pages are either readily accessible without any activated account or they are restricted by username and password. Whatever be the way the crawlers access these pages, they are (in almost all cases) cached locally and indexed by the search engines. An end-user who performs a search using a search engine is interested in the primary informative content of these web pages. Non-content blocks are very common in dynamically generated web pages. Typically, such blocks contain advertisements, image-maps, plug-ins, logos, counters, search boxes, category information, navigational links, related links, footers, headers and copyright information. These non-content sections must be removed from the pages to get the content blocks so that mining of knowledge can be done. The proposed algorithm outperforms to extract the content and also the space needed for storage requirements is also minimized by similarity measurements. This paper deals with the problem of identifying and extracting the informative contents of a web page

1. INTRODUCTION
Extraction algorithms for mining information from web pages are described in this paper. Also a propagation technique for identifying and accumulating all of the attributes related to a service entity in a web page is proposed.

A substantial part of the web pages, especially those that are created dynamically contains information that should not be classified as the primary informative content of the web page. These blocks are seldom sought by the users of the website. Such blocks are referred as non-content blocks [1].

Before the content of a web page can be used, the non-content blocks must be removed based on the tags and some special features [2], so that an end user can get the required contents easily. We have designed two algorithms, based on the general need of the web users, one to extract the required text features and the other to extract the image features from the web pages. To extract the textual content, our algorithm crawls the web pages and removes the non-content information. Once the non-content sections are removed, the remaining can be considered as the content sections [10]. Several algorithms are there to find the primary informative contents from the web pages. Most of them are based on the DOM (Document Object Model) tree of the web...
A Novel Technique for Web Page Informative Content Extraction

pages. However, because of the flexibility of HTML syntax, a lot of web pages do not obey the W3C (World Wide Web Consortium) HTML specifications, which might cause mistakes in DOM tree structure [7]. Moreover, DOM tree is initially introduced for presentation in the browser rather than description of the semantic structure of the web pages. For example, even though two nodes in the DOM tree have the same parent, it might not be the case that the two nodes are more semantically related to each other, than to other nodes [7]. HTML pages are developed by means of the HTML tags as per the W3C. Our algorithm crawls in to the web pages based on the tags and some key features and removes the non-content sections. Certain non-content sections may have their own specific tags or else they are removed by identifying their specific features [2], which made it easier to remove the non-content sections.

The input to the proposed algorithm is a web page and this algorithm crawls the web page based on the tags and the predefined set of specific functions to remove the unnecessary sections. Once the non-content sections are removed the remaining is considered as the content sections. Before the extracted contents from a web page can be used, it must be subdivided into smaller semantically homogeneous sections based on their contents [1]. Such sections are called blocks. A block ‘B’ is a portion of a web page enclosed within an open-tag and its matching close-tag, where the open and close tags belong to an ordered tag-set ‘T’ that includes tags, such as <TR>, <P>, <HR> and <UL>[1]. Text parts alone are extracted from each blocks, based on certain features [1].

Also the system is designed to checks the similarity of each blocks using Competitive Neural Network (CNN) pattern recognition mechanism. The first block is taken as the input and it will be compared with all the other blocks in the page. We used a threshold ( ) with a numerical value of 0.75 in our implementation. If the similarity between the two blocks is greater than the threshold then the blocks are considered as similar and only one block is stored. This minimizes the storage requirements. The threshold can be varied based on the requirements. Several familiar web pages were examined and our algorithm produces best result comparable to other similar algorithms.

An algorithm to retrieve the images from the web pages is also provided. If the user is interested to view only the image or may be, the page is a specialized image based web page, then the user can get best results with our algorithm. As per the W3C guidelines the images are in the tag <IMG>. The proposed algorithm crawls the web page to retrieve the particular tags and a path is made to folder where the images are stored. Only the images that are informative alone is displayed, which is identified by the use of pair of tags into which the images are embedded.

2. IDENTIFICATION OF NON CONTENT INFORMATION

As per the guidelines of the W3C, almost in all the web pages, either all or most of the non-content information mentioned earlier are available. The way the designers develop a web page, it may have one or more presentation styles used in it, which leads to the addition of non-content blocks. As well, the contents inside these non-content blocks may have a similarity. The proposed algorithm identifies these presentation styles and or the common contents by the use of a set of inputs provided to the algorithm. The inputs may be some ordered tags or some common features. The algorithm is designed so that the input set can be altered at any time depending on the application. For example the anchor tag <A> is mainly used for the link. The default behavior associated
with a link is to redirect the user to another web resource. Hence this information can be removed by the use of the tag set as an input to remove the non content sections. Like this our algorithm functions and removes all the non-content information and separate the content part as detailed below.

**Input**: HTML pages H1, H2,…Hn (H), sorted tag and Features (T).

**Output**: Content Blocks and their associated Page.

**Function sim** :

**Input**: Block1, Block2

**Output**: Similarity measure

**Begin**

1. For each HTML page
2. read H1
3. For each input tag and feature Do
4. Read input tag and feature
5. If non-content blocks present Then remove all
6. End for
7. Remove tags
8. Store the output blocks B1, B2, .....Bn
9. For each block
10. Read Bx
11. Read B1, B2,.....Bn excluding Bx
12. Sim(Bx : By)
13. If similarity (Bx : By) > ▲
14. Return Bx
15. Read next B
16. End for
17. Read next input tag and feature
18. End for
19. Read next (H)
20. End

**3. BLOCK FEATURES**

As mentioned in the previous section, in a HTML web page, let block B1 and block B2 are the portions of the web page, enclosed with an open and its matching close tag. The algorithm is designed in such a way to perform the search operation based on Breadth First Search (BFS) method [1],[9]. The input set of tags as per the W3C guidelines is given. This algorithm takes an input tag from the set of tags and performs the search. If a matching tag set is found, that block is separated or else, the total contents will be the output. Once a block is found, once again BFS is performed in to the separated block to find existence of any matching block is present inside it. If any matching block is found it is also separated.

For example a table can be created using the HTML tag `<TABLE>` and inside this table, table rows are created using the `<TR>` tag and the table data are created using the tag `<TD>`. By the use of the BFS, each and every block is searched for all the input set of tags and the result of the search brings the atomic blocks. Several features are added with their standard tags but not all. Atomic blocks may have features like text, images, applets, java scripts etc. The input tag set includes text, text tag, list, table, object, frame, form and script [8]. If any of the other features are important, our frame work can be modified by adding the new features.
4. Identification of Similar Blocks

The separated blocks of each web page is stored as tree in a buffer. All the blocks are checked for the similarity between them. The Competitive Neural Tree [3], [4], [5], has a structured architecture, used to identify similar content blocks. A hierarchy of identical nodes forms an m-ary tree as shown in Fig.1(a) and Fig.1(b) shows a node in detail. Each node contains m slots s1, s2, . . . , sm and a counter age that is incremented each time an example is presented to that node. The behavior of the node changes as the counter age increases. Each slot si stores a prototype pi, a counter count, and a pointer to a node. The prototypes pi • P have the same length as the input vectors x. They are trained to match the patterns obtained from each node.

The slot counter count is incremented each time the prototype of that slot is updated to match an example. Finally, the pointer contained in each slot may point to a child-node assigned to that slot.

A NULL pointer indicates that no node was created as a child so far. In this case, the slot is called terminal slot or leaf. Internal slots are slots with an assigned child-node.

A. Learning at the Node-Level

In the learning phase [5] the tree grows starting from a single node, the root. The prototypes of each node form a minuscule competitive network. All prototypes in a node compete to attract the examples arriving at this node. These networks are trained by competitive learning. When an example x ∈ γ arrives at a node, all of its prototypes p1, p2, . . . , pm compete to match it. The closest prototype to x is the winner. If d(x, pk) denotes the distance between x and pk, the prototype pk is the winner if d(x, pk) < d(x, pj) ∀ j ≠ k. The distance measure used in this paper is the squared Euclidean norm, defined as

\[ d(x, p_k) = \| x - p_k \|^2 \]

The competitive learning scheme used at the node level resembles an unsupervised learning algorithm proposed to generate crisp c- partitions of a set of unlabeled data vectors [5],[6]. According to this scheme, the winner pk is the only prototype that is attracted by the input x arriving at the node. More specifically, the winner pk is updated according to the equation

\[ p^{\text{new}}_k = p^{\text{old}}_k + \alpha (x - p^{\text{old}}_k) \]

where \( \alpha \) is the learning rate. The learning rate \( \alpha \) decreases exponentially with the age of a node according to the equation

\[ \alpha = \alpha_0 \exp(-\alpha_d \text{age}) \]

where \( \alpha_0 \) is the initial value of the learning rate and \( \alpha_d \) determines how fast \( \alpha \) decreases. The update equation (2) will move the winner \( p_k \) closer to the example x and therefore decrease the distance between the two. After a sequence of example presentations and updates, the prototypes will respond each to examples from a particular region of the input space. Each prototype \( p_j \) attracts a cluster of examples \( R_j \).
The prototypes split the region of the input space that the node sees into sub regions. The examples that are located in a sub region constitute the input for a node on the next level of the tree that may be created after the node is mature. A new node will be created only if a splitting criterion is true.

B. Life Cycle of Nodes

Each node goes through a life cycle. The node is created and ages with the exposure to examples. When a node is mature, new nodes can be assigned as children to it. A child-node is created by copying properties of the slot that is split to the slots of the new node. More specifically, the child will inherit the prototype of the parent slot. Right after the creation of a node, all its slots are identical. As soon as a child is assigned to a node, that node is frozen. Its prototypes are no longer updated in order to keep the partition of the input space for the child-nodes constant. A node may be destroyed after all of its children have been destroyed.

C. Training Procedure

The generic training procedure is described below:

Do while stopping criterion is FALSE:

- Select a block.
- Traverse the tree starting from the root to find a terminal prototype \( P_k \) that is close to \( x \). Let \( n_l \) and \( s_k \) be the node and the slot that \( P_k \) belongs to, respectively.
- If the node \( n_l \) is not frozen, then update the prototype \( P_k \) according to equation (2).
- If a splitting criterion for slot \( S_k \) is TRUE, then assign a new node as child to \( S_k \) and freeze node \( n_l \).
- Increment the counter count in slot \( S_k \) and the counter age in node \( n_l \).

Depending on the type of the search method, the second step is implemented and various learning algorithms can be developed. The search method is the only operation in the learning algorithm that depends on the size of the tree. Hence, the computational complexity of the search method determines the speed of the learning process.

Sample pseudo code used for training the network

/*Assigning inputs to each neuron*/
- Set the initial value \( i = 0 \)
- For all neuron In Input
  - Neuron.Output = Inputs(i)
  - \( i = i + 1 \)
- End

/*Calculating the weight of each neuron*/
- For all input neuron connected to This Neuron
  - netValue = netValue + (Weight Associated With InputNeuron * Output of InputNeuron)
- End

/*Calculating the error value */

/*Calculating the output */
- For each layer in Input layers
  - neuron.Update(Input* Weight)
- End

/*Calculating the Bias Value */
- Set netValue As Single = bias
- For all input neuron connected to ThisNeuron
  - netValue = netValue + (Weight Associated With InputNeuron * Output of InputNeuron)
- End
A human expert is able to provide a better threshold value to get the similarity. Based on various experiments performed on different web sites, a threshold value of $\Delta = 0.75$ is decided, which produces good results. Depending on the application this threshold can be varied to produce a better output [10]. The first block from the buffer is taken and it is compared with all the other blocks in the buffer. If the similarity value between the blocks is greater than the threshold then, the two blocks are considered to have similar contents and only one block is stored. This minimizes the space requirement for the storage purposes. Then the next block is taken and the same procedure of comparison is repeated until all the blocks are compared with all other blocks. Hence the output will not have any redundant blocks and non-content blocks.

5. REMOVAL OF IMAGES

The system is designed to retrieve images from the web pages. The Feature Extractor algorithm is able to identify only the text contents. The proposed algorithm is designed so that it can also be able to identify the image features and display it on the basis of the users' interest. Initially the images are identified by the use of the <IMG> tag, which is a standard tag for the images in HTML pages as per W3C guidelines. After the image tags are identified a path to the image where it is stored is made and the images are retrieved. In the present work the images with the extension .jpg is alone retrieved as most of the web page designers use the JPEG standard images for the informative part. Images in other format may be used for the advertisement parts. It is also possible to alter the algorithm if any other format images are necessary. Also based on the size (count) of the image pixels the image can be decided as informative or non-informative. This algorithm will find a good application where the users are interested only on the images.

Input : HTML pages H, tag <img>
Output : Images

Begin
1. For all HTML pages
2. Read H
3. If <img> present Do
4. Extract the Block and store as I
5. For all image blocks I
6. Read I
7. If extension (.JPEG) Do
8. Link Image source
9. Display the Image
10. Else read Next (I)
11. Else read Next (H)
12. End for
13. End for
End

6. EXPERIMENTAL EVALUATION

In this section, we present an empirical evaluation of our method. We also compare with two other related works.

A. b-Precision

Precision is defined as the ratio of the number of relevant items (actual primary content blocks) $r$ found and the total number of items $t$ (content blocks suggested by an algorithm) found. As the precision is calculated for blocks it is called as b-Precision. $b$-Precision = $r / t$

B. b-recall

Recall has been defined as the ratio of the number of relevant items found and the desired number of relevant items. The desired number of relevant items includes the number of relevant items found and the missed relevant
items $m$, in case of blocks, we call it as block level recall of $b$-Recall. $b$-Recall = $r(r + m)$

C. $b$-F-measure

Similar to the way it is defined in [1], we can refer to the $b$-F-measure as the contents are identified on the basis of the blocks, and it is defined as

$$b-F-measure = \frac{2 \times (b\text{-Precision}) \times (b\text{-Recall})}{(b\text{-Precision}) + (b\text{-Recall})}$$

7. PERFORMANCE COMPARISON

Both the algorithms are implemented in Jdk1.5.0 on a Pentium - based Windows platform. The $b$-Precision and $b$-Recall for the text features are calculated and also performance is compared with the LH and CE [1].

