International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 1, ISSUE 4

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2008 Volume 1, Issue 4
Publishing Date: 31-12-2007
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 1, Issue 4, November/ December 2007.

Pages

Similarity-Based Estimation for Document Summarization using Fuzzy Sets

Masrah Azrifah Azmi Murad* masrah@fsktm.upm.edu.my
Department of Information Systems
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400, Serdang, MALAYSIA

Trevor Martin trevor.martin@bris.ac.uk
Department of Engineering Mathematics
University of Bristol
BS8 1TR, UK

Abstract

Information is increasing every day and thousands of documents are produced and made available in the Internet. The amount of information available in documents exceeds our capacity to read them. We need access to the right information without having to go through the whole document. Therefore, documents need to be compressed and produce an overview so that these documents can be utilized effectively. Thus, we propose a similarity model with topic similarity using fuzzy sets and probability theories to extract the most representative sentences. Sentences with high weights are extracted to form a summary. On average, our model (known as MySum) produces summaries that are 60% similar to the manually created summaries, while *tf.isf* algorithm produces summaries that are 30% similar. Two human summarizers, named P1 and P2, produce summaries that are 70% similar to each other using similar sets of documents obtained from TREC.

Keywords: fuzzy sets, mass assignment, asymmetric word similarity, topic similarity, summarization

1. INTRODUCTION

Information is increasing every day and thousands of documents are produced and made available in the Internet. The amount of information available in documents exceeds our capacity to read them. We need access to the right information without having to go through the whole document. Therefore, documents need to be compressed and produce an overview so that these documents can be utilized effectively. To generate a summary, we need to identify the most important information in the document avoiding the irrelevant or less important ones, [1] discussed three phases involved in summarizing text automatically: 1) selection of more salient information, 2) aggregation of information, and 3) generalization of specific information to a more general concept.

Recently, researchers have tried to extract a summary using various techniques such as word frequencies [2; 3; 4] and clustering [5]. The first automated sentence extraction system [6] uses term frequencies to weight sentences, which are then extracted to form an abstract. Since then,

* Corresponding author

many approaches have been explored in automatically extracting sentences from a document. Most existing text summarization systems use an extract approach. This approach is known to be safe because important information is taken or copied from the source text. An abstract approach produces a summary at least some of whose material is not available in the source text, but at the same time retaining the content originality. A summary can be generic meaning the content is broad and not addressed to any specific audience. Nonetheless, it could be tailored or user-focused in which the content is addressed to a group with specific interests. Further, the content of a summary can be indicative or informative. Indicative content provides an indication of the main topics. Therefore, it helps the user to decide whether to proceed with reading the source text or otherwise, while informative content represents the original document.

The objective of this work is to produce a similarity model with topic similarity using the theories of probability and fuzzy sets incorporating mass assignment to find the similarity between two words. We compute frequencies of triples of words exist in the document collection and convert these frequencies to fuzzy sets. Probability of two words is then computed using the semantic unification of two fuzzy sets. Our results show that using asymmetric word similarity with topic similarity able to extract the most relevant sentences and produce summaries that are almost similar to manually created summaries. The remainder of the paper is organized as follows: In section 2 we discuss briefly on common algorithm, *tf.isf*, and use it to benchmark against our algorithm, section 3 explains the methodology used, section 4 discusses in detail on the word similarity algorithm, section 5 discusses the similarity model in extracting sentences, section 6 discusses the results, and finally section 7 concludes the paper.

2. *tf.isf*

This section outlines related work done in summarization particularly extracting sentences from a document. We described a method known as *tf.isf* (term frequency x inverse sentence frequency) [4]. This method is used later in comparing against MySum (our proposed model) and a manually created summary. *tf.isf* is an adaptation of the conventional *tf.idf* [7]. Sentences are extracted using average sentence similarities and those with high weights (above a certain threshold) are extracted to form a summary. The computation of *tf.isf* is similar to the computation of *tf.idf*.

$$w_{ik} = tfi_k \; x \; idf_{ik} \tag{1}$$

The only difference is the notion of *document* that is being replaced by *sentence*. Each sentence is represented as a vector of *tf.isf* weights. Sentences with high values of *tf.isf* are selected to produce a summary of the source text. Hence, the *tf.isf* measure of a word *w* in a sentence *s*, is computed using the following

$$tf.isf(w,s) = tf\,(w,s) \times isf(w) \tag{2}$$

where *tf(w, s)* is the number of times the word *w* occurs in sentence *s*. *isf(w)* is the inverse sentence frequency of word *w* in sentence *s* given by

$$isf(w) = log\; S \;/\; sf(w) \tag{3}$$

where *sf(w)* is the number of sentences in which the word *w* occurs and *S* is the total number of sentences in the document. For each sentence *s*, the average *tf.isf* of the sentence is computed by calculating the average of the *tf.isf(w, s)* weight over all of the words *w* in the sentence, as shown in the following formula

$$\sum_{i=1}^{W(s)} tf.isf\,(i,s) \Big/ W(s) \tag{4}$$

where *W(s)* is the number of words in the sentence *s*. Sentences with the largest values of average *tf.isf* are selected as the most relevant sentences and will be produced as a summary. Using *tf.isf* is simple and fast. Further, *tf.isf* only relies on the frequency of words in documents, therefore, it's possible to use *tf.isf* in summarizing texts other than English. However, *tf.isf* may not be a good algorithm in extracting sentences. For example, *tf.isf* cannot reflect similarity of words and only count the number of overlapping words. The algorithm does not consider any synonymy and syntactic information. In addition, there could be some relevant or important sentences missing, as they use different words to express the same interests.

3. FUZZY SETS AND MASS ASSIGNMENT

This section outlines the theory of fuzzy sets and mass assignment that are used extensively in our work. A fuzzy set is an extension to a classical set theory, which has a problem of defining the border of the set and non-set [8]. Unlike a classical set, a fuzzy set does not have a clearly defined boundary by having elements with only a partial degree of membership [9]. For example, consider a weight of a person with labels such as *thin, average,* and *fat*. These labels are considered fuzzy because not everyone will agree with the same subset of the value domain as satisfying a given label. Nevertheless, if everyone agrees, we could write precise definitions of *thin, average*, and *fat* in this context.

A mass assignment theory was proposed by Baldwin in 1991 as a general theory for evidential reasoning under uncertainty [9; 10]. This theory is used to provide a formal framework for manipulating both probabilistic and fuzzy uncertainties [9]. Consider the following example taken from [10], suppose we have a set of people labeled 1 to 10 who are asked to accept or reject a dice value of *x* as *small*. Suppose everyone accepts 1 as *small*, 80% accept 2 as *small* and 20% accept 3 as *small*. Therefore, the fuzzy set for *small* is defined as

$$small = 1 \,/\, 1 + 2 \,/\, 0.8 + 3 \,/\, 0.2 \tag{5}$$

where the membership value for a given element is the proportion of people who accept this element as satisfying the fuzzy set. The probability mass on the sets is calculated by subtracting one membership from the next, giving MA_{small} as

$$MA_{small} = \{1\} : 0.2, \{1, 2\} : 0.6, \{1, 2, 3\} : 0.2 \tag{6}$$

The mass assignments above correspond to families of distribution. In order to get a single distribution, the masses are distributed evenly between elements in a set. This distribution is known as *least prejudiced distribution (LPD)* [11] since it is unbiased towards any of the elements. Thus, in the example above, the mass of 0.6 is distributed equally among 1 and 2 and the mass 0.2 is distributed equally among 1, 2 and 3. Therefore, the *least prejudiced distribution* for *small* is

$$\begin{aligned} LPD_{small} = \; & 1 : 0.2+0.3+0.0667=0.5667, \\ & 2 : 0.3+0.0667=0.3667, \\ & 3 : 0.0667 \end{aligned} \tag{7}$$

3.1 Semantic Unification

Semantic Unification is a concept in Fril [12] proposed by Baldwin in 1992 that is used to unify vague terms by finding a support for the conditional probability of the match. Unification is possible if two terms have the same meaning, however, if they only have similar meaning, then the match will not be perfect and can be supported with a support pair. A mass assignment with the least prejudiced distribution is used to determine the unification of two fuzzy sets. For example, suppose the fuzzy set for *medium* in the voting model is

$$medium = 2 / 0.2 + 3 / 1 + 4 / 1 + 5 / 0.2 \quad (8)$$

and the mass assignment would be

$$MA_{medium} = \{3, 4\} : 0.8, \{2, 3, 4, 5\} : 0.2 \quad (9)$$

Thus, the least prejudiced distribution is

$$LPD_{medium} = 2 : 0.05, 3 : 0.45, 4 : 0.45, 5 : 0.05 \quad (10)$$

Suppose we want to determine the *Pr (about_3 | medium),* and the fuzzy set is

$$about\_3 = 2 / 0.4 + 3 / 1 + 4 / 0.4 \quad (11)$$

with mass assignment as

$$MA_{about\_3} = \{3\} : 0.6, \{2, 3, 4\} : 0.4 \quad (12)$$

We use the point semantic unification algorithm [11] to determine the conditional probability. Thus,

Pr (dice is **about_3** | dice is **medium**)
= 0.6Pr (dice is 3 | dice is **medium**) + 0.4Pr (dice is {2, 3, 4} | dice is **medium**)
= 0.6 (0.45) + 0.4(0.05 + 0.45 + 0.45)
= 0.65

The point semantic unification can be calculated using the following tableau.

	0.8 : {3,4}	**0.2 : {2,3,4,5}**
0.6 : {3}	1/2 x 0.8 x 0.6	1/4 x 0.2 x 0.6
0.4 : {2,3,4}	0.8 x 0.4	3/4 x 0.2 x 0.4

TABLE 1: Tabular Form of the Pr (about_3 | medium).

The entries in the cells are the supports from the individual terms of the mass assignments. Each entry has an associated probability. Thus, the *Pr (about_3 | medium)* is 0.65. The computation of the probability above can be shown using the following formula. Consider two fuzzy sets *f1* and *f2* defined on a discrete universe *X*. Let

$\mu_{f1}(x)$ be the membership of element *x* in the fuzzy set *f1*.
$MA_{f1}(S)$ be the mass associated with set *S*.
$LPD_{f1}(x)$ be the probability associated with element *x* in the *LPD*.

(and similarly for *f2*). Therefore

$$Pr(f1 \mid f2) = \sum_{\substack{S1 \subseteq X, \\ S2 \subseteq X, \\ S1 \cap S2 \neq \phi}} \frac{MA_{f1}(S1) \times MA_{f2}(S2)}{|S2|}$$

$$= \sum_{x \in X} \mu_{f1}(x) \times LPD_{f2}(x) \quad (13)$$

4. ASYMMETRIC WORD SIMILARITY

In this section, we propose a novel algorithm in computing word similarities asymmetrically using mass assignment based on fuzzy sets of words. We concentrate on how sentences use a word, and not on their meaning. Words in documents are considered to be similar if they appear in similar contexts. Therefore, these similar words do not have to be synonyms or belong to the same lexical category. Further, this algorithm is incremental such that any addition or subtraction of words (and documents) will only require minor re-computation.

4.1 Document Preprocessing and Similarity Algorithm

Before the measurement of the similarity algorithm is implemented, documents need to go through preprocessing stage so that only meaningful keywords are obtained from those documents. The first step is to remove common words, for example, *a, the, or,* and *all* using a list of stop words. If a word in a document matches a word in the list, then the word will not be included as part of the query processing. The second step is to stem a word to become a root word, for example, *subtraction* becomes *subtract.* In this work, we applied the process of Porter stemmer [13] to every word in the document.

The underlying objective of our method is the automatic computation of similar words. The method is based on the observation that it is frequently possible to guess the meaning of an unknown word from its context. The method assumes that similar words appear in similar contexts and therefore, these words do not have to be synonyms or belong to the same lexical category. A key feature of the algorithm is that it is incremental, i.e. words and documents can be added or subtracted without extensive re-computation. Our method is based on finding the frequencies of n-tuples of context words in a set of documents where frequencies are converted to fuzzy sets, which represent a family of distributions, and find their conditional probabilities. Consider the following example, taken from [14]

A bottle of *tezgüno* is on the table.
Everyone likes *tezgüno.*
Tezgüno makes you drunk.
We make *tezgüno* out of corn.

From the sentences above, we could infer that *tezgüno* may be a kind of an alcoholic beverage. This is because other alcoholic beverages, for example, *beer* tends to occur in the same contexts as *tezgüno*. The idea that words occurring in documents in similar contexts tend to have similar meanings is based on a principle known as the Distributional Hypothesis [15]. We use this idea to produce a set of related words, which can be used as the basis for taxonomy, or to cluster documents. In this experiment, we use Fril to compute asymmetric similarities such that the similarity between <w1> and <w2> is not necessarily the same as between <w2> and <w1> expressed as

$$ws(<w1>,<w2>) \neq ws(<w2>,<w1>)$$

This is because to compute similarity between two fuzzy sets, i.e. ws(<w1>,<w2>), we multiply the memberships of fuzzy sets of <w1> with the corresponding frequencies in frequency distributions of <w2>. In order to calculate ws(<w2>,<w1>), we multiply the memberships of fuzzy sets of <w2> with the corresponding frequencies in frequency distributions of <w1>. In most cases, the values for two fuzzy sets are different; therefore, the similarity measures will be different. In the next phases, we present the algorithms used in finding the similarity between words. AWS consists of two phases. In Phase I [16], we compute the frequency distributions of words to fuzzy sets. In Phase II [16], we find the conditional probabilities of the fuzzy sets using the semantic unification algorithm and show the creation of AWS matrix.

Phase I – Computation of frequency distributions to fuzzy sets

Each document is described by a set of all words called vocabulary. We run a pre-processing procedure by removing inappropriate words and stemming words. Removing inappropriate words allow us to save space for storing document contents and at the same time reduce the time taken during the search process. We define a document D_j that is represented by a set of an ordered sequence of n_j words as the following

$$D_j = \{w_0, w_1, w_2, \ldots, w_{nj}\}$$

with w being the sub-sequence of document D_j. The ordering of words in the document is preserved. We calculate the frequency distributions of every word available in the document. For any sub-sequence $W_n(x) = \{w_x, w_{x+1}, \ldots, w_{x+n}\}$, let $p(x)$ be a word that precedes word x such that

$$p(x) = \{w_{x-k}, w_{x-k+1}, \ldots, w_{x-1}\}$$

and $s(x)$ be a word that succeeds word x such that

$$s(x) = \{w_{x+l+1}, w_{x+l+2} \ldots, w_{x+l+k}\}$$

where k and l are a given block of k words preceded and succeeded by blocks of l words, and n is the total number of words in the document. We give a value of 1 to k and l as we need to consider the start and end of the document. Consider a document D_j containing sentences as the following.

The quick brown fox jumps over the lazy dog. The quick brown cat jumps onto the active dog. The slow brown fox jumps onto the quick brown cat. The quick brown cat leaps over the quick brown fox.

