Multilevel Index Algorithm in Search Engine

[1]Laxmi Ahuja, [2]Dr Ela Kumar
[1]Amity Institute of Information Technology, Amity University, Uttar Pradesh Noida
[2] School of Information Technology Gautam Buddha University,
[1]laxmiahuja@aiit.amity.edu, [2] ela_kumar@rediffmail.com

ABSTRACT

In case of single level algorithm that is used for block level estimations; computational requirements have increased many-folds. This has introduced the need for multi-level search algorithms for real-time implementations of the video coding standards that can be used for searching. With this objective in mind, our present work aims at information overloading and personalization of characteristics for user information requirement and thus introduces a fast multi-level search algorithm. A multilevel search algorithm has been included in this work. The concept of multi-level search along-with its; benefits and implementation have also been included in this. Better performance and stronger scalability has been achieved through multi-level index structure and thus the algorithm can be applied to large-scale information filtering system.

Keywords *Indexing, Single Level Search Algorithm, Multilevel random search Algorithm*

I. INTRODUCTION

In order to facilitate fast and accurate information retrieval, Search engine indexing collects, parses, and stores data. Concepts from different disciplines like linguistics, cognitive psychology, mathematics, informatics, physics and computer science have been incorporated into the Index design. Alternatively named as Web Indexing, the process is used in the context of search engines designed to find web pages on the Internet. Other available engines focus on online and natural language documents, even to the extent that media types like video, audio and graphics are also searchable.

Cache-based search engines permanently store the index along with the corpus, whereas Meta search engines reuse the indices of other services and do not store a local index. The depth indexed is restricted in order to reduce the index size in partial-text services, unlike the full-text services. Agent-based search engines index in real time, whereas indexing is performed at a predetermined time interval by larger services due to the required time and processing costs. Speed and performance are optimized in finding relevant documents for a search query by storing an index; without which the search engine would scan every document in the corpus, and that would require considerable time and computing power. Several issues related with the indexing in search engine are discussed in the present paper and it produces an algorithm which is based on multi-level indexing.

II. INDEXING

As per the Random House Webster's Dictionary's definition, Indexing is defined as an "alphabetical list of names, places, and topics with the page numbers on which they are mentioned." Another definition given by the American National Standards Institute (ANSI) is "A systematic guide to items contained in or concepts derived from a collection". Now based on the complexity of the problem, it can be single level or multi level. These are explained as:

III. SINGLE LEVEL SEARCH ALGORITHM

All the candidate blocks in a search window need to be exhaustively searched in order to find the best matching block, in the single level search algorithm for the motion estimation process. This exhaustive search process becomes, however, very time consuming and hence is not suitable for complex applications having huge database. Thus, multilevel search algorithms were proposed in the literature [1, 3, 7, 8]. In order to reduce the computational complexity of motion estimation, these algorithms try to decrease the block matching criterion as much as possible.

IV. MULTI LEVEL SEARCH ALGORITHM

First Level Index can be categorized into semantic category that has been decided based upon the content of data itself, viz., name, designation, sex etc. The second level index is a subset of the category included in the first level index. In the given table, the designations are arranged for every department in the ascending order. Therefore, while filtering, system finds the classes relevant with incoming documents starting from the first level index and then it enters into the second level index and so on [4, 5].

First Level	Second Level	Data
Department	**Designation**	**Name**
HR	Executive Manager	Mr. X Mr. A
Marketing	Manager Representative	Ms. X Mr. Y
…	…	…

Two approaches for this purpose are often employed in the literature, and they can be summarized as follows.

- **Selection of the search locations in every step of the motion estimation**

The algorithms in this approach divide the motion estimation process into several steps. At every search step, a number of search locations are checked. The search location with the minimum mean absolute difference is the center of the next search process. The three-step search algorithm the four-step search algorithm is some of the algorithms belonging to this approach [1,2].

- **Use of lower bounds for the block matching criterion**

The use of lower bounds for the block matching criterion can reduce significantly the number of times needed to compute the block matching criterion without any loss of accuracy, since often the computation of this lower bound already indicates that the current search location is not a better one. The selective elimination algorithm, the multi-level selective elimination algorithm**,** and the vector based algorithm are some of the algorithms belonging to this approach [3].

Here, both of the above approaches are employed to reduce the number of times necessary to compute the block matching criterion and thus reduce the computational complexity of the motion estimation.

There is lot many application area of multi level indexing. Jian Zhai el. al [6] used this concept in rich media. Because of the potentially very large index size, it is hard to adopt and adapt content-based method to search rich media files on the Web. In this work, authors describe a multi-level indexing method to solve this problem by proposing a novel technique of indexing rich media at different levels in a large collection.

V. ALGORITHM ASSUMPTIONS:

If system avoids comparison of irrespective topics and their respective profile lists due to the fact the IF system finds the classes relevant with the incoming document from the first level index and then subsequently enters the second level index. So, each document has the knowledge *p(d/t)* about relevant topics, hence the comparison does not need to be continued when the user interesting degree $i_u(d)$ for the incoming document is less than the threshold of one item in the profile list, since the value $i_u(d)$ will not satisfy the threshold requirement of profiles in the latter case. Therefore, the computing quantity can be reduced more, and at the same time the system performance is increased without affecting the precision and recall.