The $b$-precision, $b$-recall and the $b$-F-measure for five different sites were calculated and are shown in the tables below. The proposed algorithm outperforms LH algorithm in all sites in all categories. This may due that the LH algorithm works only at the feature level.

<table>
<thead>
<tr>
<th>Site</th>
<th>LH $b$-Precision</th>
<th>LH $b$-Recall</th>
<th>LH $b$-F-measure</th>
<th>EA $b$-Precision</th>
<th>EA $b$-Recall</th>
<th>EA $b$-F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC</td>
<td>0.85</td>
<td>0.90</td>
<td>0.87</td>
<td>0.92</td>
<td>0.95</td>
<td>0.93</td>
</tr>
<tr>
<td>CNN</td>
<td>0.70</td>
<td>0.80</td>
<td>0.75</td>
<td>0.85</td>
<td>0.90</td>
<td>0.87</td>
</tr>
<tr>
<td>YAHOO</td>
<td>0.60</td>
<td>0.70</td>
<td>0.65</td>
<td>0.75</td>
<td>0.80</td>
<td>0.75</td>
</tr>
<tr>
<td>BB</td>
<td>0.50</td>
<td>0.60</td>
<td>0.55</td>
<td>0.65</td>
<td>0.70</td>
<td>0.65</td>
</tr>
<tr>
<td>ABC</td>
<td>0.40</td>
<td>0.50</td>
<td>0.45</td>
<td>0.55</td>
<td>0.60</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Figure 2. Performance Comparison With LH Algorithm

The proposed algorithm outperforms CE algorithm also in most web sites. On analysis, it is found out that the primary reason for this may be, CE algorithm works only on block level. When compared with both the algorithms, the proposed algorithm performs better, since this algorithm works at both feature and block level. Initially the features were considered to remove the non-content blocks, and then the remaining information is divided into blocks. Also due to string comparison there is no possibility for the redundancy. The performance comparison with LH and CE is given in Fig.2 and Fig.3.

Figure 4. Performance Comparison With CE Algorithm

The snap shot images of the screen displaying the extracted text out put of our system for the given input web page is shown in Fig.4.

8. RELATED WORKS

Sandip Debnath, Prasenjit Mitra, Nirmal Pal and Lee [1] proposed an algorithm to identify the primary content of web pages by finding the inverse block document frequency based on the DOM tree structure and the minimum requirement for the content extractor is two web pages. Feature extractor is proposed only for the text features and in the block property changes from the atomic blocks they have defined. Yi and Liu [2] have proposed an algorithm for identifying non-content blocks, referred as “noisy” blocks of Web pages. Their algorithm examines several Web pages from a single Website. If an element of a Web page has the same style across various Web pages, the element is more marked as content block.

The algorithm they proposed by Lin and Ho [3] also tries to partition a Web page into blocks and identify content blocks. They used the entropy of the keywords used in a
A Novel Technique for Web Page Informative Content Extraction

Cai, has introduced a Vision-based Page Segmentation [7] algorithm. This algorithm segments a Web page based on its visual characteristics, identifying horizontal spaces and vertical spaces delimiting blocks much as a human being would visually identify semantic blocks in a Web page. They use this algorithm to show that better page segmentation and a search algorithm based on semantic content blocks improves the performance of Web searches. Ramaswamy proposed a Shingling algorithm to identify fragments of Web Pages and use it to show that the storage requirements of Web-caching are significantly reduced. Kushmerick has proposed a feature-based method that identifies Internet advertisements in a

9. CONCLUSION AND FUTURE WORK

All the related works in the filed studied are solely geared towards removing advertisements and they do not remove other non-content blocks. The proposed technique is capable of removing advertisements and other specific non-content blocks also. This can further be improved for the retrieval of informative image blocks of a web page. "Content-based" search will analyze the actual contents of the image. The term 'content' in this context might refer colors, shapes, textures, or any other information that can be derived from the image itself. Without the ability to examine image content, searches must rely on metadata such as captions or keywords, which may be laborious or expensive to produce.
REFERENCES


Author’s Biography

Nallaperumal Krishnan received M.Sc., degree in Mathematics from Madurai Kamaraj University, Madurai, India in 1985, M.Tech degree in Computer and Information Sciences from Cochin University of Science and Technology, Kochi, India in 1988 and Ph.D., degree in Computer Science & Engineering from Manonmaniam Sundaranar University, Tirunelveli. Currently, He is heading Centre for Information Technology and Engineering from Manonmaniam Sundaranar University, Tirunelveli. His research interests include Signal and Image Processing, Remote Sensing, Visual Perception, Mathematical Morphology Fuzzy Logic, Data mining and Pattern recognition. He has authored three books, edited 18 volumes and published 25 scientific papers in Journals. He is a Senior Member of the IEEE.

K. Senthamarai Kannan is currently working as a Professor in Statistics, at the Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu. He has more than 18 years of teaching experience at post-graduate level. He has published more than 30 research papers in international and national journals and authored four books. He has visited Turkey, Singapore and Malaysia. He has been awarded TNSCST Young Scientist Fellowship and SERC Visiting Fellowship. His area of specialization is 'Stochastic
Processes and their Applications'. His other research interests include stochastic modeling in the analysis of birth intervals in human fertility, bio-informatics, data mining and precipitation analysis.

M.Karthikeyan received B.E degree in Electronics and Communication Engineering from Bharathiar University, Coimbatore, India in 1990, M.Tech., degree in Computer and Information Technology from Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India. Currently, he is doing Ph. D., in Computer and Information Technology at Faculty of Engineering, Manonmaniam Sundaranar University Tirunelveli. His research interests include Data mining and Image Processing. He is a Member of the IEEE.

T.Rajesh received the M.Tech., degree in Computer and Information Technology in the year 2007 from CITE, Manonmanium Sundaranar University, Tirunelveli, India. He is currently a Lecturer in Shirdi Sai Engineering College, Bangalore. Currently he is working in the area of data mining and its application to medical informatics. He is a member of ISTE and CSI.
Detection and Removal of Cracks in Digitized Colour Paintings

G. Wiselin Jiji, L. Ganesan

Abstract

This paper presents an automated method for detection and removal of cracks in digitized colour paintings. The algorithm starts with the extraction of crack centerlines, which are used as guidelines for the subsequent crack-filling phase. For this phase, the output of four direction differential operators are processed in order to select connected sets of candidate points to be further classified as centerline pixels using crack derived features. The final segmentation is obtained using an iterative region growing method that integrates the contents of several images resulting from crack width dependent morphological filters. The methodology has been shown to perform very well on digitized paintings suffering from cracks.

1. Introduction

Many paintings, especially old ones, suffer from breaks in the substrate, the paint, or the varnish. These patterns are usually called cracks and are caused by aging, drying, and mechanical factors. Age cracks can result from non-uniform contraction in the canvas or wood-panel support of the painting, which stress the layers of the painting. Drying cracks are usually caused by the evaporation of volatile paint components and the consequent shrinkage of the paint. Finally, mechanical cracks result from painting deformations due to external causes, e.g. vibrations and impacts. The appearance of cracks on paintings deteriorates the perceived image quality. However, one can use digital image processing techniques to detect and eliminate the cracks on digitized paintings. Such a "virtual" restoration can provide clues to art historians, museum curators and the general public on how the painting would have looked like in its initial state, i.e., without the cracks. Furthermore, it can be used as a non-destructive tool for the planning of the actual restoration. A system that is capable of tracking and interpolating cracks is presented in [1]. The user should manually select a point on each crack to be restored. A method for the detection of cracks using multi-oriented Gabor filters is presented in [2]. Different approaches for interpolating information in structured [3], [4], [5], [6], [7] have been developed. A technique that decomposes the image to textured and structured areas and uses appropriate interpolation techniques depending on the area where the missing information lies has also been proposed [8]. The results obtained by these techniques are very good. A methodology for the restoration of cracks on digitized paintings, which adapts and integrates a number of image processing and analysis tools, is proposed in this paper. The methodology is an extension of the crack removal framework presented in [9]. The technique consists of the following stages:

- Crack detection.
- Separation of the thin dark brush strokes, which have been misidentified as cracks.
- Crack filling (interpolation).
A certain degree of user interaction, most notably in the crack detection stage, is required for optimal results. However, all processing steps can be executed in real time and thus the user can instantly observe the effect of parameter tuning on the image under study and select in an intuitive way the values that achieve the optimal visual result. Needless to say that only subjective optimality criterion can be used in this case since no ground truth data are available. The opinion of restoration experts that inspected the virtually restored images was very positive.

2. OVERVIEW OF PROPOSED METHOD

The method herein presented can be schematically described by the functional block diagram in Figure 1, where we identify three main processing phases: 1) preprocessing, for background normalization and thin crack enhancement; 2) crack centerline detection, for defining a set of connected segments in the central part of the cracks; and 3) crack segmentation, for finally labeling the pixels belonging to the crack. These phases are further subdivided in several steps, as follows:

**Preprocessing Phase**
1) *Background normalization*, which normalizes the input image by subtracting an estimate of the background obtained by filtering with a arithmetic mean kernel. 2) *Thin crack enhancement* by processing with a set of line detection filters, corresponding to the six orientations 0, 30, 60, 90, 120 and 150; for each pixel, the highest filter response is kept and added to the normalized image.

**Crack Centerline Detection Phase:**
- Selection of Crack centerline candidates, using directional information provided from a set of six directional Difference of Offset Gaussians filters.
- Connection of the candidate points obtained in the previous step, by a region growing process guided by some image statistics.
- Validation of centerline segment candidates based on the characteristics of the line segments; this operation is applied in each one of the six directions and finally combined, resulting in the map of the detected crack centerlines.

**Crack Filling Phase:**
- Crack filling by a region growing process using as initial seeds the pixels within the centerlines obtained in the crack centerline detection phase; the growing is successively applied to the four scales and, in each growing region step, the seed image is the result of the previous aggregation. Each one of these phases is detailed and illustrated in the next sections.

3. DETECTION OF CRACK CENTERLINES

The green channel is considered in our work as the natural basis for crack segmentation because it normally presents a higher contrast between cracks and
background. Crack centerlines are herein defined as connected sets of pixels which correspond to intensity maxima computed from the intensity profiles of the crack cross sections. Geometric Property derives from local intensity properties, the crack cross profile-taking advantage of the fact that the maximum local intensity usually occurs at the crack central pixels. To locate candidate pixels belonging to the central part of a crack, the original methodology developed by the authors for the detection of perifoveolar capillaries [10] was further adapted and extended. The underlying idea of this approach is that, the response of directional differential operators, using kernels adapted to the local crack direction, has opposite signs on the two hillsides of an ideal crack cross profile; we will, therefore, explore this fact by considering the occurrence of specific combinations of filter response signs. To carry out the initial selection of the most likely centerline segments, the magnitude of the filter response is kept on the positions that verify one of the established sign conditions; this newly generated image is then segmented using region growing in order to retain just those points where restrictive intensity and connectivity conditions meet. For each one of these segments, we compute the mean intensity, the maximum intensity, and the length of the pixel set; these features are further combined for final validation of the segments belonging to crack centerlines. The detailed processing operations involved are described in the following subsections.

A. Preprocessing Phase

Median filter is used to normalize the input image. As small cracks are very thin structures and usually present low local contrast, their segmentation is a difficult task. To improve the discrimination between these thin cracks and the background noise, the normalized image is processed with a set of line detection filters [11], corresponding to the four orientations 0, 45, 90, and 135. The set of convolution kernels used in this operation is shown in Figure 2. For each pixel, the highest filter response is kept and added to the normalized image. This resulting image is the base of all subsequent operations used for the detection of crack centerlines.

\[
\begin{align*}
\begin{pmatrix}
-1 & -1 & -1 \\
2 & 2 & 2 \\
-1 & -1 & -1
\end{pmatrix} & \quad \begin{pmatrix}
-1 & 2 & 2 \\
-1 & 2 & -1 \\
-1 & -1 & -1
\end{pmatrix} \\
\begin{pmatrix}
-1 & 2 & -1 \\
-1 & 2 & -1 \\
-1 & 2 & -1
\end{pmatrix} & \quad \begin{pmatrix}
2 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{pmatrix}
\end{align*}
\]

**Figure 2**: Line Detector Filters Used For Thin Crack Enhancement

B. Detection Of Centerline Segment Candidates

i) Initial Selection of Candidate Points:

The first operation is to extract crack centerline pixels. When a first-order derivative filter is applied orthogonally to the main orientation of the crack, derivative values with opposite signs are obtained on the two crack hillsides, which simply mean that there will be positive values on one side of the crack cross section and negative values on the other.

Cracks can occur in any direction, the selection of a set of directional filters whose responses can be combined to cover the whole range of possible orientations is required. We tested two (0° and 90°), four (0°, 45°, 90°, and 135°), and six (0°, 30°, 60°, 90°, 120° and 150°) directions of same mask and concluded that the solution based on four directions is an interesting trade-off between detection accuracy and computation time. The particular kernels used in this work are first-
order derivative filters, known as difference of offset Gaussians filters (DoOG filters), with prevailing responses to horizontal (0°), vertical (90°), and diagonal (45°, 135°) directions. The DoOG filters have demonstrated greater immunity to noise because they depend on larger kernel derivative filters.

\[
\begin{bmatrix}
-1 & -2 & 0 & 2 & 1 \\
 2 & -4 & 0 & 4 & 2 \\
-1 & -2 & 0 & 2 & 1
\end{bmatrix}
\]

Figure: 3 Kernel Of Doog Filter Used In My Work

The particular kernel used for detecting vertical centerline candidates is the row gradient filter presented in Fig 3; the other three kernels are just rotated versions of this filter. The centerline candidates that are retained after the analysis of image rows. The intensity of each pixel is representative of the filter response. Finally, the adequacy of our methodology to locate points in the central part of the cracks can be confirmed. The kernels for processing the images, and the distinctive directions that are searched for the occurrence of derivative sign combinations.