TABLE 2: Example of Sentences in Document D_j

From the sentences, we obtain

W(1)=quick, p(1)=the, s(1)=brown
W(2)=brown, p(2)=quick, s(2)=fox

using

p(x) = W(x-1)
s(x) = W(x+1)

The computation of frequency distributions of words in the document will be built up incrementally. Hence, for each word x, we incrementally build up a set <context-of-x> containing pairs of words that surround x, with a corresponding frequency. Let

pre(x) be the set of words that precedes word x
suc(x) be the set of words that succeeds word x
$N = \{pre(x), x, suc(x)\}$ being the total number of times the sequence of *{pre(x), x, suc(x)}* occurs in document D_j

Thus, the frequency of each <context-of-x> is given by the following

$$f_{cw} = \{pre(x), x, suc(x)\} / N$$

Once we computed the frequency distributions of each word, we convert the frequencies to memberships as shown in the following algorithm.

Input:

f_{cw} :	**array of frequency counts.**
T :	**total frequency count for this word =** $\sum_{P,S} f_{cw}(P,S)$ **where P and S are precedence and successor respectively.**

Output:

m_{cw}:	**array of memberships**
1.	**Sort frequency counts into decreasing order, $f_{cw}[0]$ ... $f_{cw}[n-1]$ such that $f_{cwi} \geq f_{cwj}$ iff i > j**
2.	**Set the membership corresponding to maximum count, $m_{cw}[0] = 1$**
3.	**for i=1 ... n-1, i.e., for each remaining frequency count $m_{cw}[i] = m_{cw}[i-1] - (f_{cw}[i-1] - f_{cw}[i]) * i / T$**

FIGURE 1: Algorithm for Converting Frequencies to Memberships

The complexity of the above algorithm lies in its sorting step, nevertheless, the remaining steps are linear in the size of the array. Using the example of sentences in Table 2, we obtain the frequencies for word *brown* with *N=6*

quick - brown - cat occurs three times
quick - brown - fox occurs two times
slow - brown - fox occurs once

We use mass assignment theory to convert these frequencies to fuzzy sets (as described in Figure 1), and obtain the fuzzy set for word *brown* as

(quick, cat):1, (quick, fox):0.833, (slow, fox):0.5

In the next phase, we use the fuzzy sets to compute the probability of any two words.

Phase II – Computation of Word Probabilities

To compute a point semantic unification for two frequency distributions f_{cw1} and f_{cw2}, we calculate membership for f_{cw1} and multiply by the frequency for the corresponding element in f_{cw2}.

Input:

m_{cw1} :	**array of memberships.**
f_{cw2} :	**array of frequency counts.**
T_{cw2} :	**total frequency counts for w2 =** $\sum_{P,S} f_{cw2}(P,S)$ **where P and S are precedence and successor respectively.**

Output:

Semantic Unification Value - Pr (w1|w2) , Pr (w2|w1)

1.	**Convert f_{cw1} to m_{cw1} using steps in Algorithm I.**
2.	**Calculate the sum of m_{cw1} multiply by f_{cw2} for the common elements giving the point semantic unification for two frequency distributions.**
3.	**To compute the asymmetric probability, simply reverse the calculation in steps 1 and 2.**

FIGURE 2: Point Semantic Unification Algorithm

Hence, for any two words <w1> and <w2>, the value

Pr (<context-of-w1>|<context-of-w2>)

measures the degree to which <w1> could replace <w2>, and is calculated by semantic unification of the two fuzzy sets characterizing their contexts. For example, suppose there is sentences in the document that give the fuzzy context set of *grey* as

(quick, cat):1, (slow, fox):0.75

We calculate the asymmetric word similarity of the two fuzzy sets of *brown* and *grey* using point semantic unification algorithm, giving the conditional probabilities as

Pr (brown | grey) = 0.8125
Pr (grey | brown) = 0.625

By semantic unification of the fuzzy context sets of each pair, we obtain an asymmetric word similarity matrix. For any word, we can extract a fuzzy set of similar words from a row of the matrix. We also note that there are important efficiency considerations in making this a totally incremental process, i.e. words (and documents) can be added or subtracted without having to recalculate the whole matrix of values as opposed to a straightforward implementation that requires $O(n_a \times n_b)$ operations per semantic unification, where n_b is the cardinality of the fuzzy context set that requires $O(n^2)$ semantic unification and *n* is the size of the vocabulary. Therefore, any addition of a new word or a new document using a straightforward implementation would require the whole re-computation of the matrix. Figure 3 below shows the creation of AWS matrix with elements described in the algorithm as having non-zero values.

1.	**Store each word with a list of its context pairs with number of times each context pair has been observed.**
2.	**Calculation of the corresponding memberships and elements are not done until needed. Otherwise,** **if a word *W* is read, then mark elements *Pr(W|w_i)* and *Pr(w_i|W)* as needing recalculation.**
3.	**If a new context, *P-W-S* is read,** **search for other words w_j which have the same context *P-w_j-S*.** **mark the elements *Pr(W|w_j)* and *Pr(w_j|W)* as needing calculation.**

FIGURE 3: Algorithm for Creating AWS Matrix

This process creates an asymmetric word similarity matrix *Sim,* whose rows and columns are labeled by all the words encountered in the document collection. Each cell *Sim(w_i, w_j)* holds a value between 0 and 1, indicating to which extent a word *i* is contextually similar to word *j*. For any word we can extract a fuzzy set of similar words from a row of the matrix. Many of the elements are zero. As would be expected, this process gives both sense and nonsense. Related words appear in the same context (as with *brown* and *grey* in the illustration above), however, unrelated words may also appear, for example, the phrase *{slow fat fox}* would lead to a non-zero similarity between *fat* and *brown.*

5. THE SIMILARITY MODEL

Recall the AWS algorithm we have described in Section 4 above

$$Sim(w_i, w_j) \tag{14}$$

We now introduce the sentence similarity measures *sim*(S_i, S_j) to find the similarities between sentences available in a document using AWS. Hence

$$\sum_{i \in sentence1} \sum_{j \in sentence2} f_i Sim(w_i, w_j) f_j \tag{15}$$

where *f* is the relative frequency of a word in a sentence and *Sim*(w_i,w_j) is the similarity matrix developed in Section 4. We also compute the asymmetric sentence similarity, which would produce a different similarity measure. We introduce a topic similarity measure *sim*(S_i, *t*) for the purpose of increasing the importance measure of a sentence *Si* to the topic *t*. We compute a weight for topic similarity using the frequency of overlapping words in the sentence as well as the topic. Identical words will have a value of 1, with 0 for non-identical words. Hence, the formula is defined as

$$\sum_{i \in sentence} \sum_{j \in topic} f_i ws(w_i, w_j) f_j \tag{16}$$

where *f* is the relative frequency of a word in a sentence and topic respectively and *ws*(w_i,w_j) is the similarity of overlapping words. We named the two similarity measures above (as in Eq. 15 and 16) as the two score functions and these score functions will be used in extracting sentences from the document.

Sentence Extraction

The two score functions, i.e. sentence similarity and topic similarity measures are used to compute the weight for each sentence. We measure the importance of a sentence S_i as an average similarity *AvgSim(i)*. The weight of a sentence is defined by summing similarity measure of sentence S_i with other sentences in the document divided by *N* the total number of sentences. Thus, the *AvgSim(i)* is defined as

$$\frac{\sum sim(S_i, S_j)}{N} \tag{17}$$

where *sim*(S_i, S_j) is the pairwise asymmetric sentence similarity. Next, we add the weight of average sentence similarity and topic similarity to produce the final score of a sentence, given in the following formula

$$MySum = AvgSim(i) + sim(S_i, t) \tag{18}$$

Once the score for each sentence has been computed, the sentences are ranked in descending order. Sentences with high values will be selected to produce a summary. Then the sentences are arranged according to their chronological order in the original article to form a summary.

6. RESULTS

In order to evaluate the effectiveness of our method, we compare the summaries produced by our system against the manually created summaries produced in the DUC 2002 [17]. In addition, we also compare the performance of other system, *tf.isf* against the manually created summaries. Our final comparison is between MySum and *tf.isf* against the manually created summaries. Each document of DUC 2002 produces two versions of manually created summaries written by two different human readers. In this experiment, we produce a hypothetical test by making a comparison of summaries produced by two different human summarizers. This is to show that in reality it is very unlikely for two different systems or humans to produce an identical summary from a document.

In DUC 2002, Task 1 is a single-document summarization in which the goal is to extract 100 word summaries from each document in the corpus. We use the summaries produced in Task 1 in our comparison stage. The comparison is made using individual matching, i.e. each sentence in a summary produced by MySum or *tf.isf* is compared against each sentence of the manually created summary. In this case, a sentence generated by MySum or *tf.isf* and a sentence from manually created summary are considered similar if the similarity is equal to the proportion of identical words. If all sentences produced by MySum or *tf.isf* are the same as the sentences in the manually created summary, the similarity measure is equal to 1. However, if only a proportion of sentences are equal to the sentences in the manually created summary, the similarity measure would be the number of sentences generated by MySum or *tf.isf* that is similar to the number of sentences in the manual summary divided by total number of sentences in the manually created summary. In this paper, we presented only a few of our results, while the remaining is reported elsewhere. Figures 4 to 6 show the comparison of similarity produced by MySum and *tf.isf* against human summarizers, P1 and P2. In the figures, the comparisons of two summaries produced by P1 and P2 are used as a hypothetical test.

On average, MySum produces summaries that are 60% similar to the manually created summaries, while *tf.isf* produces summaries that are 30% similar. It is worth pointing out that the human summarizers, P1 and P2 produce summaries that are 70% similar to each other. Overall, MySum produces a fairly good result and none of the documents generated by MySum produce a zero similarity comparison against the manually created summaries. Our method shows that it could generate a summary from a document as close to what a human summarizer could produce.

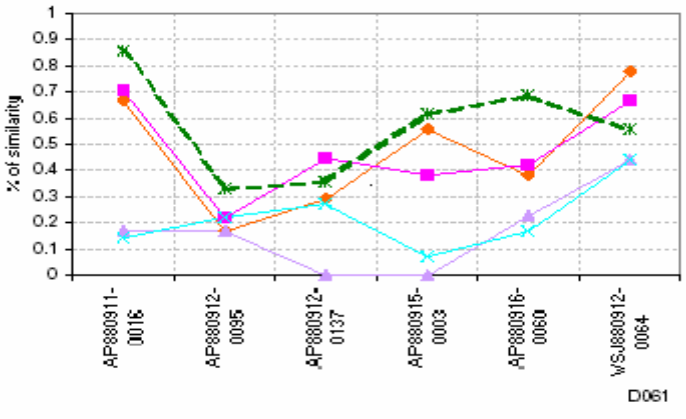


FIGURE 4: Result on Summarization using Document Set D061

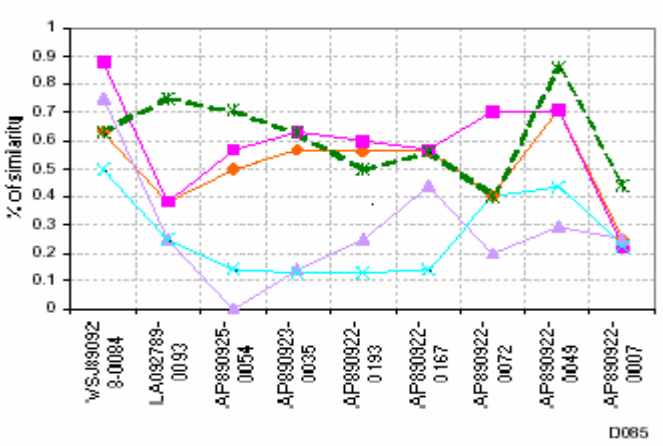


FIGURE 5: Result on Summarization using Document Set D085

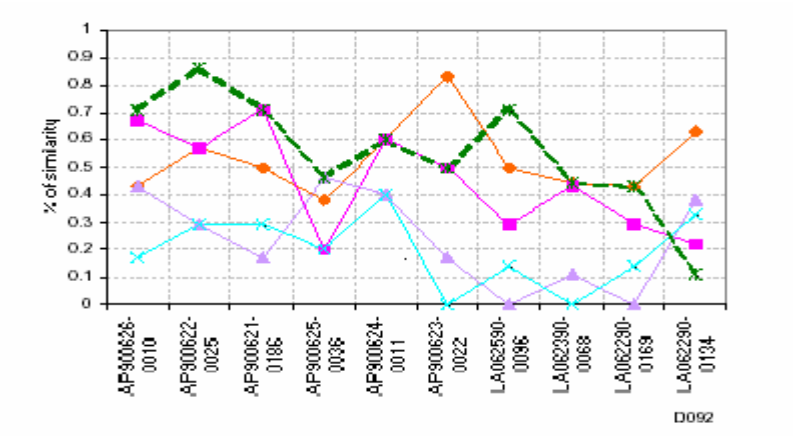
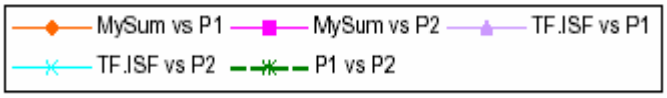


FIGURE 6: Result on Summarization using Document Set D092



7. CONCLUSION AND FUTURE WORK

This paper presented a detailed algorithm in computing the asymmetric similarity between words using fuzzy context sets and topic similarity in extracting the most relevant sentences. The asymmetric word similarity measure words that appear in similar context in the sentences, while topic similarity compute the frequency of overlapping words appear in the sentence and topic. Experiments show that using the combination of both the word similarity and topic similarity able to extract the most important sentences from a document that is fairly close to the manually created summaries. Although MySum did not produce an exact summary to the one created by human, on average, MySum is able to give a representable extractive summary. The difference between MySum and human summarizers in producing summary is only 10 percent. On the other hand, MySum outperforms *tf.isf* when compared against the manually created summaries. In future, we hope to test MySum for multi-document summarization. This work can also be extended in looking at abstract summarization or how to combine similar sentences together as how a human summarizer would do.