With these underlying assumptions, we come up with the following filtering algorithm:

The comparison will go on so long as there is any class item of the first level index that is still left to be compared with the current incoming document. If a match is found that shows that the current incoming document belongs to the class being compared, then we enter into the second level index list pointed to by the current class. Within this, we go on comparing the topics till there is no topic item left to be compared with the current incoming document. If a match is found that shows that the current incoming document fits for the current topic, then we enter into the profiles list pointed to by the current topic. Within this, we go on comparing the profiles till there is no profile left to be compared with the current incoming document. Now, for each profile, we calculate the user interesting degree as obtained by the current profile for the current document as per the formula [7]:

$$i_u(d) = p(t/u)\, p(d/t)$$

If we get $i_u(d)$ as greater than or equal to the threshold of the current profile , then we select this document as a match and return as our selected output.

Based on the above analysis, the user provides the filtering algorithm as follows:

```
Start of algorithm

  While (There is a class item of the first level index that has not been compared (with current_ incoming document)
      Take the next class that has not been compared to be the current class
      If (The current document belongs to the current class) Then
          Enter into the second level index list pointed to by the current class
          While (There is a topic item that has not been compared with the current incoming document)
              Take the next topic that has not been compared as the current topic
              If (The topic of the current document fits for the current. topic) Then
                  Enter into the profiles list pointed to by the current topic
                  While (There is a profile that has not been compared with the current incoming document)
```

```
                    Take the next profile to be the current. profile
                    iu (d) = p(t/u)p(d/t)  (Calculate the user interesting degree described by the current_
                                    profile for the current document)
                   if (iu (d)= > the threshold of the current. profile) Then
                        Select this document
                   Else
                        Exit  ‘  from the procedure
                 End If
              Wend
           End If
         Wend
   End If
Wend
‘ End of algo
```

```
    If (The topic of the current document fits for the current. topic) Then
       {     Enter into the profiles list pointed to by the current topic;
    While (There is a profile that has not been compared with the
    current incoming document)
       {     Take the next profile to be the current. profile;
            iu (d) = p(t/u)p(d/t); (Calculate the user interesting degree
              described by the current profile for the current document)
              if (iu (d)= > the threshold of the current. profile) Then
            {      Select this document;
                 }
                  Else
                       Exit;
            }}}}}
      End
```

Using multi-level indexes, we can reduce the number of block accesses required to search for information given its indexing field value.

VI. CONCLUSION

In case of single level algorithm that are used for block level estimations, computational requirements have increased many-folds. This has introduced the need for multi-level search algorithms for real-time implementations of the video coding standards that can be used for searching. With this objective in mind, our present work aims at information overloading and personalization of characteristics for user information requirement and thus introduces a fast multi-level search algorithm. A multilevel search algorithm has been included in this work. The concept of multi-level search along-with its; benefits and implementation have also been included in this. Better performance and stronger scalability has been achieved through multi-level index structure and thus the algorithm can be applied to large-scale information filtering system.

In this paper, a fast three-step search algorithm has been discussed, which can be applied to large-scale information filtering system and provides better performance and stronger scalability. However, the proposed work still needs some experimentation evaluation of performance improvement with the proposed technique.

REFERENCES

[1] Sung-Tae Jung, Sang-Seol Lee. “A 4-way Pipelined Processing Architecture for Three-Step Search Block-matching Motion Estimation”. IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, pp.674-681 MAY 2004.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T.Ishiguro, “Motion compensated interframe coding for video conferencing,” in Proc. NTC81, pp. C9.6.1-9.6.5, New Orleans, LA, Nov./Dec. 1981.

[3] L. M. Po and W. C. Ma, “A novel four-step search algorithm for fast block matching,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, no.3, pp. 313–317, Jun. 1996.

[4] L. K. Liu and E. Feig, “A block-based gradient descent search algorithm for block-based motion estimation in video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 4, pp. 419–422, Aug. 1996.

[5] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block matching,” IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 2, pp.287–290, Feb. 2000.

[6] Jian Zhai1 Jun Yang, Qing Li, Liu Wenyin, Bo Feng, Rich Media Retrieval on the Web – a Multi-level Indexing Approach, at http://www2003.org/cdrom/papers/poster/p176/WWW03-Poster-Final-submit.htm

[7] He Jun, Zhou Mingtian, Multilevel index structure for information filtering based on user characteristic, Journal of Electronics, Vol.18 No.3, July 2001, pp:267-271.

[8] Shinya Fujiwara and Akira Taguchi, "Motion-Compensated Frame Rate Up-Conversion Based on Block Matching Algorithm with Multi-Size Blocks," Proceedings of 2005 International Symposium on

Intelligent Signal Processing and Communication Systems, December 2005:353-356.

[9] A fast three-step search algorithm by the utilization of multilevel vector partial sums, Canadian Conference on Electrical and Computer Engineering, 2003. IEEE CCECE 2003, vol.3, pp: 1981 – 1984

[10] An improved multilevel successive elimination algorithm for fast full-search motion estimation, Proceedings of International Conference on Image Processing, 2003 (ICIP 2003), Vol.3, pp: II - 351-4