Connection of Candidate Points

One of the four images resulting from the sequence of operations described before is processed independently in order to produce a set of four connected crack segments with a common main orientation. From each image containing the selected set of candidate points in one specific direction, an initial collection of centerline segments is generated by a region growing process that is started with a set of seed points, verifying restricted value conditions, which are afterwards extended by aggregating other neighboring pixels with lower filter responses. Both seed and aggregation thresholds are defined based on statistics derived from the histogram of the image containing the set of candidate points; the values of the seeds are above a limit depending on the mean and standard deviation of this distribution, while aggregation threshold is the histogram mode. The threshold value, \( T_{\text{seed}} \), for seed selection is evaluated by the function in equation (1).

\[
T_{\text{seed}} = \mu - \alpha \sigma
\]  

In this equation, the value of the coefficient \( \alpha \) is equal for all the images of the database, while \( \mu \) and \( \sigma \), respectively, the mean and standard deviation of the distribution are dependent on the properties of each particular image. A correct choice of this threshold value is critical for the elimination of noisy segments, usually found in the background. The result of this processing sequence is a set of connected segments, as shown in the Fig 4(b).
iii.) Validation of Centerline Segments

Every candidate segment is confirmed or rejected as a valid centerline segment based on two features:

- Intensity of the segment, evaluated by the geometric mean between the average and maximum intensity values of the segment. The reference value associated with direction $x$, $I_x$, is defined in equation (2) as
  \[ I_x^{ref} = \max \{ I_x, I_y \}, \quad \text{for} \ x = h, v, d_{45}, d_{135} \] (2)
  where $I_x$ is the directional intensity value calculated from the subset of candidate segments with direction $x$ and are obtained from four subsets.

- Length of the segment, measured by the number of points of the candidate segment. The length value is approximated using a scaling factor equal to $(-1, 2)$. The length reference value associated with direction $x$, $L_x^{ref}$, is defined in equation (3) as
  \[ L_x^{ref} = \max (L_x, L_y) \quad \text{for} \ x = h, v, d_{45}, d_{135} \] (3)
  Comparing its intensity and length features with image dependent reference values that validates each centerline candidate segment. The classification of each individual segment $s$ of a particular directional set $x$ as a crack centerline segment is based on the inequality presented in equation (4). In this equation, where a product rule is used for combining the two feature values, $I_x^{ref}$ and $L_x^{ref}$ are, respectively, the intensity and length features calculated for the segment $s$, and $I_x^{ref}$ and $L_x^{ref}$ are the references obtained from (2) and (3) for the directional set containing the segment under analysis
  \[ I_x^{ref} \geq I_x^{ref} \sqrt{L_x^{ref}} \quad \text{for} \ x = h, v, d_{45}, d_{135} \] (4)

C. Crack Segmentation

a. Crack Enhancement

In our method, the background-normalized image is processed by top-hat operators using elements of increasing radius. The option for such a multiscale approach, instead of using a single monoscale operator, is justified by the expected dependence of the operator response from the vessel width. The range of the radius of the structuring elements varies from 1 to 8 pixels, covering the overall range of crack widths. For images with different scales, the set of structuring element radii should be adapted accordingly. The classical top-hat transform is defined as the difference between an image and its opened version. A problem associated with this classical implementation is sensitivity to noise, as a consequence of the fact that pixel values in an opened image are always less than or equal to the original ones; in such conditions, the difference image retains all small intensity fluctuations that can be found in the data. To overcome this inconvenience, a modification was adapted from [13], by considering two new steps in the top-hat definition: a closing precedes the opening result which is followed by a comparison, using a minimum operator, to get an image equal to the original one everywhere except for peaks and ridges. Equation (5) represents this modified top-hat transform, where $I$ is the image to be processed, while $S_c$ and $S_o$ stand for the structuring elements for closing ($\circ$) and opening ($\bullet$) operators

\[ \text{TopHat}_I = I - \min (I \bullet S_c) \circ S_o \circ I \] (5)

As a consequence of the distinct size of the structuring elements, each image is adapted to highlight cracks
Detection and Removal of Cracks in Digitized Colour Paintings

Figure 5. Output for Crack Filling

within a limited range of widths. Essentially, the images obtained with the smallest structuring elements just retain correct width data for small cracks, since only the central part of the other crack segments is represented. On the other hand, larger structuring elements are able to extract larger cracks, but thinner structures are naturally blurred.

b. Crack Filling Method

1) Crack Filling Based On Seed Points And Region Growing Technique

After identifying cracks and separating misclassified brush strokes, the final task is to restore the image using local image information to fill the cracks. We obtain a final image with the segmented cracks by iteratively combining the centerline image with the set of images that resulted from the crack segments reconstruction phase. In the first iteration, crack centerline pixels are used as seeds for a region-growing algorithm, which breed these points by aggregating the pixels in the reconstructed image derived from the top-hat operator with the smallest structuring element size. In each of the subsequent three iterations, the reconstructed images corresponding to the cracks with increasing width are in turn used for extending the output of the previous region-growing step.

The final crack segmentation is obtained after a cleaning operation aiming at removing all pixels completely surrounded by crack points, but not labeled as part of a crack. Considering that each pixel with at least six neighbors classified, as crack points must also belong to a crack does this. The final results of the iterative crack-filling phase are illustrated in Figure 5.

4. CONTROLLED ANISOTROPIC DIFFUSION

Anisotropic diffusion [12] is an image enhancement method that successfully combines smoothing of slowly varying intensity regions and edge enhancement. Smoothing is modeled as a diffusion that is allowed along homogeneous regions and inhibited by region boundaries. Anisotropic diffusion is described by the following equation (6)

\[ \frac{\partial f}{\partial t} = \text{div} \left( c(x,y,f) \nabla f \right) \]

Where, \( \text{div} \) denotes the divergence operator and \( \nabla \), the gradient and Laplacian operators with respect to the space variables \( x, y \). At each position and iteration, diffusion is controlled by the conduction Coefficients \( c(x,y,f) \). Since diffusion should be inhibited across regions separated by discontinuities, the conduction coefficients should obtain small values in pixels with large intensity gradient magnitude. The iterative discrete solution to equation (6) is governed by the equation (7)

\[ W(x,y,f)_{t+1} = W(x,y,f)_t + \Delta t \left( c(x,y,f) \nabla W(x,y,f)_{t-1} - W(x,y,f)_{t-1} \right) \]
where \( \theta = \frac{\pi}{4} \) for the scheme to be stable, N, S, E, and W are the mnemonics for North, South, East, and West, and the symbol \( \Delta \) indicates nearest-neighbor differences as in equation (8).

\[
\begin{align*}
D_N I_{i,j} & = I_{i-1,j} - I_{i,j} \\
D_S I_{i,j} & = I_{i+1,j} - I_{i,j} \\
D_E I_{i,j} & = I_{i,j+1} - I_{i,j} \\
D_W I_{i,j} & = I_{i,j-1} - I_{i,j}
\end{align*}
\]  

(8)

The conduction coefficients are evaluated at every iteration as a function of the magnitude of the intensity gradient. In our implementation, the following approximation was used as in equation (9).

\[
\begin{align*}
C_{N,i,j}^I & = g\left(D_N I'_{i,j}\right) \\
C_{S,i,j}^I & = g\left(D_S I'_{i,j}\right) \\
C_{E,i,j}^I & = g\left(D_E I'_{i,j}\right) \\
C_{W,i,j}^I & = g\left(D_W I'_{i,j}\right)
\end{align*}
\]  

(9)

The following function equation (10), proposed in [12], and has been used in our case:

\[
g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}
\]  

(10)

In our experiment, we fixed the value of \( K \) as 25. In order to fill the cracks, the anisotropic diffusion algorithm was applied selectively, in neighborhoods centered on crack pixels. All pixels within these neighborhoods participate in the diffusion process. However, only the values of the crack pixels are updated in the output image. Further improvements were obtained by taking into account crack orientation, i.e., by applying the operation only in a direction perpendicular to the crack direction. The final results of the iterative crack-filling phase by anisotropic Diffusion method are shown in Figure 6.

![Figure 6. Output For Crack Filling](a)

(a)

Figure 6. Output For Crack Filling

(a) Original Image

(b) Crack Filled (Controlled Anisotropic Diffusion)

The final results of the iterative crack-filling phase by region growing method and Anisotropic Diffusion method is shown in figure 7.
5. DISCUSSION AND CONCLUSION

In this paper, we have presented an integrated strategy for crack detection and filling in digitized paintings. Our method can be classified as a pixel processing-based approach. The initial step of crack centerline detection combines local information, used for early pixel selection, with structural features, as the crack length. Global intensity characteristics and local crack width information are adaptively exploited in the subsequent crack-filling phase. A major feature of the method is its adaptability to particular image intensity properties, as most algorithm settings are based on threshold values computed from local or global image information. In our work, cracks are filled by using region growing technique in the seed points (Figure 7.c) and Anisotropic Diffusion technique (Figure 7.d). Region growing approach gives better results. When compared with crack segmentation approaches, the present proposal assigns a hard classification result to each image point.

REFERENCES


Author’s Biography

G. Wiselin Jiji graduated her Engineering (CSE) from Manonmanium Sundarnar University in 1994 and completed her post graduation from Madurai Kamaraj University in 1998. She is currently submitted her research thesis in Anna University, Chennai. Her current research interests include remote sensing, image processing and texture analysis. He is working as Head of the Department in Computer Science and Engineering, Dr. Sivanthi Aditanar College Of Engineering, Tiruchendur. She Published 8 research papers in International Journals, one paper in National Journal and 25 papers in International /National Conferences.

Dr. L. Ganesan completed his B.E in Electronics and Communication Engg. from Thiagaraja College of Engineering, Madurai and M.E in Computer Science and Engg. from Government College of Technology, Coimbatore. He completed his Ph.D from Indian Institute of Technology, Kharapur in the area image processing. He has authored more than forty publications in reputed International Journals and more than 150 National / International conferences. His area of interest includes image processing, multimedia and compressions. He is currently working as head of the department in Computer science and engineering, A.C. College of Engg. and Technology, Karaikudi.
Discovering Significant Rules For Constructing An Network Intrusion Detction Model

Mrutyunjaya Panda¹, Manas Ranjan Patra²

¹Department of ECE, GIET, Gunupur, Orissa, India-765022, e-mail: panda_mrutyunjaya@yahoo.com
²Department of Computer Science, Berhampur, University, Orissa, India. e-mail: panda_mrutyunjaya@yahoo.com, mrpatra12@gmail.com

ABSTRACT
Network intrusion detection systems (NIDSs) have become an important component in network security infrastructure. Currently, many NIDSs are rule-based systems whose performances highly depend on their rule sets. Unfortunately, due to the huge volume of network traffic, coding the rules by security experts becomes difficult and time consuming. Since, data mining techniques can build intrusion detection model adaptively, data mining based NIDSs have significant advantages over the rule-based system. Therefore, we apply Association rule mining for network intrusion detection. Its mining algorithms discover all item associations (or rules) in the data that satisfy the user-specified minimum support (minsup) and minimum confidence (minconf) constraints. In many applications, association rules will only be interesting if they represent non-trivial correlations between all constituent items. In this paper, Apriori and predictive Apriori methodologies are used for discovering the significant rules. We also report the experimental results over KDDCup’99 datasets. Finally, the result analysis shows that the proposed approach provides the most significant rules, thereby providing strict control over false discoveries.

Keywords: Intrusion Detection, Data Mining, Association Rule Mining, Network Security.

1. INTRODUCTION
With the tremendous growth of network-based services and sensitive information on networks, network security is getting more importance than ever. Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can protect network-based systems, there are still many undetected intrusions. Thus, network intrusion detection systems (NIDSs) play a vital role in network security. Network intrusion detection systems detect attacks by observing various network activities, while Host-based Intrusion Detection systems (HIDSs) detect intrusions in an individual host. An IDS usually does not affect the normal network operation of the targets.

There are two major approaches for detecting intrusions, signature-based and anomaly-based intrusion detection. In the first approach, attack patterns or the behaviour of the intruder is modelled (attack signature is modelled). Here, the system will signal the intrusion once a match is detected. However, in the second approach, normal behaviour of the network is modelled. In this approach, the system will raise the alarm once the behaviour of the network does not match with its normal behaviour. There is another Intrusion Detection (ID) approach that is called specification-based intrusion detection. In this, the normal behaviour (expected behaviour) of the host is specified and consequently modelled [1]. In this approach, as a direct price for the security, freedom of operation for the host is limited. Currently, many NIDSs such as Snort [2] are rule based systems, which employ misuse detection techniques and have limited
extensibility for novel attacks. Their performances highly rely on the rules identified by the security experts. However, the amount of network traffic is huge, and it is very difficult to specify some intrusions using the rules. Therefore, the process of encoding rules is expensive and slow.

To overcome the rule based systems, a number of IDSs employ data mining techniques. Data mining is the analysis of (often large) observational data sets to find patterns or models that are both understandable and useful to the data owner [3]. Data mining can efficiently extract patterns of intrusions for misuse detection, establish profiles of normal network activities for anomaly detection, and build classifiers to detect attacks, especially for the vast amount of audit data. Data mining-based systems are more flexible and deployable. The security experts only need to label the audit data to indicate intrusions instead of hand-coding rules for intrusions.

Another major problem in this research area is the speed of detection. Computer networks have a dynamic nature in a sense that information and data within them are continuously changing. Therefore, detecting an intrusion accurately and promptly, the system has to operate in real time. Operating in real time is not just to perform the detection in real time, but also to adapt to the new dynamics in the network. Real time operating IDS is an active research area perused by many researchers. Most of the research works are aimed to introduce the most time efficient methodologies. The goal is to make the implemented methods suitable for the real time implementation.