8. REFERENCES

1. K. Sparck Jones. *"Automatic Summarizing: Factors and Directions"*. In I. Mani and M.T. Maybury, Editors, Advances in Automatic Text Summarization, Cambridge, MA: The MIT Press, pp 1-12, 1999

2. S.H. Lo, H. Meng, and W. Lam. *"Automatic Bilingual Text Document Summarization"*. In Proceedings of the Sixth World Multiconference on Systematic, Cybernetics and Informatics. Orlando, Florida, USA, 2002

3. S. Yohei ''*Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles (TSC-8)*'' NTCIR Workshop 3 Meeting TSC, pp 55-59, 2002

4. J. Larocca Neto, A.D. Santos, C.A.A. Kaestner, and A.A. Freitas. "*Document Clustering and Text Summarization*". In Proceedings of the 4th Int. Conf. Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), London: The Practical Application Company, pp 41---55, 2000b

5. M. Amini and P. Gallinari. *"The Use of Unlabeled Data to Improve Supervised Learning for Unsupervised for Text Summarization".* In SIGIR, Tampere, Finland, 2002

6. H. Luhn *"The Automatic Creation of Literature Abstracts"*. IBM Journal of Research and Development, 2(92):159 - 165, 1958

7. G. Salton and C. Buckley. "*Term-weighting Approaches in Automatic Text Retrieval".* Information Processing and Management 24, pp 513-523, 1988. Reprinted in: Sparck Jones K. and Willet P. (eds). Readings in Information Retrieval, Morgan Kaufmann, pp 323-328, 1997

8. G.J. Klir and B. Yuan. "*Fuzzy Sets and Fuzzy Logic - Theory and Applications".* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1995

9. J.F. Baldwin. *"Fuzzy and Probabilistic Uncertainties".* In Encyclopedia of AI, 2nd ed., S.C. Shapiro, Editor 1992, Wiley, New York, pp. 528-537, 1992

10. J.F. Baldwin. *"Combining Evidences for Evidential Reasoning".* International Journal of Intelligent Systems, 6(6), pp. 569-616, 1991a

11. J.F. Baldwin, J. Lawry, and T.P. Martin. *"A Mass Assignment Theory of the Probability of Fuzzy Events"*. Fuzzy Sets and Systems, (83), pp. 353-367, 1996

12. J.F. Baldwin, T.P. Martin and B.W. Pilsworth. "*Fril - Fuzzy and Evidential Reasoning in Artificial Intelligence".* Research Studies Press Ltd, England, 1995

13. M.F. Porter. *"An Algorithm for Suffix Stripping".* Program, 14(3):130-137, 1980

14. D. Lin. *"Extracting Collocations from Text Corpora"*. Workshop on Computational Terminology, Montreal, Canada, 1998

15. Z. Harris. *"Distributional Structure".* In: Katz, J. J. (ed.) The Philosophy of Linguistics. New York: Oxford University Press, pp. 26-47, 1985

16. M.A. Azmi-Murad. "*Fuzzy Text Mining for Intelligent Information Retrieval".* PhD Thesis, University of Bristol, April 2005

17. DUC. *"Document Understanding Conferences"*. http://duc.nist.gov, 2002

Utilizing AOU'VLE with other Computerized Systems

Bayan Abu-Shawar **b_shawar@aou.edu.jo**
Information Technology and Computing
Arab Open University
Amman, P.O.Box 1339 Amman 11953, Jordan

Abstract

We present in this paper our experience of utilizing the virtual learning environment system with other computerized systems. Arab Open University is one of the first organizations that adopt an e-learning methodology in the Arabic region. We present Moodle as a virtual learning environment (VLM) used at AOU. The integration process of VLE with other online systems such as student information system (SIS) and human resource system (HRS) is discussed. In addition to that we describe the in-house development and enhancement generated to the VLE to cope with AOU regulations and rules. The quality assurance strategy of AOU is clarified.

Keywords: e-learning, open learning, LMS, SIS.

1. INTRODUCTION

The growth of Internet-based technology have brought new opportunities and methodologies in many fields including education and teaching represent in e-learning, online learning, distance learning, and open learning. These approaches are typically use in place of traditional methods and mean that students deliver their knowledge though the web rather than face-to-face tutoring.

Researchers and practitioners were divided into two camps when the concept of distance learning was proposed. Some believed that online and distance learning will reduce the quality of education based on the absence of face-to-face relationships between students and their tutors, and between the student themselves [1]; [2]; [6]. Others supported using Internet-based education, and proved the effectiveness of it by applying both methods in parallel on some courses and comparing student's results, which were nearly equivalent [19]; [5].

In the same respect, many studies address the challenges of distance learning to be accepted in the education community [3]. Johnson et al. [4] claim that "the primary among these challenges is how to meet the expectations and needs of both instructor and the student and how to design online courses so they provide a satisfying and effective learning environment". Owston [7] agrees that "the key to promoting improved learning with the web appears to lie in how effectively the medium is exploited in the teaching and learning situation".

E-learning is a new trend of education system, where students deliver their materials through the web. E-learning is the "use of internet technology for the creation, management, making available, security, selection and use of educational content to store information about those who learn and to monitor those who learn, and to make communication and cooperation possible." [8].

Kevin kruse [17] addressed the benefits of e-learning for both parties: organization and learners. Advantages of organizers are reducing the cost in terms of money and time. The money cost is reduced

by saving the instructor salaries, and meeting room rentals. The reduction of time spent away from the job by employees may be most positive shot. Learning time reduced as well, the retention is increased, and the contents are delivered consistently. On another hand, learners are able to find the materials online regardless of the time and the place; it reduces the stress for slow or quick learners and increases users' satisfaction; increases learners' confidence; and more encourages students' participations.

In this paper the e-learning platform of the AOU is described in section 2. The integration process between virtual learning environment and other computerized system is presented in section 3. Section 4 discussed the requirements and the strategies of quality assurance unit at AOU. Finally, section 5 concludes this paper.

2. The e-learning platform of the AOU

Arab Open University was established in 2002 in the Arabic region, and adopted the open learning approach. An open learning system is defined as "a program offering access to individuals without the traditional constraints related to location, timetabling, entry qualifications."[12].

The aim of AOU is to attract large number of students who can not attend traditional universities because of work, age, financial reasons and other circumstances. The "open" terminology in this context means the freedom from many restrictions or constraints imposed by regular higher education institutions which include the time, space and content delivery methods.

Freed et al. [9] claimed that the "interaction between instructors and students and students to students remained as the biggest barrier to the success of educational media". The amount of interaction plays a great role in course effectiveness [10]. For this purpose and to reduce the gap between distance learning and regular learning, the AOU requires student to attend weekly tutorials. Some may argue that it is not open in this sense; however the amount of attendance is relatively low in comparison with regular institutions. For example, 3 hours modules which require 48 hours attendance in regular universities, is reduced to 12 hours attendance in the AOU.

In order to give a better service to students and tutor, to facilitate accessing the required material from anywhere, and to facilitate the communication between them, an e-learning plat form is needed. A learning platform “is software or a combination of software that sits on or is accessible from a network, which supports teaching and learning for practitioners and learners.”[18]. A learning platform is considered as a common interface to store and access the prepared materials; to build and deliver learning activities such quizzes and home-works; support distance learning and provide a set of communication possibilities such as timetables, videos, etc.

AOU has partnerships with the United Kingdom Open University (UKOU) and according to that at the beginning the AOU used the FirstClass system as a computer mediated communication (CMC) tool to achieve a good quality of interaction. The FirstClass tool provides emails, chat, newsgroups and conferences as possible mediums of communication between tutors, tutors and their students, and finally between students themselves. The most important reason behind using FirstClass was the tutor marked assignment (TMA) handling services it provided. However, the main servers are located in the UKOU which influences the control process, causes delays, and totally depends on the support in UKOU for batch feeds to the FirstClass system [11].

To overcome these problems, AOU use Moodle nowadays as an electronic platform. Moodle is an open-source course management system (CMS) used by educational institutes, business, and even individual instructors to add web technology to their courses. A course management system is "often internet-based, software allowing instructors to manage materials distribution, assignments, communications and other aspects of instructions for their courses." [13] CMS's, which are also known as virtual learning environments (VLE) or virtual learning environments (VLE), are web applications, meaning they run on a server and are accessed by using a web browser. Both students and tutors can access the system from

anywhere with an Internet connection. The Moodle community has been critical in the success of the system. With so many global users, there is always someone who can answer a question or give advice. At the same time, the Moodle developers and users work together to ensure quality, add new ,modules and features, and suggest new ideas for development [14, 15]. Moodle also stacks up well against the feature sets of the major commercial systems, e.g., Blackboard and WebCT [16]. Moodle provides many learning tools and activities such as forums, chats, quizzes, surveys, gather and review assignments, and recording grades.

Moodle has been used in AOU mainly to design a well formed virtual learning environment which facilitates the interaction among all parties in the teaching process, students and tutors, and more over to integrate the VLE with the student information system (SIS) and the human resource system (HRS).

FIGURE 1: The unified image of the AOU e-learning systems

In addition that Moodle is easy to learn and use, and that it is popular with large user community and development bodies. Moodle is flexible in terms of:

- Multi-language interface,
- Customization (site, profiles),
- Separate group features, and pedagogy.

The unified image of the e-learning platform of the AOU from the starting web page shown in figure1, the users will be able to:

- Connect to the SIS, where they could do online registration, seeing their grades and averages as presented in figure2.
- Perform learning activities through the VLE, such as submitting assignments, do online quizzes, etc.
- Retrieve resources through AOU digital library subscriptions.

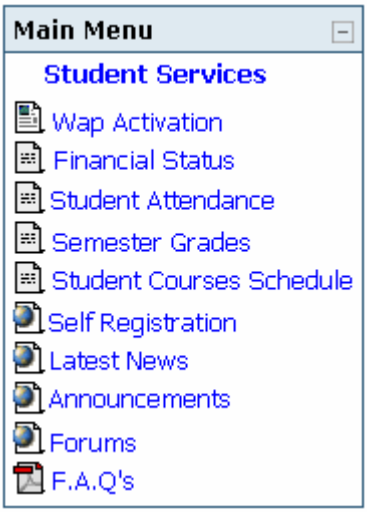


FIGURE 2: The SIS of the AOU

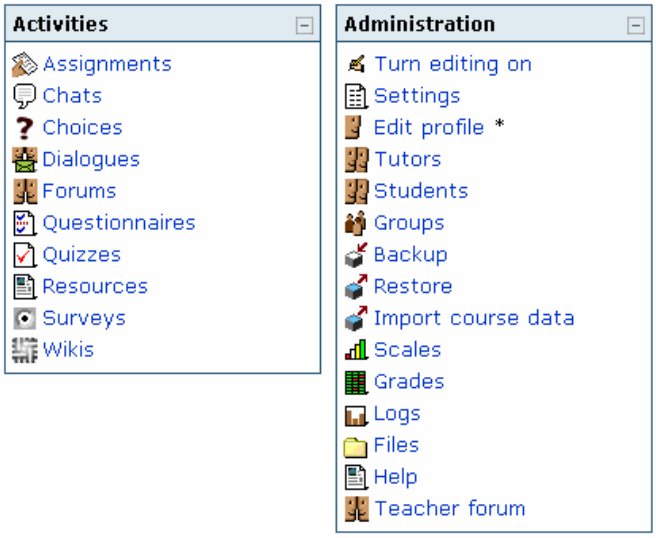


FIGURE 3: VLE course activities and administration

3. Integrating VLE with other computerized systems at AOU

The virtual learning environment (VLE) is software that automates the administration of training events. The term VLE is now used to describe a wide range of applications that track student training and may include functions to:

- Manage users logs, course catalogs, and activity reports
- Provide basic communication tools (email, chat, whiteboard, video conferencing)
- Manage competency (e-Tests, e-Assignments)
- Allow personalization (user profiles, custom news, recent activity, RSS)
- Enable monitoring activities (QA, accreditation, external assessment).

The usefulness of the VLE could be summarized as follows:

- Simplicity, easy creation and maintenance of courses.
- Reuse, support of existing content reuse.
- CMC, TMA, Tests, Progress, learner involvement.
- Security, secure authentication/authorization
- Administration, intuitive management features
- Technical support, active support groups
- Language, true multi-lingual
- Affordability, maintenance and annual charges.

AOU has many computerized systems that facilitate services to students and staff. In the following subsections we will discuss the integration process done on Virtual learning environment (VLE) with Student Information system (SIS), Human Resource System (HRS), and the enhancement needed to integrate such systems together.

3.1 Integrating VLE with SIS

The student information system (SIS) is an Oracle based program which provides the necessary information such as students' information, courses registered, faculties, grades, etc. VLE integration with SIS (or VLE-SIS) is a system used inside the university to reducing accessing time, automatically generating accounts, minimizing faults, mistakes and errors to null, obtaining availability of requirements and simplifying registering, entering and filling process as shown in figure 4.

Arab Open University contains multiple systems that were never designed to work together. The business units that fund these information systems are primarily concerned with functional requirements rather than technical architectures because information systems vary greatly in terms of technical architecture. Enterprises often have a mix of systems and these systems tend to have incompatible architectures. The SIS of AOU is organized into three logical layers: presentation, business logic, and data. When we integrate multiple systems, we usually want to be as non-invasive as possible. Any change to an existing production system is a risk, so it is wise to try to fulfill the needs of other systems and users while minimizing disturbance to the existing systems. The idea is to isolate internal structure of the SIS. Isolation means that changes to one of SIS's internal structures or business logic do not effect other applications like VLE. Without isolated data structures, a small change inside an application could cause a ripple effect and require changes in many dependent applications. Reading data from a system usually requires little or no business logic or validation. In these cases, it can be more efficient to access raw data that a business layer has not modified.

Many preexisting applications couple business and presentation logic so that the business logic is not accessible externally. In other cases, the business logic may be implemented in a specific programming language without support for remote access. Both scenarios limit the potential to connect to an application's business logic layer.

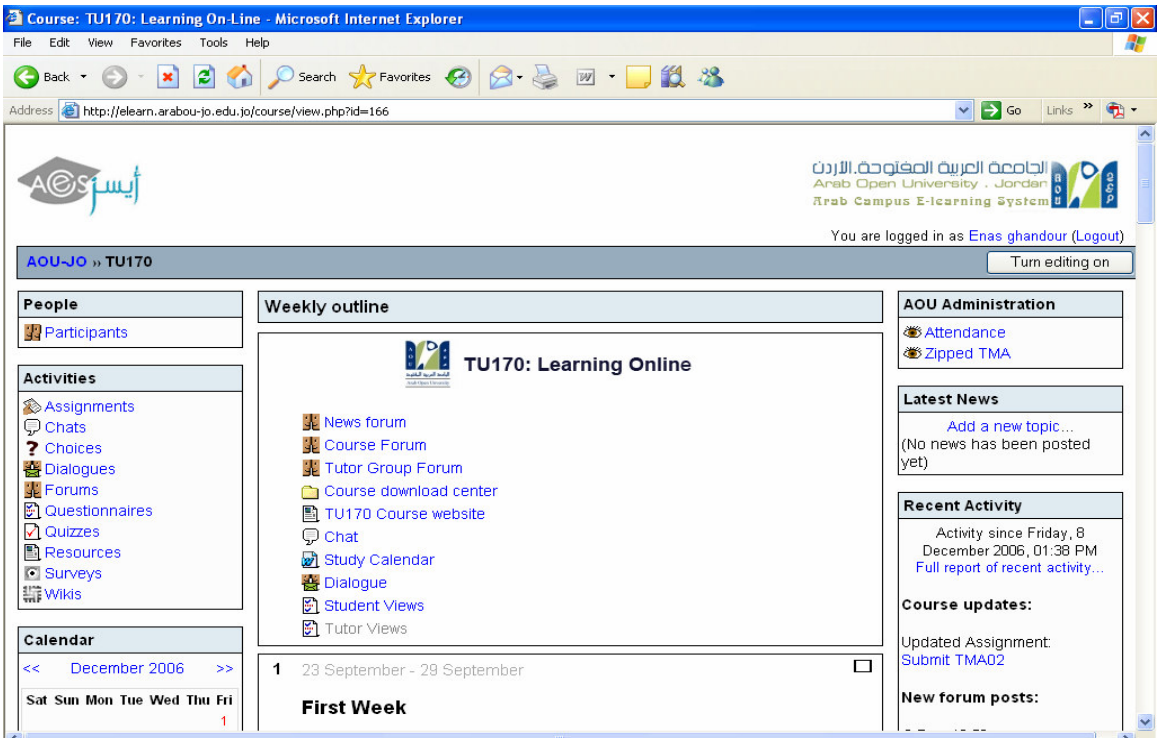


FIGURE 4: The VLE of AOU

When making updates to SIS's data, the advantages of its business logic is that it performs validation and data integrity checks, and this should be considered. The integration between SIS and VLE at the logical data layer is achieved by allowing the data in SIS to be accessed by VLE as shown in Figure 5.