Association rule discovery [4] finds collections of items that co-occur frequently in data. In many applications, such rules will only be interesting if they represent non-trivial correlations between all constituent items. In this paper, such associations are described as significant rules and all remaining associations are treated as false discoveries [5, 6, and 7]. This paper builds upon this body of previous work in terms of arbitrary statistical hypothesis tests, while providing strict control over the risk of false discoveries. The proposed approach is evaluated using the KDDCup’99 datasets, which were used for the third International Knowledge Discovery and data Mining Tools Competition [8]. The contest involved building a classifier for detecting computer network intrusions from a very large database of network traffic. Our experimental results show that the detection performance is improved by employing various interestingness measures, using Apriori and Predictive Apriori algorithm. The rest of the paper is organised as follows. The section 2, we describe about the problem statement of the proposed approach. The experiments and performance evaluations are presented in section 3. In section 4, some related work will be discussed. Finally, conclusion and future scope of the research will be outlined in section 5.

2. PROBLEM STATEMENT

In this section, an overview of the proposed framework is discussed at first. Then, we will illustrate how to apply the association rule mining to build patterns with the most significant rules for detecting network intrusions.

2.1. Overview Of The Framework

The proposed framework applies association rule mining technique to build patterns for network intrusion detection. The working environment of the proposed
Discovering Significant Rules For Constructing An Network Intrusion Detection Model

NIDS is shown in Figure 1. There are two phases in the framework: off-line phase and on-line phase. The system builds patterns of intrusion in the off-line phase and detects intrusion in the on-line phase.

In the off-line phase, training data sets are fed in to the pattern builder module, which can build the patterns of intrusions. The module employs the feature selection algorithm, handles imbalanced intrusions, and builds the patterns by association rule mining with user specified minsup and minconf constraints. After mining the patterns for intrusions, the module outputs the pattern as the input of the detector module.

In the on-line phase, the system captures the packets from network traffic. The features for each connection are constructed by the pre-processors from the captured network traffic. Then, in the detector module, the connections are classified as different intrusions or normal traffic using the patterns built in the off-line phase. Finally, the system raises an alert when it detects any intrusion.

2.2 Association Rule Mining Approach

We consider the problem of finding rules from data $D = \{t_1, t_2, \ldots, t_n\}$, where each transaction or record $t_i \subseteq I$ and $I = \{\text{item}_1, \text{item}_2, \ldots, \text{item}_m\}$ is the set of items of which transactions are composed. For market-basket data items are atomic forms and for attribute-value data items have the form $a = v_i$, where $a_i$ represents an attribute and $V_i$, a value of $a_i$. For attribute value data, no transaction $t_i$, $1 \leq i \leq n$ may contain two items $a_i = v_{ij}$ and $a_i = v_{ik}$, $j \neq k$. That is, each transaction may contain at most one value for each attribute. Rules take the form $x \Rightarrow y$, where $x \in I$ and $y \in I$.

Note that we limit the consequent $y$ to a single value. While many association rule techniques allow multiple values in the consequent $y$, the technique we present generalize directly to multiple-value consequents and a single value with multiple elements in the consequent can be represented by multiple rules with single elements in the consequent.

In this paper, we are potentially interested in a number of properties of a rule $x \Rightarrow y$ relative to $D$, and the properties vary from application to application. In this, we utilize support [4], confidence [4], lift [9], and leverage [10], defined as follows:

$$\text{Sup}(x \Rightarrow y, D) = \left| \{i : x \subseteq t_i \cap y \in t_i\} \right| \quad \text{(1)}$$

$$\text{Conf}(x \Rightarrow y, D) = \frac{\text{Sup}(x \Rightarrow y, D)}{\left| \{i : x \subseteq t_i\} \right|} \quad \text{(2)}$$

$$\text{Lift}(x \Rightarrow y, D) = \frac{\text{Conf}(x \Rightarrow y, D)}{\left(\frac{\left| \{i : y \in t_i\} \right|}{n}\right)} \quad \text{(3)}$$

$$\text{Leverage}(x \Rightarrow y, D) = \frac{\text{Sup}(x \Rightarrow y, D)}{\left| \{i : x \subseteq t_i\} \right| \left| \{i : y \in t_i\} \right|} \quad \text{(4)}$$

Note that the parameters $x \Rightarrow y$ and $D$ will be omitted from these functions where they can be determined from the context.

The original association rule task [4] was to find all rules $x \Rightarrow y$ such that $\text{sup} \geq \text{minsup}$ and $\text{conf} \geq \text{minconf}$, where minsup and minconf are user specified constraints.

Typically, rules will only be interesting if they represent non-trivial correlations between items. Relatively high
values of minsup and minconf usually deliver rules for which x and y are correlated when applied to the sparse data typical of market-basket analysis. However, as will be demonstrated in the experiments below, this is not the case for the dense data such KDDCup'99 data. Also, there is a serious problem that x may contain items that are independent of y, and hence potentially misleading. However, in most contexts unproductive rules will be of no interest so long as the first rule is known.

Apriori, one might expect there to be very large numbers of unproductive rules, as from every single productive rule $x \Rightarrow y$ many unproductive rules can be generated by inserting into x any arbitrary collections of unrelated items.

In order to alleviate this problem, we have used Predictive Apriori Algorithm to obtain various interestingness measures such as accuracy, lift and leverage, to discover most interesting ones out of many obtained for detecting network intrusions.

3. EXPERIMENTS AND EVALUATION

In this section, we summarize our experimental results to build patterns for intrusion detection over the KDDCup'99 datasets. We first describe the datasets used. Finally, we evaluate our approach based on the various interestingness measures to discover most significant rules.

3.1. Dataset and Pre-Processing

Under the sponsorship of Defence Advanced Research Projects Agency (DARPA) and Air force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of computer network intrusion detection systems [11]. DARPA dataset is the most popular dataset used to test and evaluate a large number of IDSs. The KDD'99 dataset is a subset of DARPA dataset prepared by Stolfo and Wenke Lee [12]. The data was pre-processed by extracting 41 features from the tcpdump data in the 1998 DARPA datasets. The KDD’99 dataset can be used without further time-consuming pre-processing and different IDSs can compare with each other by working on the same dataset. Therefore, we carry out our experiments on the KDDCup'99 dataset.

The KDD'99 dataset includes the full training set, the 10% training set, and the test set. The full training set has 4,898,431 connections. The 10% training set has 494,020 connections. The attacks in the data set fall into four categories [13] as shown below.

- Denial-of-Service (DoS) attacks have the goal of limiting or denying services provided to the user, computer network. A common tactic is to severely overload the targeted system (e.g. apache, Smurf, Neptune, etc.).
- Probe or Surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network (e.g. Saint, PortSweep, etc.).
- U2R (User-to-Root) attacks have the goal of gaining root or super-user access on a particular computer or system on which the attacker previously had user level access. These are attempts by a non-privileged user to gain administrative privileges (e.g. Perl, Xterm, etc.).
- R2L(remote-to-Local) attack is an attack in which a user sends packets to a machine over the internet, which the user does not have access to in order to expose the machine vulnerabilities and exploit privileges which a local user would have on the
computer (e.g. Xclock, Dictionary, guest_password, etc.).

3.2. Experimental Results

For our experiments, we choose the Apriori and predictive Apriori algorithm in Weka [14], using user specified minsup and min confidence. All the experiments are carried out in Intel Pentium4 CPU, 2.8GHz, and 512MB RAM PC.

3.3. Evaluation and Discussion

We carry out the experiment over 10% KDDCup’99 attack dataset. We evaluate the performance of our system by the accuracy and interestingness measures, such as lift and conviction, apart from the user specified minsup and minconf.

**Confidence (Strength):** The confidence of an association rule is the proportion of the isolates that are covered by LHS of the rule that are also covered by the RHS. Values of confidence near value 1 are expected for an important association rule.

**Support:** The support of an association rule is the proportion of the isolates covered by LHS and RHS among the total number of isolates. Support can be considered as an indication of how often a rule occurs in a data set and as a consequence how significant a rule is.

**Coverage:** The coverage of an association rule is the proportion of isolates in the data that have the attribute values or items specified on the LHS of the value. Values of coverage near Value 1 is expected for an important association rule. This is otherwise treated as accuracy of the discovered association rule.

Lift \((X \rightarrow Y) = \text{conf}(X \rightarrow Y)/P(Y)\), an equivalent definition is: \(P(X, Y)/P(X)P(Y)\). Lift is a symmetric measure. A lift well above 1 indicates a strong correlation between \(X\) and \(Y\). A lift around 1 says that \(P(X, Y) = P(X)P(Y)\). In terms of probability, this means that the occurrences of \(X\) and the occurrence of \(Y\) in the same transaction are independent events; hence \(X\) and \(Y\) are not correlated. Another definition can be found in [14].

Conviction \((X \rightarrow Y) = [1-P(Y)]/ [1-\text{conf}(X\rightarrow Y)]\).
Conviction is not a symmetric measure. A conviction around 1 says that \(X\) and \(Y\) are independent, while conviction is infinite as \(\text{conf}(X \rightarrow Y)\) is tending to 1. It is to be noted that if \(P(Y)\) is high, \(1-P(Y)\) is small. In that case, even if \(\text{conf}(X \rightarrow Y)\) is strong, conviction \((X \rightarrow Y)\) may be small.

**Leverage:** The leverage of an association rule is the proportion of additional isolates covered by both the LHS and RHS above those expected if the LHS and RHS were independent of each other. Leverage takes values inside \([-1, 1]\). Values equal or under value 0, indicate a strong independence between LHS and RHS. On the other hand, values near 1 are expected for an important association rule.

From the figures 2 and 3, it can be observed that, our method is an effective and efficient one, in discovering the most significant rules and reducing the false discoveries, thus avoiding the effect of combinatorial explosion, which otherwise would have caused with many insignificant rules.
Figure 2 : Predictive Apriori to Measure the Accuracy of the Discovered Rules.

Figure 3 : Apriori with Interestingness Measures
4. RELATED WORK

In recent years, many Data Mining-based research work have been proposed for intrusion detection. MADAMID (Mining Audit data for automated Models for Intrusion detection) [15] is one of the best known data mining projects in intrusion detection. It is an offline IDS to produce anomaly and misuse intrusion models. Association rules and frequent episodes are applied in MADAM ID to replace hand-coded intrusion patterns and profiles with the learned rules.

ADAM (Audit Data Analysis and Mining) [16] is the second most widely known and well published project in the field. It is an on-line network-based IDS. ADAM can detect known attacks as well as unknown attacks. Association rules and classification, two data mining techniques are used in ADAM.

IDDM (Intrusion detection using data Mining techniques) [17] is a real-time NIDS for misuse and anomaly detection. It applied association rules, Meta rules, and characteristic rules. IDDM employs data mining to produce description of a network data and uses this information for deviation analysis.

In contrast to the previously proposed data mining based IDSs, we have implemented the predictive Apriori and Apriori with various interestingness measures to discover the most significant association rules.

5. CONCLUSION AND FUTURE SCOPE

In this paper, we employ Apriori algorithms in NIDSs to discover the accurate and most significant association rules. In this way, it avoids the combinatorial explosion of having so many unnecessary rules. This model enables us to find rare item rules yet without producing a huge number of meaningless rules with frequent items. The experimental results show the effectiveness in building an efficient intrusion detection model. However, Apriori algorithm with multiple minimum support criteria may be thought of as future work, so as to develop a more powerful and flexible network intrusion detection model.

REFERENCES


Author’s Biography

Mrutyunjaya Panda holds a Master Degree in Engineering and is presently working as an Assistant Professor in the Department of Electronics & Tele Comm. Engg., Gandhi Institute of Engg. and Technology, Gunupur, India. He has 10 years of teaching experience. Currently, he is pursuing Doctoral research in Computer Science. He has about 10 publications to his credit. His research interests include Data Mining, Network Security, Intrusion Detection and Soft Computing.
Adaptive Interpolation and Sharpening for Single Sensor Digital Camera Images

Krishnan Nallaperumal¹, S.S. Vinsley², C.Seldev Christopher³

ABSTRACT

A single sensor equipped with a Color Filter Array (CFA) is used in many Digital Still Cameras (DSC) to capture any of the three primary color components, R (red), G (green) or B (blue) on each pixel location, in order to reduce the size. This single band is used to generate the true RGB image. Demosaicing is the process of interpolating the missed colors, which aims to reconstruct the missing colors as close as possible by keeping the less computational complexity. In this paper a novel adaptive edge preserving, edge directed inter-plane interpolating technique is proposed for color reproduction from blurred Bayer mosaic images. To sharpen the image, an adaptive edge directed sharpening algorithm is introduced. The proposed interpolation algorithm aims to effectively estimate the missing green component in the edge and texture regions.

Depending on the sharpness, an adaptive weighted interpolation method is introduced. Experimental results show that the proposed method performs much better than other latest techniques in terms of Color Peak Signal to Noise Ratio (CPSNR), and Structural Similarity.

Keywords: Color demosaicing, interpolation, color filter array, single sensor digital camera, sharpening.

1. INTRODUCTION

In order to reduce the hardware cost, many digital still cameras use a single sensor equipped with a color filter array to capture any of the three primary color components R, G, or B on each pixel location. Among the various suggested CFAs, the Bayer CFA pattern is the most prevalent one, where G pixels occupy half of all, R and B pixels share the others [1]. A representation of a full-color image needs all the information from the three colors on each pixel location. As a result, the missing two colors on each pixel location have to be interpolated back to get a full-color image. The process of interpolating the missing colors is called as demosaicing or color interpolation, whose main objective is to reconstruct the missing colors as close as to the original by keeping the computational complexity is very low [5] [2].

There are many interpolation methods proposed, namely nearest-neighbor replication, bilinear interpolation [14], and cubic spline interpolation, but they were not able to preserve the edge details. Other approaches are based on edge details and the reconstructions are performed in a directional way.

For reducing the various visual artifacts such as aliasing and color shifts resulted from demosaicing, camera manufacturers place a deblurring filter in the optical path [17]. The motion blur reduces both the sharpness and the

¹Professor and Head, CITET, M.S.University, Tirunelveli, E-mail: krishnan@computer.org.
²Narayanaguru college of Engineering, Manjalamoodu, E-mail: vinsleyss@yahoo.com.
³C.S.I Institute of Technology, Thovalai, E-mail: seldev@ieee.org.
resolution of the captured image. To remove this type of blur from the image, an adaptive edge directed sharpening algorithm is introduced along with a novel adaptive weighted color interpolation algorithm. This algorithm aims to estimate the optimum weight value according to edge level for interpolation. So that, the proposed algorithm performs superbly both in textured and edge regions. In section 2, image sharpening using adaptive weighted approach is introduced, Section 3 presents the details of our adaptive weighted color interpolation technique, and Section 4 presents the comparison of simulation results in terms of both Color Peak Signal to Noise Ratio and Structural Similarity[16]. Finally, concluding remarks are made in section 5.

2. PROPOSED METHOD FOR IMAGE SHARPENING

To reconstruct a full-color image from CFA samples, the two missing color values at each pixel are to be estimated from neighboring CFA samples. The green plane is estimated first and the other color planes are estimated based on the interpolated value of the green plane [6]. When the green plane is processed, for each missing green component in the CFA, the algorithm performs a gradient test, to identify edge direction and then carries out an interpolation along the direction of a smaller gradient to determine the missing green component [9]. The variance of color differences can be used as a supplementary criterion to determine the interpolation direction for the green components. The parameters $L^H$ and $L^V$ are computed by the equations (1) and (2) to determine the horizontal or vertical gradient change in the 5x5 testing window [7].

$$L^H = \frac{\sum_{\text{pixel}} \left( \sum_{m,n} \left( R_{\text{pixel}} - R_{\text{pixel}} \right) + \sum_{m,n} \left( G_{\text{pixel}} - G_{\text{pixel}} \right) \right) + \sum_{m,n} \left( B_{\text{pixel}} - B_{\text{pixel}} \right)}{\sum_{\text{pixel}} \left( \sum_{m,n} \left( R_{\text{pixel}} - R_{\text{pixel}} \right) + \sum_{m,n} \left( G_{\text{pixel}} - G_{\text{pixel}} \right) \right) + \sum_{m,n} \left( B_{\text{pixel}} - B_{\text{pixel}} \right)}$$

$$L^V = \sum_{\text{pixel}} \left( \sum_{m,n} \left( R_{\text{pixel}} - R_{\text{pixel}} \right) + \sum_{m,n} \left( G_{\text{pixel}} - G_{\text{pixel}} \right) \right) + \sum_{m,n} \left( B_{\text{pixel}} - B_{\text{pixel}} \right)$$

(2)

The leading edge direction is determined by computing the ratio of the above parameters. If the ratio, $e = \max \left( \frac{L^H}{L^V} \right)$ or $\left( \frac{L^V}{L^H} \right)$ is above threshold value then it is defined as a sharp edge block. The threshold value can be determined on statistical approach as in [7]. Better interpolation results are achieved for the threshold value 1.6 than others. Sharpening is performed with the knowledge of edge information as in equation (3).

$$S_{\text{sharp}}(i,j) = S(i,j) + \sum_{m,n} \left( \text{w}_{i+m,j+n} * S_{i+m,j+n} \right)$$

(3)

Where $m = -2 : 1 : 2$ and $n = -2 : 1 : 2$

$W$ be the weight matrix, for sharp edges with $L^V < L^H$, $W = M_1 / 10$

Where

$$M_1 = \begin{bmatrix} -1 & 0 & -2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -2 & 0 & -1 \end{bmatrix}$$

For sharp edges with

$$M_2 = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 10 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{bmatrix}$$
For smooth regions, \( W = M_3 / 8 \)

Where

\[
M_3 = 
\begin{bmatrix}
-1 & 0 & -1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 8 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 \\
-1 & 0 & -1 & 0 & -1 \\
\end{bmatrix}
\]

3. Proposed Color Interpolation Algorithm

3.1 Weighted Edge Interpolation of Green Channel

For blocks, which are classified as sharp edge blocks, weight of sharp transition is defined [10]. And the procedure for estimating the green value is derived as in equations (4) and (5).

\[
\begin{align*}
\text{if } & H L < V L, \\
G_{ij} &= S_w \left( \frac{G_{i,j+1} + G_{i,j-1}}{2} + \frac{2R_{i,j} - R_{i,j+2} - R_{i,j-2}}{4} \right) + \\
& \left( 1 - S_w \right) \left( \frac{G_{i,j+1} + G_{i,j-1}}{2} + \frac{2R_{i,j} - R_{i,j+1} - R_{i,j-1}}{4} \right)
\end{align*}
\]

(4)

\[
\begin{align*}
\text{if } & H L > V L, \\
G_{ij} &= S_w \left( \frac{G_{i,j+1} + G_{i,j-1}}{2} + \frac{2R_{i,j} - R_{i,j+2} - R_{i,j-2}}{4} \right) + \\
& \left( 1 - S_w \right) \left( \frac{G_{i,j+1} + G_{i,j-1}}{2} + \frac{2R_{i,j} - R_{i,j+1} - R_{i,j-1}}{4} \right)
\end{align*}
\]

(5)

\[
S_w = \frac{(R - G^V_i)/(R - G^H_i)}{(R - G^V_i)/(R - G^H_i)}
\]

if \( e < \text{threshold} \) & if \( L^H < L^V \),

\[
S_w = \frac{(R - G^V_i)(R - G^H_i)}{(R - G^V_i)(R - G^H_i)}
\]

The interpolation direction for the green components is estimated using the variance of color differences and the green component interpolation procedures defined in [7] are modified as (7), (8) and (9) to include the weight of smooth transition.

\[
\begin{align*}
\text{if } & B \sigma^V_{ij} = \min \left( H \sigma^V_{ij}, V \sigma^V_{ij}, B \sigma^V_{ij} \right) \text{ then} \\
G_{ij} &= S_m \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} ight] + \\
& \left( 1 - S_m \right) \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right]
\end{align*}
\]

(6)

\[
\begin{align*}
\text{if } & V \sigma^V_{ij} = \min \left( H \sigma^V_{ij}, V \sigma^V_{ij}, B \sigma^V_{ij} \right) \text{ then} \\
G_{ij} &= S_m \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right] + \\
& \left( 1 - S_m \right) \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right]
\end{align*}
\]

(7)

\[
\begin{align*}
\text{if } & B \sigma^V_{ij} = \min \left( H \sigma^V_{ij}, V \sigma^V_{ij}, B \sigma^V_{ij} \right) \text{ then} \\
G_{ij} &= S_m \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right] + \\
& \left( 1 - S_m \right) \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right]
\end{align*}
\]

(8)

\[
\begin{align*}
\text{if } & B \sigma^V_{ij} = \min \left( H \sigma^V_{ij}, V \sigma^V_{ij}, B \sigma^V_{ij} \right) \text{ then} \\
G_{ij} &= S_m \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right] + \\
& \left( 1 - S_m \right) \left[ \frac{G_{i,j+1} + G_{i,j-1} + G_{i,j+1} + G_{i,j-1}}{4} \right]
\end{align*}
\]

(9)

The weight \( S_m \) is chosen on statistical approach as 0.9, which has given the performance close to optimal.
3.2 Refinement on Red and Blue Interpolation

The refinement step focused on generating more consistent and close to optimal values for red and blue pixels using the values interpolated in initial step. The refinement scheme is based on green plane, and it processes the interpolated red and blue planes and to improve the quality of demosaicing result [13]. The proposed method is based on exploiting the image spectral correlation on two color planes. The weighted color difference value is calculated around a pixel under consideration of the weights along the four adjacent directions are estimated as follows:

\[ \alpha_{i,j} = \frac{g_{i-1,j} - g_{i,j} + r_{i-1,j} - r_{i,j}}{1 + \alpha_{i-1,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i+1,j}} \]

(11)

The color difference values at the four surrounding locations are estimated as follows:

\[ \alpha_{i-1,j} = \frac{g_{i,j} - g_{i,j} + r_{i,j} - r_{i,j}}{1 + \alpha_{i,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i,j}} \]

\[ \alpha_{i+1,j} = \frac{g_{i,j} - g_{i,j} + r_{i,j} - r_{i,j}}{1 + \alpha_{i,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i,j}} \]

\[ \alpha_{i,j-1} = \frac{g_{i,j} - g_{i,j} + r_{i,j} - r_{i,j}}{1 + \alpha_{i,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i,j}} \]

\[ \alpha_{i,j+1} = \frac{g_{i,j} - g_{i,j} + r_{i,j} - r_{i,j}}{1 + \alpha_{i,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i,j}} \]

(10)

The weights are then assigned to the four adjacent color difference values, as follows:

\[ \alpha_{i,j} = \frac{k_{R(i,j)}}{1 + \alpha_{i,j} + 1 + \alpha_{i,j} + 1 + \alpha_{i,j}} \]

(12)

Similar computation is carried out to estimate the color difference \( K_{B(i,j)} \). The red and blue pixel values are then refined using the weighted color difference values.

\[ \tilde{r}_{i,j} = g_{i,j} - K_{R(i,j)} \]

\[ \tilde{b}_{i,j} = g_{i,j} - K_{B(i,j)} \]

(13)

Thus the weighted color difference value has generated more consistent and optimal for interpolation.

4. Simulation Results

To evaluate the performance of this proposed color interpolation method, simulation was carried out with twelve 24 bit digital color images used [15].

Figure 1 shows the tested images. The effectiveness of proposed sharpening and interpolation algorithm is shown in Figure 2.

4.1 CPSNR Comparison

The CPSNR was used as a measure to quantify the performance of the interpolation methods[12]. It is calculated as in equation (14).
Adaptive Interpolation And Sharpening For Single Sensor Digital Camera Images

\[
CPSNR = 10 \log_{10} \left( \frac{255^2}{CMSE} \right)
\]

Where

Color Mean Square Error (CMSE) is calculated as

\[CMSE = \left( \frac{1}{3HW} \right) \sum_{x \in [1..H]} \sum_{y \in [1..W]} \sum_{i \in [1..3]} (I_o(x,y,i) - I_r(x,y,i))^2\]

\[I_o \text{ and } I_r \text{ represents the original and the reconstructed (interpolated) images of sizes H(height) x W(width) each.}\]

Table 1: CPSNR Comparisons

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.8</td>
<td>41.3</td>
<td>41.7</td>
<td>42.9</td>
<td>41.8</td>
<td>42.9</td>
<td>43.1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>27.7</td>
<td>37.9</td>
<td>35.0</td>
<td>38.1</td>
<td>37.6</td>
<td>39.1</td>
<td>40.0</td>
<td>40.2</td>
</tr>
<tr>
<td>3</td>
<td>33.4</td>
<td>41.4</td>
<td>40.5</td>
<td>42.7</td>
<td>40.9</td>
<td>41.5</td>
<td>42.2</td>
<td>42.1</td>
</tr>
<tr>
<td>4</td>
<td>23.4</td>
<td>31.0</td>
<td>30.5</td>
<td>35.2</td>
<td>33.8</td>
<td>35.9</td>
<td>36.4</td>
<td>37.4</td>
</tr>
<tr>
<td>5</td>
<td>32.2</td>
<td>41.3</td>
<td>39.6</td>
<td>42.6</td>
<td>41.1</td>
<td>41.9</td>
<td>43.0</td>
<td>43.0</td>
</tr>
<tr>
<td>6</td>
<td>29.0</td>
<td>37.4</td>
<td>36.2</td>
<td>39.5</td>
<td>37.5</td>
<td>38.9</td>
<td>39.9</td>
<td>40.1</td>
</tr>
<tr>
<td>7</td>
<td>31.1</td>
<td>41.8</td>
<td>37.9</td>
<td>41.2</td>
<td>41.4</td>
<td>41.6</td>
<td>43.6</td>
<td>42.7</td>
</tr>
<tr>
<td>8</td>
<td>27.8</td>
<td>38.1</td>
<td>35.3</td>
<td>40.1</td>
<td>38.5</td>
<td>40.0</td>
<td>41.0</td>
<td>40.8</td>
</tr>
<tr>
<td>9</td>
<td>31.5</td>
<td>39.1</td>
<td>38.7</td>
<td>41.3</td>
<td>39.3</td>
<td>40.5</td>
<td>41.1</td>
<td>41.4</td>
</tr>
<tr>
<td>10</td>
<td>28.4</td>
<td>36.2</td>
<td>35.7</td>
<td>39.0</td>
<td>36.6</td>
<td>38.9</td>
<td>39.1</td>
<td>39.6</td>
</tr>
<tr>
<td>11</td>
<td>30.1</td>
<td>36.5</td>
<td>36.5</td>
<td>38.3</td>
<td>36.5</td>
<td>37.7</td>
<td>38.0</td>
<td>39.5</td>
</tr>
<tr>
<td>12</td>
<td>26.8</td>
<td>32.5</td>
<td>33.8</td>
<td>34.8</td>
<td>33.6</td>
<td>34.8</td>
<td>35.0</td>
<td>38.7</td>
</tr>
</tbody>
</table>

Table 1 shows the comparison of CPSNR with seven existing methods BI [14], OR [4], ECI [13], ESSC [3], AHD [8], SA [15], VCD [7].

4.2 Structural Similarity Comparison

For image quality assessment, the Structural Similarity (SSIM) index is a highly useful metric, since image statistical features are highly spatial and non-stationary in general [16]. And at a typical viewing distance only, a local area in the image can be perceived with high resolution by the human observer at the instance. The SSIM index is defined as the function of luminance, contrast and structure. Figure 3 shows the SSIM index values comparison between the color interpolation using variances of color differences and the proposed method.
A novel highly edge preserving, adaptive weighted color interpolation algorithm along with adaptive weighted sharpening for single sensor digital still cameras equipped with Bayer color filter array is presented. This proposed adaptive weighted sharpening method effectively enhance the image with the help of edge details. The proposed weighted color interpolation algorithm makes use of the variance of color difference to estimate the interpolation direction and weight value according to the edge value for interpolating missing samples. The refinement of red and blue pixel values using weighted color difference effectively exploited the spectral correlation and resulted in consistent and optimal image quality. Simulation results show that the proposed interpolation algorithm is able to produce subjectively and objectively better results as compared with a number of existing algorithms. By introducing adaptive weighted method for texture region, the performance of the algorithm can be improved.