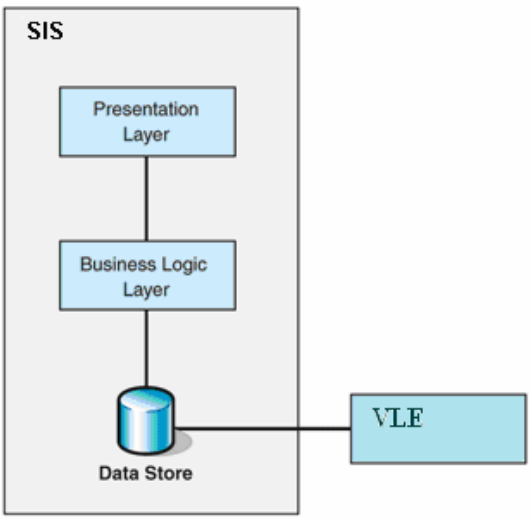


FIGURE 5: SIS-VLE integration

To connect SIS and VLE at the logical data layer, multiple copies of the database are generated instead of sharing a single instance of database between applications, so that each application has its own dedicated store. To keep these copies synchronized, data is copied from one data store to another. This

approach is common with packaged applications because it is not intrusive. However, it implies that at any time, the different data stores are slightly out of synchronization due to the latency that is inherent in propagating the changes from one data store to the next. The integration process added a lot of facilities which reduces time and cost in the following ways:

- Automatic structure enrollment: each student is provided with a username and password which enable students to register automatically.
- Automatic course enrollment: students are automatically enrolled into VLE courses they have been registered.
- Automatic group enrollment: students are automatically enrolled into VLE courses group, as they registered this group in the university.
- Automatically withdraw students from courses where students want to drop or have some financial problems.
- Student semester grades: students are enabled to see their grades through the VLE rather than bringing it from registrar.
- Students registered courses: where students could see the registered courses information such as their groups, time, course names and short names.
- Student's financial issues: where students could see their financial status and payment schedule.

The process is applied by establishing a secure connection to SIS with the minimum privileges, then acquiring data from SIS, after that manipulating it into the VLE as follows:

- Checking if a student exists, if not, register him/her and create a username and password.
- Enrolling students into their courses, and then enroll them into their groups.
- Checking if there is any change in courses or groups and setting data as it appears in SIS.
- Acquiring grades and schedule for students from SIS.
- Checking students with financial problems, and withdraw them from VLE.
- Enrolling new students into placement test, and update their results into SIS.

In addition to the previous automatic operations, other benefits were gained due to the integration process, which are:

- Non-intrusive. Most databases support transactional multi-user access, ensuring that one user's transaction does not affect another user's transaction. This is accomplished by using the Isolation property of the Atomicity, Consistency, Isolation, and Durability (ACID) properties set. In addition, many applications permit you to produce and consume files for the purpose of data exchange. This makes data integration a natural choice for packaged applications that are difficult to modify.

- High bandwidth. Direct database connections are designed to handle large volumes of data. Likewise, reading files is a very efficient operation. High bandwidth can be very useful if the integration needs to access multiple entities at the same time. For example, high bandwidth is useful when you want to create summary reports or to replicate information to a data warehouse.

- Access to raw data. In most cases, data that is presented to an end user is transformed for the specific purpose of user display. For example, code values may be translated into display names for ease of use. In many integration scenarios, access to the internal code values is more useful because the codes tend to more stable than the display values, especially in situations where the software is localized. Also, the data store usually contains internal keys that uniquely identify entities. These keys are critical for robust integration, but they often are not accessible from the business or user interface layers of an application.

- Metadata. Metadata is data that describes data. If the solution that you use for data integration connects to a commercial database, metadata is usually available through the same access mechanisms that are used to access application data. The metadata describes the names of data

elements, their type, and the relationships between entities. Access to this information can greatly simplify the transformation from one application's data format to another.

The system is intended to satisfy the special needs and methodology adopted at the AOU. The SIS is flexible enough to adapt to the specific needs of branches while maintaining a unified standard that facilitates the interoperability of the system amongst branches and the headquarters. The SIS performs all aspects of students' information functions from filing an application to admission up to graduation, within the AOU methods. The SIS deals with all the entities involved and facilitate an easy and reliable way of the entities to perform their functions.

3.2 In-house development and enhancements

To fit the AOU requirements and specification, a number of modifications and customizations were made including:

- Log records: Logs are replicated into other isolated tables, to increase performance, and to keep track records for long period, while removing these log records from original tables timely. It is important to monitor the activities of students as well as tutors over the VLE to assure full participation from all members of the learning process.
- Students' attendance and absences: The system is now capable of monitoring and recording the attendance online during face-to-face tutoring.
- Capturing random samples for the required course activities such as TMAs, online quizzes, and online finals. This automatic process creates a folder for each course and inside each folder there are subfolders according to the sections of the course. In each section folder, the system stores three random samples of the required documents according to the diagram in Fig. 6.
- Grade reports: One of the main development features of integrating VLE and SIS is to migrate grades from VLE to SIS at the end of the course where the grades are recorded for administration purposes and all statistical and grade distribution reports are generated. These reports are migrated back to VLE to be stored in the appropriate subfolders for each course sections (see Fig. 6).
- Questionnaires: Instead of performing student questionnaires manually and then entering data for analytical purposes, we have developed an online questionnaire feature to VLE where each student in each course fills this form online to evaluate the course, the tutor, and tutoring environment. This development saves a lot of efforts and the resulted statistics are becoming more accurate. Same development is applied for tutor review questioner.

3.3 Integrating VLE with HRS

AOU uses a computerized system called human resource system to serve the employees and to keep all employee's records and transactions including:

- Basic information and details related to the employee and the changes that take place.
- Personnel information related to the employee.
- Academic qualifications of employees.
- Practical experience of employees.
- Personnel documents and attended workshops.
- Allowing all employees to take leave or vacations and following up on the rejection or acceptance of these online.
- Do all financial tasks and issuing salary slips for employees and emailing them to the private account of employees.
- General different types of required reports

By connecting VLE with HRS, all the required information regarding tutors and other academic teaching personnel information will be automatically migrated from HRS to the VLE. This process saves a lot of efforts and reduces time and redundancy of storing information, in addition to the increase of efficiency and accuracy. The process starts at the beginning of each semester by creating the groups for ever

offered course over the VLE platform. All required information for creating groups and assigning tutors are migrated from the semester timetable in the SIS system. The time table contains group details in addition to tutor identification number. The rest of required tutor information such as email, department, major, title, etc are collected from the migration with the HRS. Many further studies regarding the integration of SIS with all computerized systems still in progress to obtain more efficient procedures within AOU daily functions.

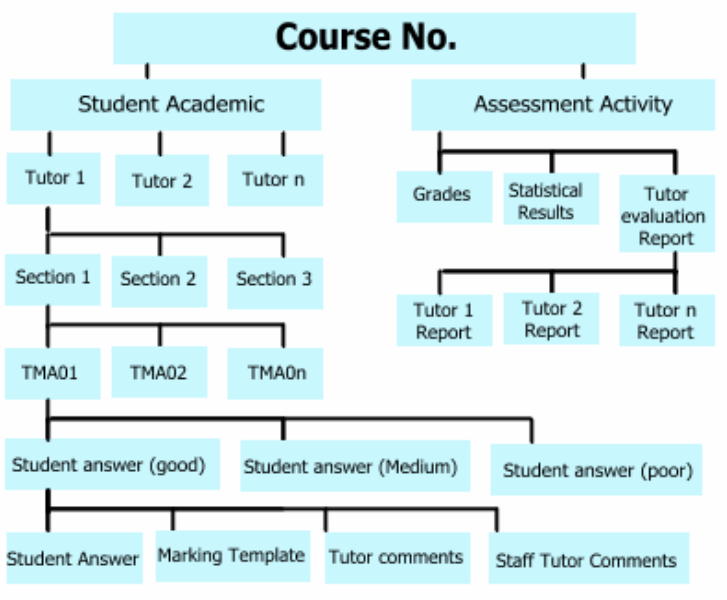


FIGURE 6: Filing structure of courses samples.

4. Quality assurance strategy at AOU

AOU partnership with UKOU requires a set of conditions that has to be fulfilled from AOU side, one of these conditions is to be subject to the review of OUVS, Open University Validation Services, which is the quality assurance accreditation unit of UKOU. OUVS follows the quality assurance standards of QAA agency. Arab Open University with the collaboration with UKOU performs a number of procedures to guarantee the quality of learning process. The descriptions of these procedures are summarized in the following points:

- TMA marking template: Tutor marked assignment template is a form filled by the tutor of a course for each submitted TMA by students. It contains the deserved grade for every part of the TMA along with the feedback comments to the students.
- TMA monitoring: A form filled by the course coordinator and the program coordinator designed for monitoring tutors marking and filling the TMA templates.
- TMA samples: Three TMA samples should be collected for each section. One with a good grade, one is average, and one with a low grade.
- Quiz samples: Three samples should be collected for each quiz.
- Final exam samples: Three samples should be collected for each section of every course.
- Student questionnaire: A questionnaire filled by the students of every section to monitor the tutor, the course, and the tutoring environment.
- Tutor view questionnaire: A questionnaire filled by tutors to monitor the course content and the tutoring environment.
- Face-to-face preview: A form filled by the program coordinator to monitor tutor performance after attending a tutoring session of a specific tutor.

- Final grade statistics and distributions: grades reports and distributing of grades generated by SIS system after submitting student final grades.

At the end of each semester, each course coordinator has to prepare a complete folder that contains the following documents:

- Three samples of a marked TMA for each tutor in the course, each sample should be associated with its marking template and its monitoring form approved by the program coordinator. Notice that the three samples should be selected randomly; one is good, one is average and one is weak, and this is done automatically nowadays.
- Three samples from each quiz during running the course. One sample from each of the good, average and weak categories.
- Three samples from the marked final exam of the course. One sample from each of the good, average and weak categories.
- 4-The face-to-face monitoring form for each tutor.
- The tutor monitoring forms
- Results of student questioners on the course level and for each tutor.
- Students' grades
- Grade distributions and statistics.

One of the duties of the program coordinator is to supervise the preparation of the above documents for all courses in the program and send them to the headquarter of the university to be reviewed from the external examiners whom usually come from UKOU.

Notice that preparing and performing such documents consume the time and efforts of many administrational and educational members of the university including tutors, course coordinators, program coordinators, and secretaries.

5. CONSLUSION & FUTURE WORK

Arab Open University is an educational institute that depends on the strategy of open learning and distance learning to deliver its educational mission. The backbone of the learning process is using e-learning technology. The need for virtual learning environments to deliver the courses online becomes a significant issue. We discussed the efficient features of Moodle as a virtual learning environment used in the Arab Open University. In this paper, a complete description of the improvements that have been conducted over the virtual learning environment at AOU is introduced. The integration process between VLE and student information system (SIS), VLE and human resource system (HRS) with clarifying the advantages of such integration is discussed. The university strict regulations on the learning process to assure the quality of delivering all learning activities in an optimal way. Accordingly, there is a need to improve the existing virtual learning environment to guarantee the implementation of such quality assurance regulations electronically to save effort and to perform all required procedures.

Our new trend at AOU is to integrate VLE system with mobile technology, where students could receive the important notifications and messages through their mobile devices. We are investigating the needs and requirements to manage to do that.

6. REFERENCES

1. Freed, K. “A History of Distance Learning”, Retrieved June 25, 2004 from http://www.media-visions.com/ed-distlrn.html, 2004.

2. Phipps, R.A., & Merisotis, J.P. “What's the difference? A review of contemporary research in the effectiveness of distance learning in higher education”. Washington, DC: Institute for Higher Education Policy. (ERIC Document Reproduction Service No. ED 429 524), 1999.

3. Hill, J.R. “Distance learning environments via world wide web”. In B.H. Khan (ED.). “Web-based instruction”. Englewood Cliffs, NJ: Education Technology Publications, pp. 75-80, 1997.

4. Johnson, S. D., Aragon, S. R., Shaik, N., & Palma-Rivas, N. “Comparative analysis of learner satisfaction and learning outcomes in online and face-toface learning environments”. Journal of Interactive Learning Research, 11 (1): 29-49, 2000.

5. LaRose, R., Gregg, J. & Eastin, M. “Audiographic telecourses for the Web: An experiment. Journal of Computer-Mediated Communication” [Online], _4(2), 1998 http://jcmc.indiana.edu/vol4/issue2/larose.html

6. Trinkle, D.A. “Distance education: A means to an end, no more, no less. Chronicle of Higher Education”. 45(48): 1, 1999.
7. [7] Owston, R. “The World Wide Web: A technology to enhance teaching and learning?” Educational Researcher, 26(2): 27-33, 1997.

8. Mikic, F., & Anido, L. “Towards a standard for mobile technology”. In Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06) - Volume 00. Pp. 217-222, 2006.

9. Freed, K. “A History of Distance Learning”. Retrieved June 25, 2004. [Online]: http://www.media-visions.com/ed-distlrn.html

10. Rovai, A.P., & Barnum, K.T. “On-line course effectiveness: an analysis of student interactions and perceptions of learning”. Journal of Distance Education, 18(1): 57-73, 2003.

11. Hammad, S., Al-Ayyoub, A.E., & Sarie, T. “Combining existing e-learning components towards an IVLE”. EBEL 2005 conference. [Online] http://medforist.grenobleem.com/Contenus/Conference%20Amman%20EBEL%2005/pdf/15.pdf

12. [Online]: www.lmuaut.demon.co.uk/trc/edissues/ptgloss.htm

13. [Online]: http://alt.uno.edu/glossary.html

14. Giannini-Gachago D., Lee M., & Thurab-Nkhosi D. “Towards Development of Best Practice Guidelines for E-Learning Courses at the University of Botswana”. In Proceeding Of Computers and Advanced Technology In Education, Oranjestad, Aruba, 2005.

15. Louca, S., Constantinides, C., & Ioannou, A. “Quality Assurance and Control Model for E-Learning”. In Proceeding (428) Computers and Advanced Technology in Education. 2004

16. Cole J., Using Moodle, O'Reilly. “First edition”. July 2005

17. Kruse, K. “The benefits of e-learning”. 2003. [Online] : http://www.executivewomen.org/pdf/benefits_elearning.pdf

18. [Online]: http://www.elearningproviders.org/HTML/pages/link.asp

19. Schutte, J.G. (1997). “Virtual teaching in higher education: The new intellectual superhighway or just another traffic jam”. 1997. [Online] http://www.csun.edu/sociology/virexp.html

Content Based Image Retrieval based on Color, Texture and Shape features using Image and its complement

P. S. Hiremath hiremathps@yahoo.co.in
Dept. of P.G. Studies and Research in Computer Science,
Gulbarga University,
Gulbarga, Karnataka, India

Jagadeesh Pujari jaggudp@yahoo.com
Dept. of P.G. Studies and Research in Computer Science,
Gulbarga University,
Gulbarga, Karnataka, India

Abstract

Color, texture and shape information have been the primitive image descriptors in content based image retrieval systems. This paper presents a novel framework for combining all the three i.e. color, texture and shape information, and achieve higher retrieval efficiency using image and its complement. The image and its complement are partitioned into non-overlapping tiles of equal size. The features drawn from conditional co-occurrence histograms between the image tiles and corresponding complement tiles, in RGB color space, serve as local descriptors of color and texture. This local information is captured for two resolutions and two grid layouts that provide different details of the same image. An integrated matching scheme, based on most similar highest priority (MSHP) principle and the adjacency matrix of a bipartite graph formed using the tiles of query and target image, is provided for matching the images. Shape information is captured in terms of edge images computed using Gradient Vector Flow fields. Invariant moments are then used to record the shape features. The combination of the color and texture features between image and its complement in conjunction with the shape features provide a robust feature set for image retrieval. The experimental results demonstrate the efficacy of the method.

Keywords: Multiresolution grid, Integrated matching,, Conditional co-occurrence histograms, Local descriptors, Gradient vector flow field.

1. INTRODUCTION

Content-based image retrieval (CBIR) [1],[2],[3],[4],[5],[6],[7],[8] is a technique used for retrieving similar images from an image database. The most challenging aspect of CBIR is to bridge the gap between low-level feature layout and high-level semantic concepts.

Color, texture and shape features have been used for describing image content. Different CBIR systems have adopted different techniques. Few of the techniques have used global color

and texture features [8],[9],[10] where as few others have used local color and texture features [2],[3],[4],[5]. The latter approach segments the image into regions based on color and texture features. The regions are close to human perception and are used as the basic building blocks for feature computation and similarity measurement. These systems are called region based image retrieval (RBIR) systems and have proven to be more efficient in terms of retrieval performance. Few of the region based retrieval systems, e,g, [2], compare images based on individual region-to-region similarity. These systems provide users with rich options to extract regions of interest. But precise image segmentation has still been an open area of research. It is hard to find segmentation algorithms that conform to the human perception. For example, a horse may be segmented into a single region by an algorithm and the same algorithm might segment horse in another image into three regions. These segmentation issues hinder the user from specifying regions of interest especially in images without distinct objects. To ensure robustness against such inaccurate segmentations, the integrated region matching (IRM) algorithm [5] proposes an image-to-image similarity combining all the regions between the images. In this approach, every region is assigned significance worth its size in the image. A region is allowed to participate more than once in the matching process till its significance is met with. The significance of a region plays an important role in the image matching process. In either type of systems, segmentation close to human perception of objects is far from reality because the segmentation is based on color and texture. The problems of over segmentation or under segmentation will hamper the shape analysis process. The object shape has to be handled in an integral way in order to be close to human perception. Shape feature has been extensively used for retrieval systems [14],[15].

Image retrieval based on visually significant points [16],[17] is reported in literature. In [18], local color and texture features are computed on a window of regular geometrical shape surrounding the corner points. General purpose corner detectors [19] are also used for this purpose. In [20], fuzzy features are used to capture the shape information. Shape signatures are computed from blurred images and global invariant moments are computed as shape features. The retrieval performance is shown to be better than few of the RBIR systems such as those in [3],[5],[21].

The studies mentioned above clearly indicate that, in CBIR, local features play a significant role in determining the similarity of images along with the shape information of the objects. Precise segmentation is not only difficult to achieve but is also not so critical in object shape determination. A windowed search over location and scale is shown more effective in object-based image retrieval than methods based on inaccurate segmentation [22]. The objective of this paper is to develop a technique which captures local color and texture descriptors in a coarse segmentation framework of grids, and has a shape descriptor in terms of invariant moments computed on the edge image. The image is partitioned into equal sized non-overlapping tiles. The features computed on these tiles serve as local descriptors of color and texture. In [12] it is shown that features drawn from conditional co-occurrence histograms using image and its complement in RGB color space perform significantly better. These features serve as local descriptor of color and texture in the proposed method. The grid framework is extended across resolutions so as to capture different image details within the same sized tiles. An integrated matching procedure based on adjacency matrix of a bipartite graph between the image tiles is provided, similar to the one discussed in [5], yielding image similarity. A two level grid framework is used for color and texture analysis. Gradient Vector Flow (GVF) fields [13] are used to compute the edge image, which will capture the object shape information. GVF fields give excellent results in determining the object boundaries irrespective of the concavities involved. Invariant moments are used to serve as shape features. The combination of these features forms a robust feature set in retrieving applications. The experimental results are compared with [3],[5],[20],[21] and are found to be encouraging.

The section 2 outlines the system overview and proposed method. The section 3 deals with experimental setup. The section 4 presents experimental results. The section 5 presents conclusions.

2. SYSTEM OVERVIEW AND PROPOSED METHOD

The FIGURE 1 shows the system overview.

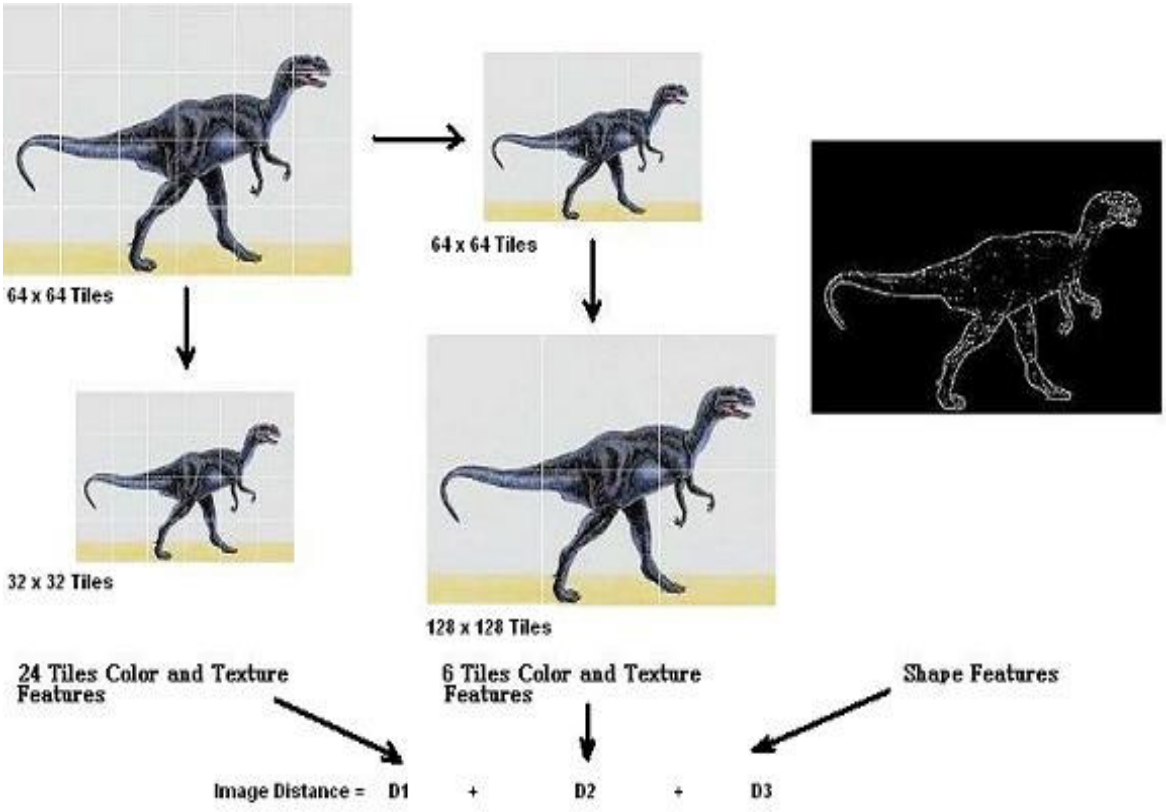


FIGURE 1: System overview.

The proposed method is described below:

2.1 Grid

An image is partitioned into 24 (4 x 6 or 6 x 4) non overlapping tiles as shown in FIGURE1. These tiles will serve as local color and texture descriptors for the image. Features drawn from conditional co-occurrence histograms between image tiles and the corresponding complement tiles are used for color and texture similarity. With the Corel dataset used for experimentation (comprising of images of size either 256 x 384 or 384 x 256), with 6 x 4 (or 4 x 6) partitioning, the size of individual tile will be 64 x 64. The choice of smaller sized tiles than 64 x 64 leads to degradation in the performance. Most of the texture analysis techniques make use of 64 x 64 blocks. This tiling structure is extended to second level decomposition of the image. The image is decomposed into size M/2 x N/2, where M and N are number of rows and columns in the original image respectively. With a 64 x 64 tile size, the number of tiles resulting at this resolution is 6 as shown in FIGURE 1. This allows us to capture different image information across resolutions. For robustness, we have also included the tile features resulting from the same grid structure (i.e. 24 tiles at resolution 2 and 6 tiles at resolution 1) as shown in FIGURE 1. The computation of features is discussed in section 3. Going beyond second level of decomposition added no significant information. So, a two level structure is used.

2.2 Integrated image matching

An integrated image matching procedure similar to the one used in [5] is proposed. The matching of images at different resolutions is done independently as shown in FIGURE 1. Since at any given level of decomposition the number of tiles remains the same for all the images (i.e. either

24 at first level of decomposition or 6 at second level of decomposition), all the tiles will have equal significance. In [23] a similar tiled approach is proposed, but the matching is done by comparing tiles of query image with tiles of target image in the corresponding positions. In our method, a tile from query image is allowed to be matched to any tile in the target image. However, a tile may participate in the matching process only once. A bipartite graph of tiles for the query image and the target image is built as shown in FIGURE 2. The labeled edges of the bipartite graph indicate the distances between tiles. A minimum cost matching is done for this graph. Since, this process involves too many comparisons, the method has to be implemented efficiently. To this effect, we have designed an algorithm for finding the minimum cost matching based on most similar highest priority (MSHP) principle using the adjacency matrix of the bipartite graph. Here in, the distance matrix is computed as an adjacency matrix. The minimum distance d_{ij} of this matrix is found between tiles i of query and j of target. The distance is recorded and the row corresponding to tile i and column corresponding to tile j, are blocked (replaced by some high value, say 999). This will prevent tile i of query image and tile j of target image from further participating in the matching process. The distances, between i and other tiles of target image and, the distances between j and other tiles of query image, are ignored (because every tile is allowed to participate in the matching process only once). This process is repeated till every tile finds a matching. The process is demonstrated in FIGURE 3 using an example for 4 tiles. The complexity of the matching procedure is reduced from $O(n^2)$ to $O(n)$, where n is the number of tiles involved. The integrated minimum cost match distance between images is now defined as:

$$D_{qt} = \sum_{i=1,n} \sum_{j=1,n} d_{ij}$$

, where d_{ij} is the best-match distance between tile i of query image q and tile j of target image t and D_{qt} is the distance between images q and t.

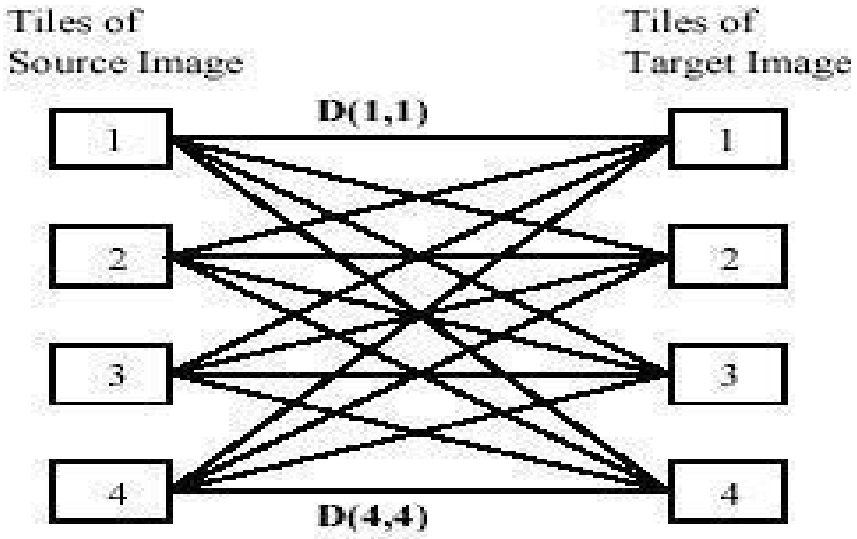


FIGURE 2: Bipartite graph showing 4 tiles of both the images.

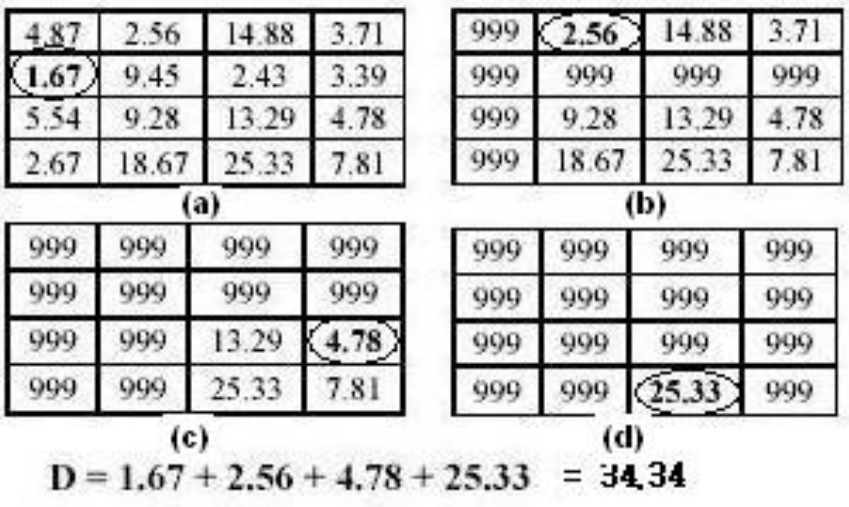


FIGURE 3: Image similarity computation based on MSHP principle, (a) first pair of matched tiles i=2,j=1 (b) second pair of matched tiles i=1, j=2 (c) third pair of matched tiles i=3, j=4 (d) fourth pair of matched tiles i=4,j=3, yielding the integrated minimum cost match distance 34.34.

2.3 Shape

Shape information is captured in terms of the edge image of the gray scale equivalent of every image in the database. We have used gradient vector flow (GVF) fields to obtain the edge image [13].

Gradient Vector Flow:

Snakes, or active contours, are used extensively in computer vision and image processing applications, particularly to locate object boundaries. Problems associated with their poor convergence to boundary concavities, however, have limited their utility. Gradient vector flow (GVF) is a static external force used in active contour method. GVF is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the images. It differs fundamentally from traditional snake external forces in that it cannot be written as the negative gradient of a potential function, and the corresponding snake is formulated directly from a force balance condition rather than a variational formulation.

The GVF uses a force balance condition given by

$$F_{\text{int}} + F_{ext}^{(p)} = 0,$$

where F_{int} is the internal force and $F_{ext}^{(p)}$ is the external force.

The external force field $F_{ext}^{(p)} = V(x, y)$ is referred to as the GVF field. The GVF field $V(x, y)$ is a vector field given by $V(x, y) = [u(x, y), v(x, y)]$ that minimizes the energy functional

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |V - \nabla f|^2 \, dxdy$$

This variational formulation follows a standard principle, that of making the results smooth when there is no data. In particular, when $|\nabla f|$ is small, the energy is dominated by the sum of squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $V = |\nabla f|$. This produces the desired effect of keeping V nearly equal to the gradient of the edge map when it is large, but forcing the field to be slowly-varying in homogeneous regions. The

parameter μ is a regularization parameter governing the tradeoff between the first term and the second term in the integrand.