REFERENCES


Adaptive Interpolation And Sharpening For Single Sensor Digital Camera Images


Author’s Biography

Prof. Dr. Nallaperumal Krishnan received M.Sc. Degree in Mathematics from Madurai Kamaraj University, Madurai, India in 1985, M.Tech Degree in Computer and Information Sciences from Cochin University of Science and Technology, Kochi, India in 1988 and Ph.D. degree in Computer Science & Engineering from Manonmaniam Sundaranar University, Tirunelveli. Currently, He is Heading Center for Information Technology and Engineering of Manonmaniam Sundaranar University.

His research interests include Signal and Image Processing, Remote Sensing, Visual Perception, and mathematical morphology fuzzy logic and pattern recognition. He has authored three books, edited 18 volumes and published 30 scientific papers in Journals. He is a Senior Member of the IEEE.

S.S.Vinsley received the B.E Degree in Electronics and Communication Engineering and M.E Degree in Communication Systems from Madurai Kamaraj University, Madurai. He is currently pursuing the Ph.D degree at Manonmaniam Sundaranar University, Tirunelveli.

C.Seldev Christopher received the B.E Degree in Electronics and Communication Engineering from Manonmaniam Sundaranar University and M.E Degree in Communication Systems from Madurai Kamaraj University, Madurai. He is currently pursuing the Ph.D degree at Manonmaniam Sundaranar University, Tirunelveli.
Abstract

Generally, software engineers are poorly trained to elicit, analyze, and specify security requirements, often confusing them with the architectural security mechanisms that are traditionally used to fulfill them. There is a great demand to apply a continuous security mechanism when developing the system. One of the most ignored parts of a security-enhanced software development lifecycle is the security requirements engineering process. This study presents a structured approach for security requirement specification. A prescriptive framework, Secured Requirement Specification Framework (SRSF), has been proposed as a major contribution in this paper.

Keywords: Software Security, Security Requirement, Risk Analysis, Use Cases, Abuse Cases, Secure Software Development Life Cycle.

1. Introduction

Software is said to be secure if it can function properly despite of malicious attacks and threats. Security is important in all aspects of life, and the increasing pervasiveness and capability of information technology makes IT infrastructure security increasingly so [1]. The continual and increasing publicity given to failures of IT security demonstrate the importance of developing and assuring software to appropriate levels of security. An article by C. Mann ‘Why is Software So Bad’, concludes that bad habits and inadequate software life cycle processes have led to the development of poor software [2]. No doubt, there is advancement in the software engineering process and tools, but the literature survey reveals that the progress in improving the quality of software is still lagging [3]. This assessment can be made with respect to security by answering the question why is software so insecure and vulnerable?

Secure software is software that is able to resist most attacks, tolerate the majority of attacks it cannot resist, and recover quickly with a minimum of damage from the very few attacks it cannot tolerate. Secure software remains dependable in spite of intentional efforts to compromise that dependability [4]. The lack of rigor and discipline in the software development process, driven by the focus on short time-to-market, performance and functionality, has produced rampant security vulnerabilities that gravely affect a large range of computing environments, from small deeply embedded safety applications to large enterprise software platforms [6].

In the traditional software development lifecycle (SDLC), security is often an afterthought, and security estimation and prediction efforts are delayed until after the software has been developed. Vulnerabilities are an emergent property of software which appears throughout the development phases. Therefore, it is highly desirable to adopt a ‘before, during, and after’ approach of software security to software development process [7].

1Department of IT, Babasaheb Bhimrao Ambedkar University, Lucknow, UP India e-mail: khanaees@yahoo.com
2Department of Computer Science, Jamia Millia Islamia, New Delhi-India e-mail: kmfarooki@yahoo.com
life cycle process that includes security assurance is needed for improving the overall security of software [3].

Requirements engineering is critical to the success of any major development project. Several efforts have been made to prove that requirements engineering defects cost 10 to 200 times as much to correct once fielded than if they were detected during requirements development [4]. It is also proven by the researchers and industry personals that reworking requirements defects on most software development projects costs 40 to 50 percent of total project effort, and the percentage of defects originating during requirements engineering is estimated at more than 50 percent. The total percentage of project budget due to requirements defects is 25 to 40 percent [4]. The need to consider security from the ground up is a fundamental tenet of secure software development. While many development projects produce next versions that build on previous releases, the requirements phase offers the best opportunity to build secure software. Therefore, it is highly desirable to define security requirements during software requirement specification.

2. SECURITY LIFE CYCLE

Applications developed with security in mind are safer than those where security is an afterthought [9]. Researchers and practitioners working in the area of Software Security Engineering have focused on using so-called best practices in the software lifecycle. These are the security-enhanced software development methodology which provides an integrated framework, or in some instances, phase-by-phase guidance for promoting security-enhanced development of software throughout the life cycle phases.

Secure software development is the term largely associated with the process of producing reliable, stable, bug and vulnerability free software. There are a number of ways that this can be undertaken within traditional application development, but the most common procedures involve phased security assessments and reviews that encompass knowledge share; design and implementation assessment and regular security health checks. There are several reasons why organizations choose to follow a secure software development program [27]. Therefore, a Secure Development Process should be integrated with all phases of the software development lifecycle. It ensures that security is a consideration at all stages of software development lifecycle, from requirement analysis through design and implementation to deployment in production environments.

Literature survey reveals that much work has been done in developing such a methodology [14-21]. Following section describes security enhanced software development methodologies proposed by various researchers and practitioners.

2.1 Microsoft’s Framework (SDL)

Microsoft developed a trustworthy computing Security Development Life Cycle (SDL) in 2002 during its security pushes. The framework encompasses the addition of a series of security-focused activities and deliverables to each of the phases of Microsoft’s software development process. The entire product team focuses on updating the product’s threat models, performing code reviews and security testing, and revising documentation. The major objective of the proposed framework was to confirm the validity of the product’s security architecture documentation through a focused, intensive effort, uncovering any deviation of the product from that architecture, and identify and remediate any residual security vulnerabilities.
2.2 Oracle's Framework (OSSA)
Oracle Corporation has made an extensive effort in developing a framework to secure software development. Its product development and maintenance process includes a comprehensive set of security assurance mechanisms and processes. The goals of these processes are to improve the strength of security mechanisms and reduce the likelihood of security flaws in products. Collectively these assurance mechanisms and processes are known as Oracle Software Security Assurance (OSSA).

2.3 Comprehensive, Lightweight Application Security Process (CLASP)
John Viega, chief security architect and vice president of McAfee, Inc, made an effort in developing a framework for secured software development in 2004, and developed a Comprehensive, Lightweight Application Security Process (CLASP) to insert security methodologies into each phase of SDLC [14]. CLASP provides a well-organized and structured approach to moving security concerns into the early stages of software development life cycle, whenever possible. CLASP consists of a set of 30 process pieces that can be integrated into any software development process.

2.4 McGraw's Approach
Gary McGraw describes Seven Touch points for Software Security in his book Software Security: Building Security In [22]. This set of software security best practices referred to as touch points. Putting software security into practice requires making some changes to the way organizations build software. These security best practices have their basis in good software engineering and involve explicitly pondering the security situation throughout the software life cycle.

2.5 Team Software Process Approach (TSP)
The SEI’s Team Software Process (TSP) provides a framework, a set of processes, and disciplined methods for applying software engineering principles at the team and individual level [23]. TSP for Secure Software Development (TSP-Secure) extends the TSP to focus more directly on the security of software applications. The TSP-Secure framework is a joint effort of the SEI’s TSP initiative and CERT program. The principal goal of this framework is to develop a TSP-based method that can predictably produce secure software.

2.6 Secure Software Development Model (SSDM)
It has been observed that producing secure software requires integrating Software Engineering (SE) process with Security Engineering [17]. Simon Adesina Sodiya, a researcher at the Nigerian University of Agriculture developed a Secure Software Development Model (SSDM), which integrates security engineering with software engineering so as to ensure effective production of secure software products[2][12].

2.7 Rational Unified Process-Secure (RUP)
The Rational Unified Process (RUP) is one of the most popular and complete process models being used by developers in recent years. This process model is extended to be used in developing secure software systems by researchers at Amirkabir University of Technology (Tehran Polytechnic) [24][25], and named as RUPSec. The major objective of the RUPSec is to define a software process model in which security requirements are considered in all development phases of a computer-
based system: business modeling, requirements, analysis and design, implementation, and testing

2.8 Security Extension to MBASE

Model-Based Architecting and Software Engineering (MBASE) is a set of guidelines that describe software engineering techniques for the creation and integration of development models for a software project. The models to be integrated extend beyond Product (development) models such as object oriented analysis and design models and traditional requirements models, to include Process models such as lifecycle and risk models, Property models such as cost and schedule, and most notably success models such as business-case analysis and stakeholder win-win.

3. Secure Requirement

Software engineering research has recently focused on improving the modeling abilities in terms of non-functional requirements such as stability [10], performance [11], fault tolerance [12] and security [13]. Unfortunately, security is assumed to be a technical issue and therefore best handled during architecture and design or, better still, during implementation. Since software requirements are often written by non-technical business analysts, this is a common conclusion [8]. An extensive literature survey reveals that a lot of work has already been done on how to effectively elicit, validate, and document software requirements, which may be extended to include security at requirement specification [8].

Security should begin at the requirements level, and must cover both overt functional security and emergent characteristics. One way to cover the emergent security space is to build abuse cases. Similar to use cases, abuse cases describe a system's behavior under attack, providing explicit coverage of what should be protected, from whom, and for how long. Table 1 describes the activities proposed by various researchers and practitioners in the software development lifecycle to come up with the secured requirement.

<table>
<thead>
<tr>
<th>Microsoft SDL</th>
<th>Oracle Secure Software Assurance</th>
<th>CLASP</th>
<th>McGraw's 7 Touch points</th>
<th>TSP-Secure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review, recommend and ensures for security team plans;</td>
<td>Ensure developers are security aware;</td>
<td>Specify operational environment</td>
<td>Develop security requirement specification</td>
<td>Design security specifications</td>
</tr>
<tr>
<td>Identifies critical objectives;</td>
<td>Ensure security standards exist and documented;</td>
<td>Perform security analysis of requirements</td>
<td>Identify assets</td>
<td>Identify assets and abuse cases</td>
</tr>
<tr>
<td>Identify security feature requirements;</td>
<td>Require security feature libraries are available.</td>
<td></td>
<td>Build abuse cases</td>
<td>Develop requirements and abuse cases</td>
</tr>
<tr>
<td>Conduct risk analysis of requirements</td>
<td></td>
<td></td>
<td>Detail misuse cases</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Activities to be carried out for Securing the Requirement Phase
4. THE FRAMEWORK

Literature survey reveals that security mechanism should be implemented at the user interface level as well as at the application-under-development level. At the user interface level security mechanism started with an analysis of user's security requirements. In order to accomplish the goal of the theme on security at requirement phase, following objectives are set forth:

- To ensure that users and client applications are identified and identities are properly verified.
- To ensure that users and client applications can only access data and services for which they have been properly authorized.
- To detect intrusion attempts by unauthorized users and client applications.
- To ensure that the unauthorized malicious programs (e.g., viruses) do not infect the application or component.
- To ensure that communications and data are not intentionally corrupted.
- To ensure that parties to interactions with the application or component cannot later repudiate those interactions.
- To ensure that centers and their components and personnel are protected against destruction, damage, theft, or surreptitious replacement (e.g., due to vandalism, sabotage, or terrorism).
- To ensure that system maintenance does not unintentionally disrupt the security mechanisms of the application, component, or center.

Taking into account the objectives discussed above a roadmap or framework for developing secured software specification, an integrated and prescriptive framework SRSF is hereby proposed. SRSF has been attempted to be highly implementable and prescriptive in nature. It has been structured into a hierarchical description including premises, generic guidelines and secured requirement specification process to be followed in order as follows.

4.1 Premises

The following premises have been considered when the proposed framework is being used to develop a secured software requirement specification:

- There is no universally agreed-upon definition for each of high-level security requirement attributes.
- The set of security attributes used in the development of the framework has been defined operationally in the context.
- A common set of features for the desired requirement specification may be used to form the basis for its development.
- The recourse optimization in SDLC depends on the early use of procedure for requirement specification and uncovering of vulnerabilities as far as possible.
- The approach to risk estimate should be more applicable to identifying low security software than the highly secured code.
Secured Requirement Specification Framework (SRSF)

Figure 1: Secured Requirement Specification Framework (SRSF)
4.2 Guidelines
The guidelines before following the process to develop the secured software specification may be listed as follows;

- Assure compliance/adherence to collect a generally-accepted set of characteristics that good requirements possess.
- Identify and persist with all the security-specific issues involved in requirements engineering.
- Identify policies and standards as a source of software security requirement.
- Assure to control somehow all the extraneous and intervening factors that may affect the outcome based prediction.

4.3 The Process
The development process of the security requirement is comprised of five phases together with prescriptive steps for each and has been depicted pictorially in SRSF, Fig. 1. Such a framework has been proposed on the basis of integral and basic components for designing secured requirement specification. The first phase starts with the identifying functional and non-functional requirements. Identifying security goals for the desired specification is treated as an important task and has been put forth as a second phase, followed by the phases termed as perform security analysis of requirement, validation and testing, review and revision and packaging. An attempt has been made to symbolically represent the spirit of developing the secured requirement specification make the framework prescriptive in nature followed by a brief description of each of the phases comprising the depicted steps in the special reference to development of the same.