The GVF field gives excellent results on concavities supporting the edge pixels with opposite pair of forces, obeying force balance condition, in one of the four directions (horizontal, vertical and diagonal) unlike the traditional external forces which support either in the horizontal or vertical directions only. The algorithm for edge image computation is given below:

Algorithm: (edge image computation)

1. Read the image and convert it to gray scale.
2. Blur the image using a Gaussian filter.
3. Compute the gradient map of the blurred image.
4. Compute GVF. (100 iterations and $\mu = 0.2$)
5. Filter out only strong edge responses using $k\sigma$, where σ is the standard deviation of the GVF. (k – value used is 2.5).
6. Converge onto edge pixels satisfying the force balance condition yielding edge image.

3. EXPERIMENTAL SETUP

(a) **Data set**: Wang's [11] dataset comprising of 1000 Corel images with ground truth. The image set comprises 100 images in each of 10 categories. The images are of the size 256 x 384 or 384 x 256.

(b) **Feature set**: The feature set comprises color, texture and shape descriptors computed as follows:

Color and Texture: Conditional co-occurrence histograms between image and its complement in RGB color space provide the feature set for color and texture. The method is as explained below:

Co-occurrence histogram computation.

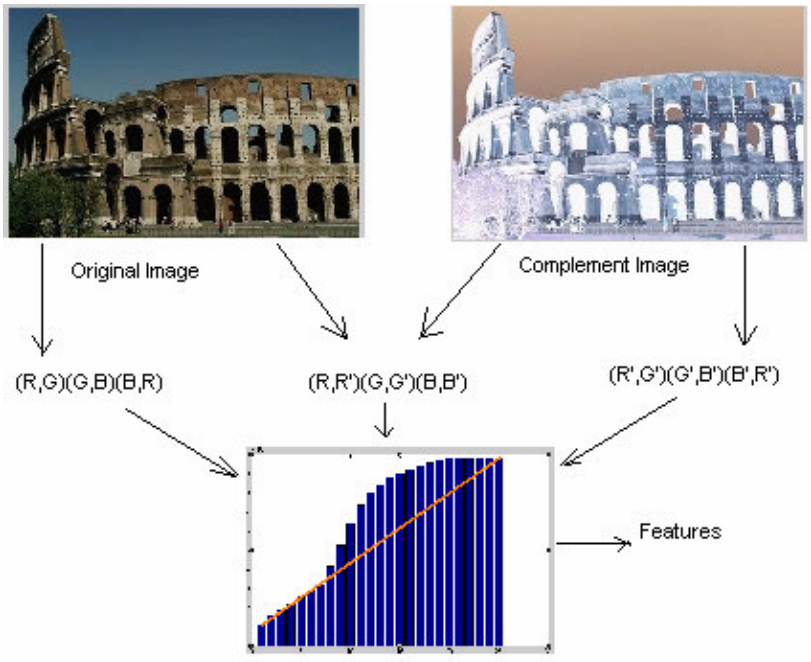


FIGURE 4: Illustration of co-occurrence histogram and feature computation.

Our proposed method is an extension of the co-occurrence histogram method to multispectral images i.e. images represented using n channels. Co-occurrence histograms are constructed, for inter-channel and intra-channel information coding using image and its complement. The complement of a color image $I = (R, G, B)$ in the RGB space is defined by $\overline{I} = (255 - R, 255 - G, 255 - B) \equiv (\overline{R}, \overline{G}, \overline{B})$. The nine combinations considered in RGB color space are: $(R, G), (G, B), (B, R), (\overline{R}, \overline{G}), (\overline{G}, \overline{B}), (\overline{B}, \overline{R}), (R, \overline{R}), (G, \overline{G})$ & $(B, \overline{B})$, where R, G & B represent the Red Green and Blue channels of the input image and $\overline{R}, \overline{G}$ & $\overline{B}$ represent the corresponding channels in the complement image. The translation vector is t[d,a] where d is distance and a is direction. In our experiments we have considered a distance of 1 (d=1) and eight angles (a=0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, 315^0). Two co-occurrence histograms for each channel pair, for each of the eight angles, are constructed using a max-min composition rule, yielding a total of 16 histograms per channel pair. Then the histograms corresponding to opponent angles are merged yielding a total of 8 histograms per channel pair i.e. 0^0 with 180^0, 45^0 with 225^0, 90^0 with 270^0 and 135^0 with 315^0. The feature set comprises of 216 features in all with 3 features each computed from the normalized cumulative histogram i.e. 9 pairs x 8 histograms x 3 features. The outline of the method is illustrated schematically in FIGURE 4. The method for histogram computation for one pair (RG), for one angle (0^0) is presented below:

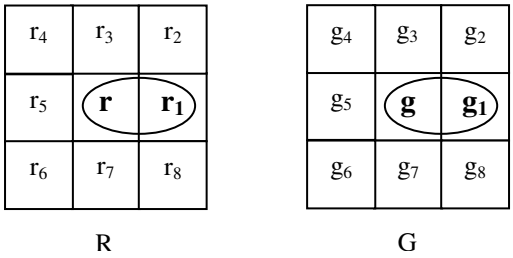


FIGURE 5: 8-nearest neighbors of r and g in R and G planes respectively.

Method of computation of Histograms:

1. A pixel r in R plane and a pixel g in the corresponding location in G plane are shown above with their immediate eight neighbors. The neighboring pixels of r and g considered for co-occurrence computation are shown by the circles in the FIGURE 5.
2. Consider two histograms H1 and H2 for R based on the maxmin composition rule stated below:
Let α = max(min(r,g_1), min(g,r_1))
Then, r ∈ H1 if α = min(r,g_1)
and r ∈ H2 if α = min(g,r_1)
It yields 16 histograms per pair, 2 for each direction.

Feature computation:

The features considered are:
a. The slope of the line of regression for the data corresponding to the normalized cumulative histograms [26].
b. The mean bin height of the cumulative histogram.

c. The mean deviation of the bins.

A total of 216 features are computed for every image tile (per resolution).

Shape: Translation, rotation, and scale invariant one-dimensional normalized contour sequence moments are computed on the edge image [24,25]. The gray level edge images of the R, G and B individual planes are taken and the shape descriptors are computed as follows:

$$F_1 = \frac{(\mu_2)^{1/2}}{m_1},$$

$$F_2 = \frac{\mu_3}{(\mu_2)^{3/2}},$$

$$F_3 = \frac{\mu_4}{(\mu_2)^2},$$

$$F_4 = \overline{\mu_5},$$

where

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r, \quad \mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)-m_1]^r, \quad \overline{\mu_r} = \frac{\mu_r}{(\mu_2)^{r/2}}$$

The z(i) is the set of Euclidian distances between centroid and all N boundary pixels of the digitized shape.

A total of 12 features result from the above computations. In addition, moment invariant to translation, rotation and scale is taken on R, G and B planes individually considering all the pixels [24]. The transformations are summarized as below:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}},$$

where

$\gamma = \frac{p+q}{2} + 1$, (Central moments) $\quad \phi = \eta_{20} + \eta_{02}$, (Moment invariant)

The above computations will yield additional 3 features amounting to a total of 15 features.

The distance between two images is computed as D = D_1+ D_2 + D_3, where D_1 and D_2 are the distance computed by integrated matching scheme at two resolutions and D_3 is the distance resulting from shape comparison.

Canberra distance measure is used for similarity comparison in all the cases. It allows the feature set to be in unnormalized form. The Canberra distance measure is given by:

$$CanbDist(x, y) = \sum_{i=1}^{d}\frac{|x_i - y_i|}{|x_i| + |y_i|},$$

where x and y are the feature vectors of database and query image, respectively, of dimension d.

4. EXPERIMENTAL RESULTS

The experiments were carried out as explained in the sections 2 and 3.

The results are benchmarked with standard systems using the same database as in [3,5,20,21]. The quantitative measure defined is average precision as explained below:

$$p(i) = \frac{1}{100} \sum_{1 \le j \le 1000, r(i,j) \le 100, ID(j)=ID(i)} 1 ,$$

where p(i) is precision of query image i, ID(i) and ID(j) are category ID of image i and j respectively, which are in the range of 1 to 10. The $r(i, j)$ is the rank of image j (i.e. position of image j in the retrieved images for query image i , an integer between 1 and 1000).
This value is the percentile of images belonging to the category of image i in the first 100 retrieved images.

The average precision p_t for category $t\ (1 \le t \le 10)$ is given by

$$p_t = \frac{1}{100} \sum_{1 \le i \le 1000, ID(i)=t} p(i)$$

The comparison of experimental results of proposed method with other standard retrieval systems reported in the literature [3,5,20,21] is presented in Table 1. The SIMPLIcity and FIRM are both segmentation based methods. Since in these methods, textured and non textured regions are treated differently with different feature sets, their results are claimed to be better than histogram based method [21]. Further, edge based system [20] is at par or at times better than SIMPLIcity [5] and FIRM [3]. But, in most of the categories our proposed method has performed at par or at times even better than these systems. FIGURE 4 shows the sample retrieval results for all the ten categories. The first image is the query image.

The experiments were carried out on a Pentium IV, 1.8 GHz processor with 384 MB RAM using MATLAB.

Class	Average Precision				
	SIMPLIcity [5]	**Histogram Based [21]**	**FIRM [3]**	**Edge Based [20]**	**Proposed Method**
Africa	.48	.30	.47	.45	.54
Beaches	.32	.30	.35	.35	.38
Building	.35	.25	.35	.35	.40
Bus	.36	.26	.60	.60	.64
Dinosaur	.95	.90	.95	.95	.96
Elephant	.38	.36	.25	.60	.62
Flower	.42	.40	.65	.65	.68
Horses	.72	.38	.65	.70	.75
Mountain	.35	.25	.30	.40	.45
Food	.38	.20	.48	.40	.53

TABLE 1: Comparison of average precision obtained by proposed method with other standard retrieval systems[3],[5,[20,[21].

5. CONSLUSIONS

We have proposed a novel method for image retrieval using color, texture and shape features within a multiresolution multigrid framework. The images are partitioned into non-overlapping tiles. Texture and color features are extracted from these tiles at two different resolutions in two grid framework. Features drawn from conditional co-occurrence histograms computed by using image and its complement in RGB color space, serve as color and texture descriptors. An integrated matching scheme based on most significant highest priority (MSHP) principle and adjacency matrix of a bipartite graph constructed between image tiles, is implemented for image similarity. Gradient vector flow fields are used to extract shape of objects. Invariant moments are used to describe the shape features. A combination of these color, texture and shape features provides a robust feature set for image retrieval. The experiments using the Corel dataset demonstrate the efficacy of this method in comparison with the existing methods in the literature.

6. REFERENCES

1. Ritendra Datta, Dhiraj Joshi, Jia Li and James Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age", Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, November 10-11, 2005, Hilton, Singapore.
2. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," in *IEEE Trans. On PAMI*, vol. 24, No.8, pp. 1026-1038, 2002.
3. Y. Chen and J. Z. Wang, "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval," in *IEEE Trans. on PAMI*, vol. 24, No.9, pp. 1252-1267, 2002.
4. A. Natsev, R. Rastogi, and K. Shim, "WALRUS: A Similarity Retrieval Algorithm for Image Databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp. 395–406, 1999.
5. J. Li, J.Z. Wang, and G. Wiederhold, "IRM: Integrated Region Matching for Image Retrieval," in *Proc. of the 8th ACM Int. Conf. on Multimedia*, pp. 147-156, Oct. 2000.
6. V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Region-based Image Retrieval Using an Object Ontology and Relevance Feedback," in *Eurasip Journal on Applied Signal Processing*, vol. 2004, No. 6, pp. 886-901, 2004.
7. W.Y. Ma and B.S. Manjunath, "NETRA: A Toolbox for Navigating Large Image Databases," in Proc. IEEE Int. Conf. on Image Processing, vol. I, Santa Barbara, CA, pp. 568–571, Oct. 1997.
8. W. Niblack *et al.*, "The QBIC Project: Querying Images by Content Using Color, Texture, and Shape," in *Proc. SPIE*, vol. 1908, San Jose, CA, pp. 173–187, Feb. 1993.
9. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases," in *Proc. SPIE Storage and Retrieval for Image and Video Databases II*, San Jose, CA, pp. 34–47, Feb. 1994.
10. M. Stricker, and M. Orengo, "Similarity of Color Images," in Proc. SPIE Storage and Retrieval for Image and Video Databases, pp. 381-392, Feb. 1995.
11. http://wang.ist.psu.edu/
12. P.S.Hiremath, Jagadeesh Pujari, "Enhancing performance of region based image retrieval system using joint co-occurrence histograms between image and its complement in RGB color space." in Proc. National Conference on Knowledge-Based computing systems and Frontier Technologies (NCKBFT-07), Manipal, India, 19-20 Feb, 2007.
13. Chenyang Xu, Jerry L Prince, "Snakes,Shapes, and Gradient Vector Flow", IEEE Transactions on Image Processing, Vol-7, No 3,PP 359-369, March 1998.
14. T. Gevers and A.W.M. Smeuiders., "Combining color and shape invariant features for image retrieval", Image and Vision computing, vol.17(7),pp. 475-488 , 1999.
15. A.K.Jain and Vailalya,, "Image retrieval using color and shape", pattern recognition, vol. 29, pp. 1233-1244, 1996.

16. D.Lowe, “Distinctive image features from scale invariant keypoints”, International Journal of Computer vision, vol. 2(6),pp.91-110,2004.
17. K.Mikolajezyk and C.Schmid, “Scale and affine invariant interest point detectors”, International Journal of Computer Vision, vol. 1(60),pp. 63-86, 2004.
18. Etinne Loupias and Nieu Sebe, “Wavelet-based salient points: Applications to image retrieval using color and texture features”, in Advances in visual Information systems, Proceedings of the 4th International Conference, VISUAL 2000, pp. 223-232, 2000.
19. C. Harris and M. Stephens, “A combined corner and edge detectors”, 4th Alvey Vision Conference, pp. 147-151, 1988.
20. M.Banerjee, M,K,Kundu and P.K.Das, “Image Retrieval with Visually Prominent Features using Fuzzy set theoretic Evaluation”, ICVGIP 2004, India, Dec 2004.
21. Y. Rubner, L.J. Guibas, and C. Tomasi, “The earth mover’s distance, multi-dimensional scaling, and color-based image retrieval”, Proceedings of DARPA Image understanding Workshop, pp. 661-668, 1997.
22. D.Hoiem, R. Sukhtankar, H. Schneiderman, and L.Huston, “Object-Based Image retrieval Using Statistical structure of images”, Proc CVPR, 2004.
23. P. Howarth and S. Ruger, “Robust texture features for still-image retrieval”, IEE. Proceedings of Visual Image Signal Processing, Vol. 152, No. 6, December 2005.
24. Dengsheng Zhang, Guojun Lu, “Review of shape representation and description techniques”, Pattern Recognition Vol. 37,pp 1-19, 2004.
25. M. Sonka, V. Halvac, R.Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, London, UK, NJ, 1993.
26. P.Nagabhushan, R. Pradeep Kumar, “Multiresolution Knowledge Mining using Wavelet Transform”, Proceeding of the International Conference on Cognition and Recognition, Mandya, pp781-792, Dec 2005.