4.3.1 Identify Functional and Non-Functional Requirement
One of the foremost tasks of this comprehensive problem-solving activity is to identify the functional and non-functional requirements. This phase will elicit the application goals and quality goals. System context will be designed. Revision of the identified functional and non-functional requirement will be based on the review of the same. Importance of this phase lies in the fact it serves as the basis for evolving initial set of specifications to subsequent phases of development.

4.3.2 Identify Security Goals
There are five general steps required to identify the security goals including identification of security specification issues, identification of the assets, development of asset compromise cases, identification of the security objectives, and validation of security goals against assets, threats and application goals. The result is a set of security goals, which are validated by ensuring that the business goals remain satisfied.

4.3.3 Perform Security Analysis of Requirement
Before performing a security analysis, one must understand what is to be built. This task should involve reviewing all existing high-level system documentation. If other documentation such as user manuals and architectural documentation exists, it is advisable to review that material as well. This phase comprises of the sub activities including identification of security requirements, ensuring developers security awareness, identification of global security policy, conducting risk analysis of requirement.
4.3.4 Validate & Test
Common wisdom, intuition, speculation and proof of concepts may not be reliable sources of credible knowledge, hence it is necessary to place the specified requirement under testing. Testing is one of the best empirical research strategies, performed through quantitative analysis of experimental data on implementation. Testing is crucial for the success of any software measurement project. This phase comprises of assuring theoretical basis, performing expert review and examination observation, designing viable experiment, performing pre-tryout and tryout, and analyzing the result and finalizing the specification.

4.3.5 Review and Revision
This phase is informal and has been placed as the fifth phase with free-to-enter at any of the earlier phases. Basic idea of such a prescription is to have adequate enough exposure and then turn back for better review, in the light of all the previous phases. However, informal reviews and revisions may be carried out at any of the stages in the requirement specification development process.

4.3.6 Packaging
This phase is the last and conclusive phase of the specification development process. During this phase the developed requirement specification is prepared with the needed accessories to become a ready-to-use product, like any other usable product.

5. Findings and Future Work
A key verification step for the framework described in this paper is the ability to show that the system can satisfy the security requirements. An experimental tryouts and statistical analyses at a large scale with typical representative samples may be needed to standardize the framework, and will be carried out in next phase. It may assist other researchers and practitioners for more developmental activities. This framework may form the basis for the development of better-refined roadmap.

6. Conclusion
Application designed with security in mind is safer than those where security is an afterthought. Traditionally, security issues are first considered during the Design phase of the software development life cycle once the software requirement specification has been frozen. This paper has presented a prescriptive framework for security requirement specification comprising of six steps including identification of functional and non-functional requirement, identifying security goals, performing security analysis of requirement, validation and testing, review and revision, and packaging. The developed framework may be used to ensure the software requirement specification contains the security specifications which helps improve the security of application and reduce the cost of re-work later.

References


Secured Requirement Specification Framework (SRSF)


Author’s Biography

Dr. R. A. Khan is currently working as a Reader in the Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, UP. His area of interest is Software Security, Software Quality and Software Testing. He has authored two books on software quality and software testing.

Dr. K. Mustafa is currently working as a Reader in the Department of Computer Science, Jamia Millia Islamia New Delhi-India. He has published several papers and articles in Internationals and National Journals. He is the author of books “Software Quality: Concepts and Practices” and “Software Testing: Concepts and Practices”. 
Detection And Tracking Of Moving Object

Amrita A. Manjrekar¹, P.P. Halkarnikar²

ABSTRACT
Motion detection and object tracking algorithms are an important research area of computer vision and comprise building blocks of various high-level techniques in video analysis that include tracking and classification of trajectories. Detecting moving objects is very important in many application contexts such as people detection, visual surveillance and so on. We present a system for event detection and analysis from video streams. Our approach is based on a detection and tracking module which extracts moving objects trajectories from a video stream. These trajectories, together with a rough description of the scene, are then used by the behavior inference module in order to recognize and classify object motion. The hierarchical tasks are performed on a buffered set of frames in order to provide accurate results by taking into account the temporal coherence of moving objects.

Keywords: Video Stream, Tracking, Motion Detection.

1. INTRODUCTION
In recent years, motion analysis has become essential in many vision systems related to time requiring examination. The rising interest in this research is in conjunction with the immense attentions of employing real time application to control complex real world systems such as in the case of traffic monitoring, airport surveillance and face verification for ATM security. Real Time Computer Vision (C.V) system can track multiple non rigid/ rigid objects, such as buildings, people, moving objects, obstacle [13]. Real-time object detection system is challenging field in Computer Vision (C.V), Robotics and Surveillance system. In [8] applications of motion detection are basically divided into two categories which are as follows:

1. Control Applications
   • Obstacle Detection
   • Auto Navigation System
   • Head Tracking for Video Conferencing

2. Surveillance/Monitoring Applications
   • Security Cameras
   • Traffic Monitoring
   • People Counting

It was first necessary to determine the community’s video surveillance needs. This was accomplished through the development of a survey to which state and local law enforcement agencies responded. The survey focused on uncovering the kinds of video surveillance assignments required of a typical police department. Those assignments (and the particular users’ needs to accomplish the assignments) would lead naturally to the specifications for equipment [3]. The critical locations that were placed in the spotlight of surveillance were:

¹Department of Computer Science and Tech, Shivaji University, Kolhapur, e-mail : amrita_manjrekar2004@yahoo.com,
²Assistant Professor, Dept of CSE, D. Y. Patil College of Engineering, Kolhapur, e-mail : pp_halkarnikar@rediffmail.com
Detection And Tracking Of Moving Object

- Airports
- Train stations
- Military compounds and airbases
- Public "hotspots"

Each of the above-mentioned locations can be a potential threat and requires situational awareness to the possible effective response of the relevant enforcement agencies. Digital technology emerged as the ultimate facilitator for surveillance needs, which enables flexible, real-time, highly manageable and tunable solution [11].

2. SYSTEM ARCHITECTURE

We propose architecture of system as depicted in Fig. 1. The main modules of system, from left to right, are: the frame grabbing module, the object detection module, the object tracking module, the motion analysis module, and finally the output module. Operation and description of each module in the system is as follows:

- **Object Detection Module**

  This module is responsible for invoking the detection algorithm. Ideally, the detection algorithm is to be run on each input frame. However, this will inhibit the system from meeting its real time requirements. To speed up the process, the detection algorithm does not look for object in the entire frame. Instead, it looks for independently moving object in the regions determined to be foreground regions. To determine the foreground regions, a stabilization algorithm is used to align the current frame with a preceding frame and with a succeeding frame. After alignment, the current frame is subtracted from the two other frames. The result of each subtraction is thresholded to form a binary image that represents the locations of foreground objects in the two subtracted frames. To know the locations of the foreground objects in the current frame, the results of the two subtractions are combined by an AND operation.

- **Object Tracking Module**

  This module processes frames and detections received from the object detection module, and retain information about all the existing tracks. When a new frame is received, the already existing tracks are extended by locating the new bounding boxes locations for each track in this frame. If the frame is received accompanied with new detections, the new detections are compared to the already existing tracks. If a new detection significantly overlaps with one of the existing tracks, it is ignored. Otherwise, a new track is created for this new detection [10].

![System Architecture](Image)

Figure 1: System Architecture

- **Frame Grabbing Module**

  The frame grabbing module is responsible for dealing with the input device. The input device can be a digital video camera connected to the computer, or a storage device on which a video file or individual video frames are stored. This module abstracts the nature of the input device away from the rest of the system so that changing the input device does not affect the rest of the system.
• **Motion Analysis Module**

When the length of a track exceeds some specific length, the motion analysis module is invoked. The motion analysis module analyzes the periodicity encountered in the track. Based on the result of this analysis, it decides whether the tracked object is moving or not. This way, the detection results are double checked by the motion analysis.

• **Output Module**

When each track in a frame has been either analyzed by the motion analysis module, or removed because of being too short to be analyzed, this frame is ready for output and passed to the output module. The output module marks the detected human locations in the frame and sends it to the output device, which can be the display monitor or a storage device.

3. **EXPERIMENTAL RESULT**

The main tasks of the software is to read the video stream which is stored on disk or capture live video data using web-camera and then process that video stream to detect and track the moving object.

Following are the steps to be performed:

1. Grab the frame
2. Model the background and subtract to obtain object mask
3. Filter to remove noise
4. Group adjacent pixels to obtain objects
5. Track objects between frames to develop trajectories

Our implementation of background subtraction algorithm is partially inspired by the study presented in [12] and works on gray scale video imagery from static camera. Background subtraction method initializes a reference background with the first few frames of video input.

Then it subtracts the intensity value of each pixel in the current image from the corresponding value in the reference background image. Temporal differencing makes use of the pixel-wise difference between two or three consecutive frames in video imagery to extract moving regions. It is a highly adaptive approach to dynamic scene changes. We implemented a two-frame temporal differencing method in our system. Let \( I_n(x) \) represents the gray-level intensity value at pixel position \( x \) and at time instance \( n \) of video image sequence \( I \) which is in the range \([0, 255]\). The two frames temporal differencing scheme suggests that a pixel is moving if it satisfies the following:

\[
|I_n(x) - I_{n-1}(x)| > T_n(x) \quad (3.1)
\]

The per-pixel threshold, \( T \), is initially set to a pre-determined value and later updated.

Our system was experimented on a set of video sequences. It has succeeded to demonstrate robustness and close to real time performance in low quality video sequence. In this section, we will present the results of sequences. In the figures presented, bounding green line is the output of the detection algorithm.
Detection And Tracking Of Moving Object

Figure 2: (a), (c), (e) - Original Image in Video Sequence
(b), (d), (f) - Detected & Tracked Gray Scale Image with Bounding Box

4. CONCLUSION
In this paper, a system for object Detection and tracking has been presented. The method will be based on the integration of detection and tracking module and a behavior inference module, and can handle noisy data and inaccurate detections. This is single moving object detection with stationary background. Further research will extend the system to detect multiple moving objects with stationary background. Improving the reliability of the behavior inference module will increase its stability with regard to false and non-detection of the moving objects. Handling these types of situations, which are likely to occur in processing real video streams, requires an automatic tuning of the scenario recognition methods. Further research is planned to model and recognize more complex scenarios. Issues like robustness of recognition and validation can be handled through intensive testing.

REFERENCES
[4] Fabian Campbell-West and Paul Milleri, "Evaluation of a robust least squares motion detection algorithm for projective sensor motions parallel to a plane", Institute of Electronics, Communications and Information Technology (ECIT) Queen's University Belfast Belfast, BT3 9DT.
[5] Gérard Medioni Ram Nevatia Isaac Cohen, "Event Detection and Analysis from Video Streams", University of Southern California Institute for Robotics and Intelligent Systems, Los Angeles, CA, 90089-0273 fmedioninjnevatiajicohen@iris.usc.edu, This research is supported by the defense Advanced Research Projects Agency (DARPA) under contract DAAB007-97-C-J023, monitored by US Army, Fort Monmouth, NJ.


[7] Nianjun Liu and Brian C. Lovell, "MMX-Accelerated Real-Time Hand Tracking System", Intelligent Real-Time Imaging and Sensing (IRIS) Group School of Computer Science and Electrical Engineering, The University of Queensland, Brisbane, Australia 4072 Email: {nianjun,

[8] Ronan Fablet, Patrick Bouthemy, Marc Gelgon, "Moving Object Detection In Color Image Sequences Using Region-level Graph Labeling", Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France e-mail : rfablet@irisa.fr, Proc. 6th IEEE Int. Conf. on Image Processing, ICIP’99, Kobe, October 1999


Miss. Amrita Arvind Manjrekar is pursuing M.Tech (Computer Science & Technology) from Shivaji University. She received the B.E (Computer Science & Engineering) from D. Y. Patil College of Engineering, Kolhapur in 2004. She worked as a Lecturer, Dept of Computer Science & Engineering, in D. Y. Patil College of Engineering & Technology. Her area of interest is including Computer Vision, Video Stream Processing & Web Technology.

Author’s Biography

Pratap P. Halkarnikar received B.E. from Government College of engineering, Pune in 1986. M.E. from Walchand College of engineering, Sangli in 1993. Presently he is working as Assistant Professor in Department of Computer Science and Engineering at D.Y.Patil College of engineering, Kolhapur. Visiting professor at P.G. center, Shivaji University, Kolhapur. He is consultant to many industries for development of microcontroller based products. His interest lies in microcontroller based instrumentation, computer vision and web technology.
Processing of Medical Records from Unstructured Medical Transcripts

Vinod Chandra S. S.

ABSTRACT
The work looks at the problem of creating a framework that identifies information in the free text medical records and maps that information into a structured representation containing clinical terms. The content of the medical records are relatively stereotyped sentence types based on its specialized word usage. This regularity makes it possible to determine a set of sublanguage-specific word classes, which correlate with the types of information conveyed in the subfield; these word classes form the bridge between the structure (syntax) of the sublanguage text and their information content (semantics). We can define sublanguages as the part of the English language used in body of the texts dealing with a particular domain. A parser acts between the medical record and the sublanguage, which extracts the medical data from the input document.

Keywords: Medical records, Medical data, Medical parser, NLP, Rule base systems

1. INTRODUCTION
Clinical medical record contains a wealth of information, largely in free-text form. Information extraction in structured format from free-text records is an important research endeavor [1]. A medical document in free text format contains information that is useful for various purposes like medical coding. But keeping track of such lengthy documents is a tedious process. This approach is to observe a large number of medical documents and find a common pattern of reporting diagnosis procedures and symptoms. The information contained in the key phrases is represented in a table like structure whose columns corresponds to the major sublanguage word classes. Different column combinations are possible in a basic sub-language sentence type. The idea is to organize the sub-language sentence into compact tabular representation so that the content of the document can be quickly inspected. In order to represent the information uniformly, the syntactically conveyed connections are translated into the occurrences of particular combinations of column entries in the format.

The growing interest in automated and integrated medical records has spurred intense research into indexing, abstraction and understanding clinical text. Several applications are enabled with Natural Language Processing (NLP) technology, but all are essentially text mining operations in terms of source documentation [2] specificity and depth of information required. The required information can range from a desire to determine which course of treatment are effective for particular conditions per patient group to wanting to know where the latest outbreak of community acquired diseases is taking shape so that a sales force can be first to market. Similarly, Demographic information is important first to identify and individual patients across multiple medical encounters.
Some of the interesting works related to medical information processing is discussed here. The ARBITER (Arterial Biology for the Investigation of the Treatment Effects of Reducing Cholesterol) is the application developed, and used MEDLINE citations [11] for information extraction. Medical Language Processing (MLP) and tagging of the medical text [4, 5] discusses statistically significant POS n-gram type overlaps of newspaper language and medical sublanguage, which has not been recognized before. A Dialogue-Based System for Identifying Parts for Medical Systems [6] describes a system that provides customer service by allowing users to retrieve identification numbers of parts for medical systems using spoken natural language dialogue. They showed a results of extremely encouraging with the system being able to successfully process approximately 80% of the requests from users with diverse accents. The use of clinical data present in the medical record to determine the relevance of research evidence from literature databases [3, 7] discussed. Here they used conventional information retrieval system for analysis of the patient's data record. Three algorithms are discussed in the above work.

Medical information processing from an unstructured text is highly related to NLP. This work deals with set of sub-languages and a parser. The parser interacts between the sub-languages and input document. A natural language processing tool for analysis the medical text and the information is extracted in a specified format using a parser.

2. PARSER DESIGN

Document Analysis

Medical document analysis that occurs in clinical documents and their associated lexical attributes are shown in Figure 1. We can observe that the lexical attributes appearing in the medical documents are the lexicon or semantic classes which we have to develop in order to arrive at the detailed analysis of the clinical documents. The statement of a medical fact is composed of a subject and a predicate (SUBJECT and PREDICATE) each of which has associated an atomic attributes or lexicons. The SUBJECT may be physically absent in the statement being modeled, but if so, it is implicit. The Medical Fact can be divided into subtypes like Clinical Fact, Treatment Fact and Response Fact [8]. Clinical Fact subtypes are distinguished by the paragraph they occur in: Examination, Diagnosis, Lab Test, History etc. The Treatment Fact subtype is subdivided into general medical management (GEN), Surgery (SURG), medications (MEDS) and all other therapies (Comp).

The idea of classifying the medical document into various fact classes is the use of different lexicons in each class. The Laboratory Fact will be characterized by the description of a test (Text box) with its associated attributes shown in the box attached to the test box. These attributes are the set of lexical or semantic classes which distinguishes the Laboratory statements from the others. The Treatment Fact is likewise distinguished from other statement types by the lexical classes shown in the box attached to the Treatment Fact box, and similarly for the Response Fact. An instance of treatment fact is often coupled to a Response fact via a response relation.

The extracted information is grouped under several medical classes like treatment fact, medical fact, test fact etc. Some information may have associated contents like affected body parts, test/procedure performed, extent of damage etc. The output must clearly state the patient state, diagnosis and procedures. The input document statements are expected to follow simple grammar rules.
so that the correct sentence structure can be defined and matching patterns can be found out. Also transcript templates may affect accuracy. So a common template is expected. Lexicons that cover the required words need to be present. The sizes of word storage files cause an increase in loading time but decrease errors. Still steps may be taken to avoid extra loading time and increased response time. The design overview is shown in Figure 2. A brief explanation of each section is given.

**Figure 1 : Analysis of Medical Documents**

**Normalization**

This step performs the initial preparation of text document. The document is read line by line. From every line each character is examined. ‘:’ is replaced by new line. Remove ‘.’ other than delimiters. For example the line is searched for patterns like ‘Mr.’, ‘Mrs.’ etc and ‘.’ are removed. Multiple spaces are grouped into single space. Finally ‘.’ is replaced by new line. This step effectively helps in correctly identifying sentence boundaries.

**Sentence Marking**

Document is split into several sentences based on delimiter (new line) and stored in sentence objects. Each sentence consists of several word objects. Words are marked using spaces as delimiters. The success of this step depends on how effectively all symbols other than proper delimiters are removed in the previous step.

**Figure 2: Parser Design Overview**

**Figure 3. Noun Finding Algorithm**

```
Procedure noun_finding ( )
Begin
    Read input line;
    Search a word in lexicon;
    If found
        Mark (noun) & Exit ( );
    If word ends with ‘s’ or ‘es’
        Search a word in lexicon;
        If found
            Mark (noun, plural) & Exit ( );
    If word ends with ‘ies’
        Replace ‘ies’ by ‘y’
        Search a word in lexicon;
        If found
            Mark (noun, plural);
    End;
```
Section Formation
A list of candidate section headings is created by examining each line of text and comparing the scanned text against regular expressions similar to the stored patterns. Each section heading may appear in several forms. These are mapped to common terminology. For example: Complaint: chief complaint, complaint

Allergy
allergy, allergies, allergy medication, allergy to medication

Physical exam
physical examination, physical exam, physical findings.

A section list is maintained, which stores the section headings in the order they appear in the document along with the offsets, with respect to the original note, of the first and last sentence in the document. The module also identifies a single section heading split across two lines.

Patient Details Extraction
Due to the special structure of medical document, certain sections formed will contain patient specific information like name, age, temperature etc. Several separate parsers are used to extract each type of information. Identify section heading 'Name': the content will be patient name. Then find sentences with words 'pulse' and find following number to get pulse. Next find sentence containing patterns like nn/nn and record as blood pressure. Then find word temperature or locate patterns nn0C or nn0 degree. The number will be patient temperature. Finally remove all sentences which are processed above.

Parts of Speech Tagger (POS)
POS tagger [9] aims at marking all the word classes like noun, verb or prepositions. The tagger is divided into two main phases of operation - lexical analysis, and contextual analysis. The lexicon contains all the possible parts of speech, such as noun, verb, or adjective, appropriate to each word contained therein. Each word from the text document is first marked with all the parts of speech listed for that particular word in the lexicon. If a word does not appear in the lexicon, the tagger will default to mark it as an unknown noun. The algorithm for finding a noun is given Figure 3.

Similar steps are undertaken for verbs which appear in forms like verb, verb + 3rd person, verb + ing, verb in past tense, irregular verbs. Verb in past tense form produce an ambiguity since it is past participle. So the word is marked as ambiguous. Each word object contains a status word where each bit corresponds to particular word class. Using several contextual rules, the contextual analysis phase processes the text further to ensure that the part-of-speech tags are disambiguated. First rule is used for this - if precedes with have/has/had mark it as participle. With this information, the tagger is able to determine the final part-of-speech tag for each word.

Phrase Processing
This processes the extracted phrases and transforms complex phrases into simple canonical syntactic structures of the medical sublanguage stored into the knowledge base [10]. The main functions includes the use rules to extract phrases from sentences, unknown words are grouped to phrases (this may be a medical term) and mark as unknown others. Noun phrases are extracted using a finite set of rules, composed of different sequences of part-of-speech tags [11]. The limit for the longest recognizable noun phrase pattern was set to seven words in length, with the shortest
pattern being obviously a noun phrase of length one, the single noun. The seven-word limit can lead to some error, as the tagger is likely to misidentify noun phrases longer than seven words as two completely separate noun phrases, which themselves may or may not be valid terms. The rules were applied to the tagged words from the text, using a sliding “window” of seven words. As the window slides over the words of the text, the noun phrase patterns are applied to the window contents. When encountered, the sentence delimiters will truncate the window. Since some of the rules are subset of other rules, the longest matching rule is used to determine the “best” noun phrase. Once a noun phrase is located, the window will slide to the next word following the phrase and commence reading the contents of a new seven-word window. Similar steps are undertaken for identifying verb phrases and preposition phrases.

Standardization

Standardization is performed so that there is uniformity through out in representing related terms. This helps in reducing the number of required sentence forms to be identified and aids in better understanding of the sentence. There are three forms of standardizations. Normalizing the phrase is used in canonical names wherever possible, to make text syntactically uniform. Which replace the modifiers by its root word through lookup in the synonym table. The Decomposition simplifies the complex phrases by replacing them by two simple phrases (Example, "Suffering from severely increasing cough and fever" to "suffering from severely increasing cough" & "severely increasing fever"). Finally, flatten the document, changes a sentence structure (Example, "no evidence of pneumonia" to "no pneumonia").

Classification

The basic sub language sentence types are identified from the given set of sentences. The characteristic combinations of sublanguage word classes in the SVO (subject-verb-object) relation. For example, in the sublanguage of clinical reporting, a frequent SVO sequence consists of a subject from the PATIENT class (usually patient or a pronoun) followed by a VERB in the V-PT class (a verb whose subject is characteristically a PATIENT noun in the sublanguage, Example, have, develop), followed by a word in the SIGN-SYMPTOM class (Example, cough, dyspnea). This basic sequence type (SVO = PATIENT + V-PT + SIGN-SYMPTOM) is seen in such text occurrences as Patient has had cough for past 4-5 months (SVO = patient + have + cough) and she had an episode of severe dyspnea while in Bangalore (SVO = she + have + dyspnea).

Another similar and rather simpler rule is BODY-PART + TEST + V-SHOW + SIGN-SYMPTOM which is illustrated in the sentence Chest x-ray shows aggressive bilateral pneumonia. Due to the complexities of the natural language, a pattern larger than the subject + verb + object pattern is needed to capture information patterns of the sublanguage. For the sentences like (chest x-ray shows density) there is a larger pattern like BODY-PART + TEST + V-SHOW + RESULT. It is possible that in some cases the BODY-PART may not be explicitly present but is implicit in the TEST word (urinalysis
implies urine). These medical classes and their associated information are stored in an object for medical summary.

**Synonym Knowledge Base**

The synonym base is used for mapping the regularized structured form to controlled vocabulary concept which is the final stage of Phrase Processing module described in later part of the report. The synonym knowledge base consists of associations between standard output forms and controlled vocabulary concepts. Some of the examples are shown below. The first argument of the synonym specification is the target or standard form of the textual phrase, the second is the controlled vocabulary concept, and the third is the semantic category of the synonym.

- Synonym (show, appear, moderate certainty, certainty)
- Synonym (enlarged heart, cardiomegaly, central finding)
- Synonym (nodular density, nodular opacity, partial finding)
- Synonym (severe, high degree, degree)
- Synonym (smaller body_location, decrease in body_location size)
- Synonym (without, with no, certainty)

**Rule Base**

Rule base is required to connect individual words to noun or verb phrases. The rules for noun phrases may be represented in Extensible Markup Language (XML) file as follows

```xml
<NounPhrase><Val>N</Val> </NounPhrase>
<NounPhrase><Val>PN</Val> </NounPhrase>
<NounPhrase><Val>PN</Val> </NounPhrase>
<NounPhrase><Val>DT</Val> <Val>N</Val> </NounPhrase>
```

The rule base is also required to map the words of the key phrases in the composite tabular structure. The simplest rule which was also earlier described in the synopsis containing a SUBJECT + VERB + OBJECT is PATIENT + V-PT + SIGN-SYMPTOM. XML files store these rules in the following format

```xml
<Rule><term>PT</term> <term>V-PT</term> <term>SYM</term></Rule>
```

3. **RESULTS AND DISCUSSION**

This paper described a system for processing the patient discharge summaries and mapping the information into a database. The word classes and the controlled vocabulary of a sublanguage grammar have been shown effective for mapping textual information into a semantically structured database. Medical text mining is characterized by market requirement for very precise information at moderately deep level. An important part of the implementation process would be to maintain the precision and accuracy of the information extraction. The volume of data is growing every year and large portion of the data may be idiosyncratic or specific to the hospital or organization. The proposed framework with its inclusion of controlled and custom vocabulary takes care of the region or hospital specific jargon. Once we have the above knowledge base with us what remains is the implementation part. In order to ensure that the proposed technique when applied to medical documents produces reliable results a procedure should be developed for the quality control.

**Case Study**

The clinical information from a medical text after parsing is shown below. Here the input file is an unstructured text document with many pages.
Patient Name: PEREZ, LUIS
Sex: Male
Age: 35 years

History Of Present Illness
Luis Perez was seen at the request of Dr. Webb for evaluation of severe back and right leg pain. He is a pleasant 36 year old male who states symptoms began in August and it persisted. He states he can only walk with use of a walker. He denies any previous symptoms or problems. Currently he denies any weakness, loss of bowel or bladder function. He complains of severe pain frequently in his right leg all the way to his toes.

Physical Examination
His examination demonstrates young male who using a walker. He has some notable antalgic gait. He has tenderness with spasm at the paralumbar region. Range of motion is very limited. He has notable nerve tension sign with positive straight leg raise, positive Laségue, positive flip test of the right lower extremity. Long track signs are same. He has weakness with both toe walk and with his anterior tib on the right compared to the left.

Medications
He is on Lorcet, Soma, naproxen, Xanax.

Allergies: Codeine, Penicillin

Lab Results
X-rays were obtained on this visit. These were an AP and lateral of his lumbar spine. These show the old fusion of L4-5.

References
Processing of Medical Records from Unstructured Medical Transcripts


Author’s Biography

Vinod Chandra S S presently worked as a faculty member in Department of Computer Science & Engineering, College of Engineering Thiruvananthapuram, Kerala. He had his BTech (Computer Science & Engineering) from University of Calicut, Kerala and M Tech (Software Engineering) from Cochin University of Science and Engineering, Kerala. Presently he is doing PhD in Computational Biology in University of Kerala. Since 1999, he has taught in various Engineering Colleges. He has a modest number of research publications in National and International levels including journals such as ACM, IEEE Conference Proceedings, IJCAI Proceedings and Journal of Computer Society of India. He is a member of IEEE, Computer Society of India, Indian Society for Technical Education, and Institution of Engineers.