A comparative study of conventional effort estimation and fuzzy effort estimation based on Triangular Fuzzy Numbers

Harish Mittal harish.mittal@vcenggrtk.com
Department of IT
Vaish College of Engineering,
Rohtak, 124001, India

Pradeep Bhatia pk_bhatia20002@yahoo.com
Department of Computer Science
G.J. University of Science & Technology
Hisar, 125001, India

Abstract

Effective cost estimation is the most challenging activity in software development. Software cost estimation is not an exact science. However it can be transformed from a black art to a series of systematic steps that provide estimate with acceptable risk. Effort is a function of size. For estimating effort first we face sizing problem. In direct approach size is measured in lines of code (LOC). In indirect approach, size is represented as function points (FP). In this paper we use indirect approach. Fuzzy logic is used to find fuzzy functional points and then the result is defuzzified to get the functional points and hence the size estimation in person hours. Triangular fuzzy numbers are used to represent the linguistic terms in Function Point Analysis (FPA) complexity matrixes We can optimise the results for any application by varying the fuzziness of the triangular fuzzy numbers.

Keywords: FP, FFP, FPA, FFPA, LOC, Fuzzy logic, Triangular Fuzzy Number, Membership function and Fuzziness.

1. INTRODUCTION

Out of the three principal components of cost i.e., hardware costs, travel and training costs, and effort costs, the effort cost is dominant. Software cost estimation starts at the proposal state and continues throughout the life time of a project.
There are seven techniques of software cost estimation:

- Algorithm Cost Model
- Expert Judgments
- Estimation by Analogy
- Parkinson's Law
- Pricing to win
- Top-down estimation
- Bottom-up estimation

If these predict radically different costs, more estimation should be sought and the costing process repeated.

Algorithm model, also called parametric model, is designed to provide some mathematical equations to provide software estimation. LOC-based models are algorithm models such as [3, 13, 14, and 15]. Ali Idri and Laila Kjjri [7] proposed the use of fuzzy sets in the COCOMO, 81 models [3]. Musilek, P. and others [11] proposed f-COCOMO model, using fuzzy sets. The methodology of fuzzy sets giving rise to f-COCOMO [11] is sufficiently general to be applied to other models of software cost estimation such as function point method [9]. Software Functional size measurement is regarded as a key aspect in the production, calibration and use of software engineering productivity models because of its independence of technologies and of implementation decisions. W.Pedrycz and others [12] found that

the concept of information granularity and fuzzy sets, in particular, plays an important role in making software cost estimation models more users friendly. Harish Mittal and Pradeep Bhatia [10] used triangular fuzzy numbers for fuzzy logic sizing. Lima, O.S.J. and Others [16] proposed the use of concepts and properties from fuzzy set theory to extend function point analysis to Fuzzy function point analysis, using trapezoid shaped fuzzy numbers for the linguistic variables of function point analysis complexity matrixes.

In this paper we proposed triangular fuzzy numbers to represent the linguistic variables. The results can be optimised for the given application by varying fuzziness of the triangular fuzzy numbers. To apply fuzzy logic first fuzzification is done using triangular fuzzy number, Fuzzy output is evaluated and then estimation is done by defuzzification technique given in this paper.
The paper is divided into sections. Section 2 introduces related terms. In section 3, the technique of estimation of fuzzy functional points and optimisation technique is given; section 4 gives experimental results and section 5 gives conclusions and future research.

2. RELATED TERMS

(a) Fuzzy Number (b) Fuzzy Logic (c) Fuzziness

(d) Function Point Analysis (e) Various criterion for Assessment of Software Cost Estimation Models

(a) Fuzzy Number:

A fuzzy number is a quantity whose value is imprecise, rather than exact as in the case of ordinary single valued numbers. Any fuzzy number can be thought of as a function, called membership function, whose domain is specified, usually the set of real numbers, and whose range is the span of positive numbers in the closed interval [0, 1]. Each numerical value of the domain is assigned a specific value and 0 represents the smallest possible value of the membership function, while the largest possible value is 1. In many respects fuzzy numbers depict the physical world more realistically than single valued numbers. Suppose that we are driving along a highway where the speed limit is 80km/hr, we try to hold the speed at exactly 80km/hr, but our car lacks cruise control, so the speed varies from moment to moment. If we note the instantaneous speed over a period of several minutes and then plot the result in rectangular coordinates, we may get a curve that looks like one of the curves shown below. However there is no restriction on the shape of the curve. The curve in figure 1 is a triangular fuzzy number, the curve in figure 2 is a trapezoidal fuzzy number, and the curve in figure3 is bell shaped fuzzy number.

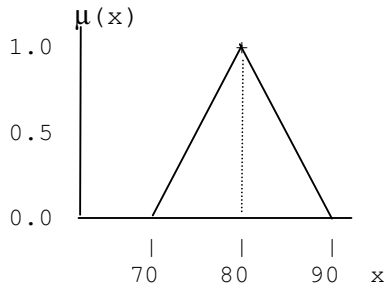


Fig1: Triangular Fuzzy Number

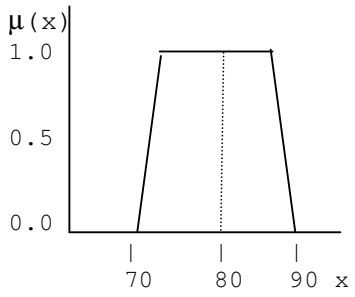


Fig2:Trapezoidal Fuzzy Number

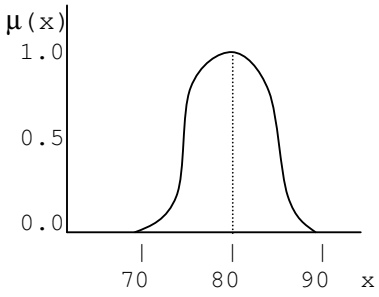


Fig3: Bell shaped Fuzzy Number

(b)Fuzzy Logic

Fuzzy logic is a methodology, to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory, and introduced in 1965 by Prof. Zadeh in the paper Fuzzy Sets [4, 5]. Use of fuzzy sets in logical expression is known as fuzzy logic. A fuzzy set is characterized by a membership function, which associates with each point in the fuzzy set a real number in the interval [0,1], called degree or grade of membership. The membership function may be triangular, trapezoidal, parabolic etc. Fuzzy numbers are special convex and normal fuzzy sets, usually with single modal value, representing uncertain quantitative information. A triangular fuzzy number (TFN) is described by a triplet (α ,m, β), where m is the modal value, α and β are the right and left boundary respectively.

(c) Fuzziness: Fuzziness of a TFN (α, m,β) is defined as:

Fuzziness of TFN (F) = $\frac{\beta - \alpha}{2m}$, 0 < F< 1 … (1)

The higher the value of fuzziness, the more fuzzy is TFN

(d) Function Point Analysis (FPA):

FPA begins with the decomposition of a project or application into its data and transactional functions. The data functions represent the functionality provided to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions describe the functionality provided to the user in relation to the processing this data by the application.

The data functions are:

1. Internal Logical File(ILF)
2. External Interface File (EIF)

The transactional functions are:

1. External Input(EI)
2. External Output(EO)
3. External Inquiry(EI)

Each function is classified according to its relative functional complexity as low, average or high. The data functions relative functional complexity is based on the number of data element types (DETs) and the number of record element types (RETs). The transactional functions are classified according to the number of file types referenced (FTRs) and the number of DETs. The number of FTRs is the sum of the number of ILFs and the number of EIFs updated or queried during an elementary process.

The actual calculation process consists of three steps:

1. Determination of unadjusted function points(UFP)
2. Calculation of value of adjustment factor(VAF)
3. Calculation of final adjusted functional points.

Evaluation of Unadjusted FP:

The unadjusted Functional points are evaluated in the following manner
UFP= $\Sigma\Sigma$ F_{ij} Z_{ij} , for j= 1 to 3 and i = 1 to 5, where Z_{ij} denotes count for component i at level (low, average or high) j, and F_{ij} is corresponding Function Points from table 1.

Level	Function Points				
	ILF	EIF	EI	EO	EQ
Low	7	5	3	4	3
Average	10	7	4	5	4
High	15	10	6	7	6

Table 1: Translation table for the terms low, average and high

Value Adjustment Factor (VAF) is derived from the sum of the degree of influence (DI) of the 14 general system characteristics (GSCc). General System characteristics are:

1. Data communications
2. Distributed data processing
3. Performance
4. Heavily utilised configuration
5. Transaction rate
6. On-line data entry
7. End-user efficiency
8. On-line update

9. Complex processing
10. Reusability
11. Installations ease
12. Operational ease
13. Multiple sites/organisations
14. Facilitate change

The DI of each one of these characteristics ranges from 0 to 5 as follows:

(i) 0- no influence
(ii) 1 -Incidental influence
(iii) 2- Moderate influence
(iv) 3- Average influence
(v) 4- Significant influence
(vi) 5- Strong influence

Total Function Points = UFP * (0.65+ 0.01 * Value Adjustment Factor)

Function points can be converted to Effort in Person Hours. Numbers of studies have attempted to relate LOC and FP metrics [16]. The average number of source code statements per function point has been derived from historical data for numerous programming languages. Languages have been classified into different levels according to the relationship between LOC and FP. Programming language levels and Average numbers of source code statements per function point are given by [17].

Complexity matrix of an ILF or EIF is given in Table 2. Complexity matrix of EO or EQ is given in Table 3**.** Complexity matrix of EI is given in Table 4

RET	**DET**		
	1 to 19	20 to 50	51 or more
1	Low	Low	Average
2 to 5	Low	Average	High
6 or more	Average	High	High

Table 2: Complexity matrix of an ILF or EIF

FTR	**DET**		
	1 to 5	6 to 19	20 or more
Less than 2	Low	Low	Average
2 or 3	Low	Average	High
Greater than 3	Average	High	High

Table 3: Complexity matrix of EO or EQ

FTR	**DET**		
	1 to 4	5 to 15	16 or more
Less than 2	Low	Low	Average
2	Low	Average	High
More than 2	Average	High	High

Table 4: Complexity matrix of EI

The value of function points for the terms low, average and high to each FPA are given in Table1.

(e) Various Criterions for Assessment of Software Cost Estimation Models

There are 4 important criterions for assessment of software cost estimation models:

1. VAF (Variance Accounted For) (%):

$$\text{VAF (\%)} = \left(1 - \frac{\text{var}(E - \hat{E})}{\text{var}\, E}\right) * 100 \quad \ldots(2)$$

2. Mean absolute Relative Error (%):

$$\text{Mean absolute error (\%)} = \frac{\sum f(R_E)}{\sum f} * 100 \quad \ldots(3)$$

3. Variance Absolute Relative Error (%):

$$\text{Variance Absolute Relative Error (\%)} = \frac{\sum f(R_E - mean\ R_E)^2}{\sum f} * 100 \quad \ldots(4)$$

4. Pred (n): Prediction at level n((Pred (n))is defined as the % of projects that have absolute relative error under n[8].

Where,

$$\text{Var x} = \frac{\sum f(x - \bar{x})^2}{\sum f} \quad \ldots(5)$$

$\bar{x}$ = mean x

E = measured effort

$\hat{E}$ = estimated effort

f = frequency

$$\text{Absolute Relative Error } (R_E) = \frac{|E - \hat{E}|}{|E|} \quad \ldots(6)$$

3. FUZZY FUNCTIONAL POINT ANALYSIS (FFPA)

FFPA consists of the following three stages:

1. Fuzzification
2. Defuzzification
3. Optimization

1 Fuzzification

We take each linguistic variables as a triangular Fuzzy numbers, TFN (α, m, β), α≤m, β≥ m. The membership function (μ(x)) for which is defined as:

$$\mu(x) = \begin{cases} 0, & x \le \alpha \\ x-\alpha / m-\alpha, & \alpha \le x \le m \\ \beta-x / \beta-m, & m \le x \le \beta \\ 0, & x \ge \beta \end{cases} \quad \ldots\ldots(7)$$

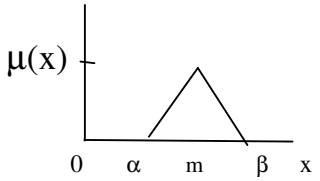


Fig4: representation of TFN (α, m,β)

We create a new linguistic variable, TFN (α, m, k), high or very high, where k is a positive integer. In case low and average are given, we create high variable. In case low, average and high are given, we create very high variable. In case average and high are given, we create very high variable. The creation of the new linguistic variable helps to deal better with larger systems.

Fuzziness of the created linguistic variable (F) = $(k-\alpha)/2m$, $0< F < 1$. …….. (8)

So that $k=2 F m +\alpha$. …… (9)

We can estimate the function points for the new variable, very high, by extrapolation using Newton's interpolation formula [16]. The estimated values of function points are 22,14,9,10 and 9 for the functions ILF, EIF, EI, EO and EQ respectively.

Modified Complexity Matrices for various data and transaction functions are given in the following tables:

DET	Complexity
1-50	Low
51-k1 $k_1 \leq 102$	Average
K_1+1 or more	High

Table 5: Modified Complexity Matrix for ILF & EIF (RET =1)

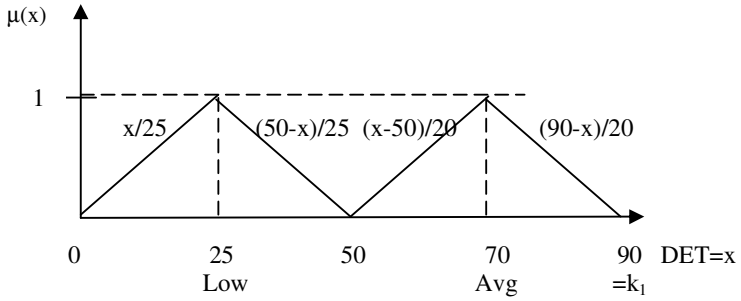


Fig 5

DET	Complexity
1-19	Low
20-50	Average
51-k2 $k_2 \leq 102$	High
K_2+1 or more	V. High

Table 6: Modified Complexity Matrix for ILF & EIF (RET = 2 to 5)

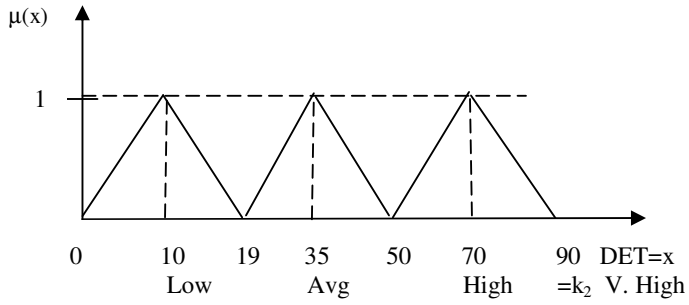


Fig 6:

DET	Complexity
1-19	Average
20-k3 $k_3 \leq 40$	High
K_3+1 or more	V. High

Table 7: Modified Complexity Matrix for ILF & EIF (RET ≥ 6)

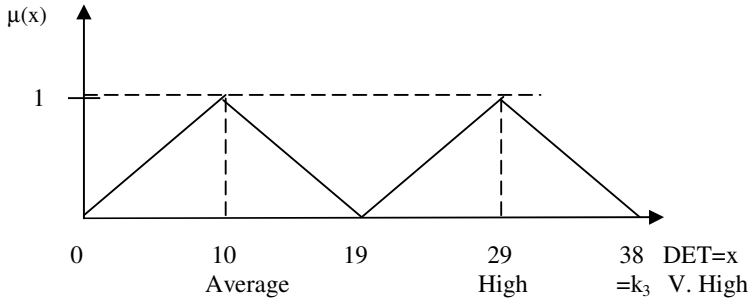


Fig7:

DET	Complexity
1-19	Low
20-k4 $k_4 \leq 40$	Average
K_4+1 or more	High

Table 8: Modified Complexity Matrix for EO & EQ (FTR $\leq$ 2)

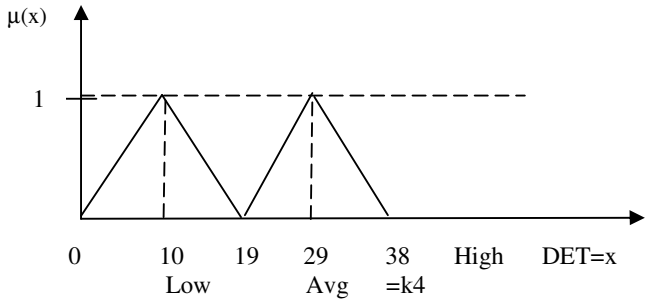


Fig 8:

DET	Complexity
1-5	Low
6-19	Average
20-k5 $k_5 \leq 40$	High
K_5+1 or more	V. High

Table 9: Modified Complexity Matrix for EO & EQ (FTR = 2 or 3)

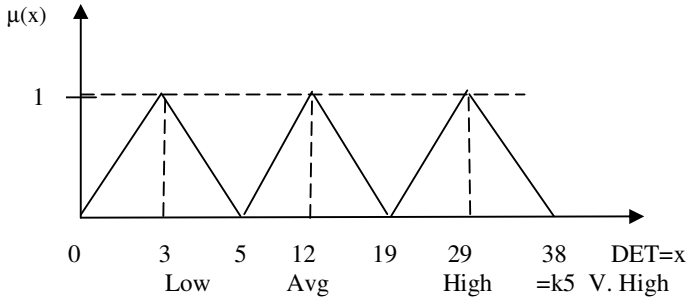


Fig 9:

DET	Complexity
1-5	Average
5-k6 $k_6 \leq 12$	High
K_6+1 or more	V.High

Table 10: Modified Complexity Matrix for EO & EQ (FTR $\geq$ 4)

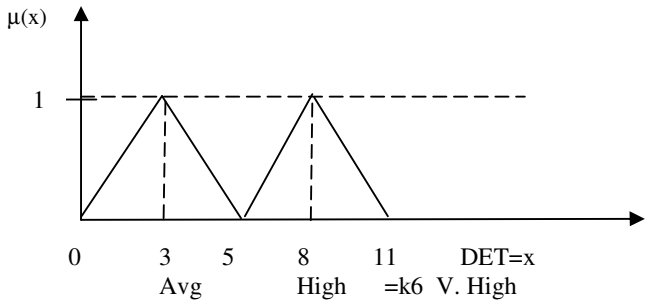


Fig 10:

DET	Complexity
1-15	Low
16-k7 $k_7 \leq 32$	Average
K_7+1 or more	High

Table 11: Modified Complexity Matrix for EI (FTR = 1)

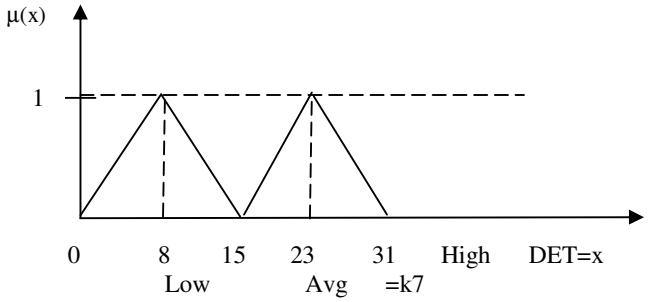


Fig 11:

DET	Complexity
1-4	Low
5-15	Average
16-k8 $k_8 \leq 32$	High
K_8+1 or more	V. High

Table 12: Modified Complexity Matrix for EI (FTR = 2)

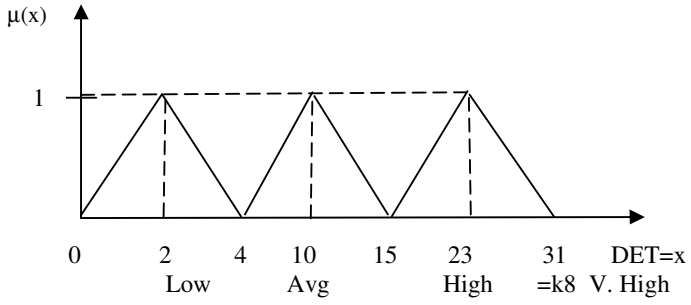


Fig 12:

DET	Complexity
1-4	Average
5-k9 $k_9 \leq 10$	High
K_9+1 or more	V.High

Table 13: Modified Complexity Matrix for EI (FTR $\geq$ 2)

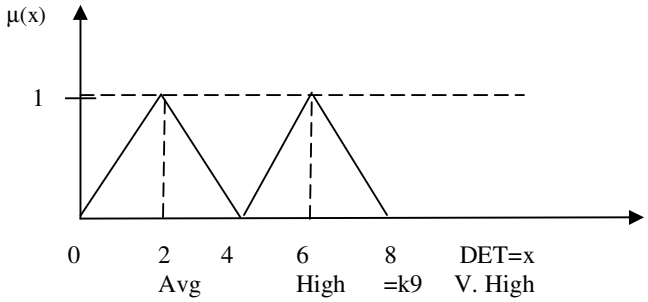


Fig 13:

Defuzzification:
Defuzzification rules for various data and transaction functions are given in the following tables.

Defuzzification for ILF and EIF

DET / FFP	1-25	25-50	50-70	70-90
ILF	μ*7	(μ*7)+ (1-μ)*10	(μ*10)+ (1-μ)*7	(μ*10)+ (1-μ)*15
EIF	μ*5	(μ*5)+ (1-μ)*7	(μ*7)+ (1-μ)*5	(μ*7)+ (1-μ)*10

Table 14: Case 1 for RET =1

DET / FFP	1-10	10-19	19-35	35-50	50-70	70-90
ILF	μ*7	(μ*7)+ (1-μ)*10	(μ*10)+ (1-μ)*7	(μ*10)+ (1-μ)*15	(μ*15)+ (1-μ)*10	(μ*15)+ (1-μ)*22
EIF	μ*5	(μ*5)+ (1-μ)*7	(μ*7)+ (1-μ)*5	(μ*7)+ (1-μ)*10	(μ*10)+ (1-μ)*7	(μ*10)+ (1-μ)*14

Table 15: Case 2 for $2 \leq$ RET ≤ 5

DET / FFP	1-10	10-19	19-29	29-38
ILF	μ*10	(μ*10)+ (1-μ)*15	(μ*15)+ (1-μ)*10	(μ*15)+ (1-μ)*22
EIF	μ*7	(μ*7)+ (1-μ)*10	(μ*10)+ (1-μ)*7	(μ*10)+ (1-μ)*14

Table 16: Case 3 for RET ≥ 6

Defuzzification for EO and EQ:

DET / FFP	1-10	10-19	19-29	29-38
EO	μ*4	(μ*4)+ (1-μ)*5	(μ*5)+ (1-μ)*4	(μ*5)+ (1-μ)*7
EQ	μ*3	(μ*3)+ (1-μ)*4	(μ*4)+ (1-μ)*3	(μ*4)+ (1-μ)*6

Table 17: Case 1 FTR < 2

DET / FFP	1-3	3-5	5-12	12-19	19-29	29-38
EO	μ*4	(μ*4)+ (1-μ)*5	(μ*5)+ (1-μ)*4	(μ*5)+ (1-μ)*7	(μ*7)+ (1-μ)*5	(μ*7)+ (1-μ)*10
EQ	μ*3	(μ*3)+ (1-μ)*4	(μ*4)+ (1-μ)*3	(μ*4)+ (1-μ)*6	(μ*6)+ (1-μ)*4	(μ*6)+ (1-μ)*9

Table 18: Case 2 FTR = 2 or 3

RET / FFP	1-3	3-5	5-8	8-11
EO	μ*5	(μ*5)+ (1-μ)*7	(μ*7)+ (1-μ)*5	(μ*7)+ (1-μ)*10
EQ	μ*4	(μ*4)+ (1-μ)*6	(μ*6)+ (1-μ)*4	(μ*6)+ (1-μ)*9

Table 19: Case 3 FTR > 3

Defuzzification for EI:

DET / FFP	1-8	8-15	15-23	23-37
EI	μ*3	(μ*3)+ (1-μ)*4	(μ*4)+ (1-μ)*3	(μ*4)+ (1-μ)*6

Table 20: Case 1 FTR < 2

DET / FFP	1-2	2-4	4-10	10-15	15-23	23-31
EI	μ*3	(μ*3)+ (1-μ)*4	(μ*4)+ (1-μ)*3	(μ*4)+ (1-μ)*6	(μ*6)+ (1-μ)*4	(μ*6)+ (1-μ)*9

Table 21: Case 2 FTR= 2

RET / FFP	1-2	2-4	4-6	6-8
EI	μ*4	(μ*4)+ (1-μ)*6	(μ*6)+ (1-μ)*4	(μ*6)+ (1-μ)*9

Table 22: Case 3 FTR > 2

Optimisation

Optimization of result for an application can be done on the basis any of the four criteria given in section 2, by varying one or more variables k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8 and k_9.

4. EXPERIMENTAL STUDY

The data for experimental study is taken from [18]. Calculation of Unadjusted Fuzzy Function points for real life application is given in tables 23 to 26.

K	DET	RET	μ	Count	FFP	FP
K_2=90	60	3	0.5	2	17.00	20
	75	3	0.75	1	11.00	10
			Total	3	28.00	30

Table 23: Calculation of FP and FFP for EIF

K	DET	FTR	μ	Count	FFP	FP
K_4=38	22	1	0.30	3	12.90	15
	10	2	0.71	4	18.86	20
	22	3	0.30	3	16.80	21
			Total	10	48.56	56

Table 24: Calculation of FP and FFP for EO

K	DET	FTR	μ	Count	FFP	FP
K_4=38	2	1	0.20	1	0.80	3
	21	1	0.20	3	12.60	12
K_5=38	1	2	0.33	1	1.33	3
	7	2	0.29	2	8.57	8
K_6=11	2	4	0.67	2	5.33	8
			Total	9	28.64	34

Table 25: Calculation of FP and FFP for EQ

K	DET	FTR	μ	Count	FFP	FP
K_7=31	2	1	0.25	3	2.25	9
	16	1	0.13	5	15.63	20
K_8=31	4	2	0.00	2	8.00	6
	7	2	0.50	3	10.50	12
	13	2	0.40	6	21.60	24
K_9=9	3	3	0.50	8	40.00	32
			Total	27	97.98	103

Table 26: Calculation of FP and FFP for EI

Comparison of Function Points using conventional and Fuzzy Technique:

	FP	FFP
ILF	0	0
EIF	30	28.00
EO	56	48.56
EQ	34	28.64
EI	103	97.98
Total UFP	**211**	**203.17**

Table 27

	UFP	VAF	Total
FP	211	1.13	238.43
FFP	203.17	1.13	229.58

Table 28

5. CONCLUSION AND FUTURE RESEARCH

The proposed study extends function point analysis to fuzzy function point analysis, using triangular fuzzy numbers. In FPA linguistic terms are used for some ranges of DET for which function points are considered to be the same. Of course they vary throughout these ranges. By using trapezoid shaped fuzzy numbers the problem is solved to some extent. We get better results then FPA by using Trapezoid shaped fuzzy numbers for linguistic terms for DETs in the border areas while for a considerable middle part of the range represented by linguistic term, the problem is not solved. In the proposed study triangular fuzzy numbers are used for linguistic terms, with the help of which we get variation of function points throughout the range represented by a linguistic term. Surely we must get better results. The methodology of fuzzy sets used for, in the proposed study, is sufficiently general and can be applied to other areas of quantitative software engineering.

6. REFERENCES

1. Alaa F. Sheta, “Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects”. Journal of Computer Science 2(2):118-123, 2006.
2. Bailey, J.W. and Basili, “A Meta model for software development resource expenditure”. Proc. Intl. Conf. Software Engineering, pp: 107-115, 1981.
3. Boehm, B., “Software Engineering Economics”, Englewood Cliffs, NJ. Prentice-Hall, (1981).
4. L.A. ZADEH., “From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions”, Int. J. Appl. Math. Computer Sci., Vol.12, No.3, 307-324., 2002.
5. L.A. ZADEH, “Fuzzy Sets, Information and Control”, 8, 338-353, 1965.
6. Roger S. Pressman, “Software Engineering; A Practitioner Approach”, Mc Graw-Hill International Edition, Sixth Edition (2005).
7. Ali Idri , Alain Abran and Laila Kjiri, “COCOMO cost model using Fuzzy Logic”, 7th International Conference on Fuzzy Theory & Technology Atlantic, New Jersy, 2000.
8. Emilia Mendes, Nile Mosley, “Web Cost Estimation: An Introduction, Web engineering: principles and techniques”, Ch 8, 2005.
9. J.E. Matson, B.E. Barrett, J.M. Mellichamp, “Software Development Cost Estimation Using Function Points”, IEEE Trans. on Software Engineering, 20, 4, 275-287, 1994.

10. Harish Mittal, Pradeep Bhatia, “Optimization Criterion for Effort Estimation using Fuzzy Technique”. CLEI Electronic Journal, Vol. 10 Num. 1 Pap. 2, 2007.
11. Musílek, P., Pedrycz, W., Succi, G., & Reformat, M., “Software Cost Estimation with Fuzzy Models”. ACM SIGAPP Applied Computing Review, 8(2), 24-29, 2000.
12. W.Pedrycz, J.F.Peters, S. Ramanna, “A Fuzzy Set Approach to Cost Estimation of Software Projects”, Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center, Edmonton Alberta, Canada, 1999.
13. A. J. Albrecht, “Measuring application development productivity”, SHARE/GUIDE IBM Application development Symposium.
14. V. R. Basili, K. Freburger, “Programming Measurement and Estimation in the Software Engineering Laboratory”, Journal of System and Software, 2, 47-57, 1981.
15. B. W. Boehm et al., “Software Cost Estimation with COCOMO II”, Prentice Hall, (2000).
16. Lima O.S.J., Farias, P.P.M. Farias and Belchor, A.D., “A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments”, CLEI EJ 5 (2), 2002.
17. Jones, C., 1996, “Programming Languages Table”, Release 8.2, March
18. Chuk Yau, Raymond H.L. Tsoi, “Assessing the Fuzziness of General System Characteristics in Estimating Software Size”, IEEE, 189-193, 1994.

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA