Elimination of Repeated Occurrences in Multimedia Search Engines

Saed Alqaraleh and Omar Ramadan
Department of Computer Engineering, Eastern Mediterranean University, North Cyprus

Abstract: *In this paper, we have proposed a new method for eliminating repeated occurrences in multimedia search engines. We have built software that extracts information from multimedia databases which will compare these multimedia files and marks only one copy of repeated files. The developed software can work with any search engine and can also work in a routine manner to deal with any updates on the databases. Moreover, the software allows multiple copies to be executed in parallel and consequently it improves the efficiency of multimedia searching.*

Keywords: *Search engines, multimedia search engines, information retrieval.*

Received November 9, 2011; accepted December 30, 2012; published online January 29, 2013

1. Introduction

In the last few years, the number of multimedia files and applications, based on access multimedia on the internet, has considerably grown. With huge number of multimedia files, it is hard to find the required files easily for many reasons. First, the search engines are still using metadata or keywords to create multimedia databases. Metadata can't deal with different meanings of words and there may be no relation between the contents of the multimedia and their names. For example, when one uses a camera for taking images or recording a video, the camera generates random (automatic) names for those files, with no relation with the multimedia content. Second, when the user doesn't know how to describe the required files, it is hard to find out the multimedia s/he is trying to find. Finally, information redundancy requires extra time for checking the results.

In recent years, several developments in this field have been presented. In [4, 6], special search engines working on extracting information from files (content based retrieval) have been developed. In [14], query by example technique has been improved. When, one wants to get files similar to what she/he already has, the query by example technique is more powerful. In [15], another method called query by sketch has been developed to improve the performance of the query by example approach. This method gives the user more options, like using drawing tools for describing what exactly is required. In [17], search engines have been improved by introducing a 3D model retrieval technique based on 3D factional Fourier transform. It has been observed that the developed software has improved the chance of getting the required files. In [9], a semantic approach for multimedia documents has been proposed. In this approach, multimedia documents have been presented based on different platforms and conceptual neighborhoods graphs. The conceptual neighborhood graph mechanism has been presented to find the similarity degree of the relations.

Recently, it has been found that around 40% of web pages and files on the internet are duplicated [8]. In [2], new software for eliminating the repetition in image search engines has been introduced. The developed software creates an image database and then it calculates the hash values [2] for each image, and finally, it compares these hash values to find repetitions, and marks only one copy of repeating files for further use. In [16], a framework for eliminating near-duplicate videos on social web has been developed. This framework combines the contextual information with the content of the files to find the near-duplicate videos and eliminate them from the top rank list. In this framework, the comparison process will be done on each query, which will increase the query time. Also, this framework cannot deal with other types of multimedia. To the best of our knowledge, there is no research done on eliminating the repetition in multimedia search engines.

The main objective of the work presented in this paper is to improve the efficiency of multimedia search engines by eliminating repeated occurrences. The software can deal with any type of multimedia files like images, video and audio. Also, the performance of the developed software can be improved by running multiple copies at the same time. The developed software will eliminate repetition during creating or through updating process of the database. Moreover, by using the developed software, the comparison process will be done only once. and hence, the query time will be decreased.

This paper is organized as follows: Section 2 presents the methodologies used in the developed

software. Discussion of the developed software is presented in section 3. Simulation study is presented in section 4 and finally, conclusion is given in section 5.

2. Methodology

The software developed in this paper uses the following two techniques for performing the comparison process: The Hash [3, 13] and the feature extraction approaches [7].

2.1. Hash Algorithms

Hash algorithms are cryptography functions that take any information as input and convert it to a numeric code. The outputs of these algorithms are unique for each file, and it is like a fingerprint. By using Hash Algorithms, files can be compared in fewer amounts of data [3, 13]. In this paper, a special Hash Algorithm called Message-Digest 5 (MD5) [13] is used to check if the files being compared are same or not. MD5 Hash algorithm was chosen because it has the following advantages: The message size can be infinite, and the output of hashing is small in size (16 bytes) as compared to other Hash algorithms like SHA265 and SHA512 [1].

2.2. Feature Extraction

Feature extraction refers to the process of getting a set of features (useful information) from input data. In the proposed software, we have extracted low level information from the multimedia files themselves by using Multimedia Content Description Interface feature (MPEG-7). In this respect, the file is segmented into regions and the main features are extracted based on the following descriptors of information: Layout, color structure, dominant color, scalable color, edge histogram, homogeneous texture, contour shape and region shape. These descriptors of information have been used to complement each other and to describe the object appearance in detail.

3. The Software Developed

The software developed in this paper can work and create multimedia databases and also can be connected with the search engine databases. In addition, it can extract and compare information from multimedia databases to keep only one copy of repeated files.

As it is well known, the ranking of any website depends on the number of hits, the keywords, the website meta-tags and the contents of the website [11]. The website with a high rank will be shown at the beginning of the listed results. In the developed software, and in order to keep the ranking position of websites, we do not physically delete repeated files from the database. Instead, a flag field is added to the database. This flag is set to 1 for the first file in the repeated list, and set to zero for all others files. In the following, the algorithm for creating and/or adding files to search engines databases using the developed software are summarized:

```
// N=Number of files which will be added to the database
For i=1 to N do
  Get the multimedia file (i)
   Flag (new file (i))=1
   //Create the file properties (hash value using MD5 or
     extract low level features using MPEG-7).
     If (Type of (new file (i)) =Image)
        Properties (new file (i)) =MD5 (new file (i))
      Else
        Properties (new file (i)) = extract the features (new file
       (i))
     End If
  For j=1 to Y do //Y=Number of files in the database
    Get the file properties.
      If properties (newfile(i))= properties(file(j)) then
        Make flag= zero
        Exit
      End If
  End for
Save new file (i), Flag (new file (i)) and
Properties (new file (i)) in the database.
End for
```

In order to improve the efficiency of the proposed searching process, many copies of the developed software are allowed to work in parallel. In this case, the number of multimedia files is divided evenly between the parallel copies. The server administrator decides on the number of copies depending on the total number of multimedia files in the database.

Finally, it is important to note that we have developed a second version of the software which will compute the Hash value or extract the low level features based on the mechanism used for each file during the multimedia database creation. If there are any similar files in the database, flag zero will be assigned otherwise the flag value will be set to one. In the query process, the system will list those files with flag equals one only.

4. Simulation Study

The developed software was tested by creating an artificial database by randomly taking multimedia files from internet search engines. Three types of databases are created: image, video, and audio databases as described below.

4.1. Working with Image Database

In this type of database, bit-wise [5] and Hash Algorithms have been used in the comparison process between images. The bit-wise comparison technique compares all pixels of two images one by one and if all pixels in both images are the same, then only one image will be considered in a later search. On the other hand, the Hash Algorithms compare only limited number of bits. Figure 1 shows the execution time

versus number of images for the Hash and the bit-wise approaches. It is clear from Figure 1 that the bit-wise method is not efficient for large image databases. On the hand, the hash technique was found to be much faster than the bit-wise comparison technique, especially for a large number of images in the database.

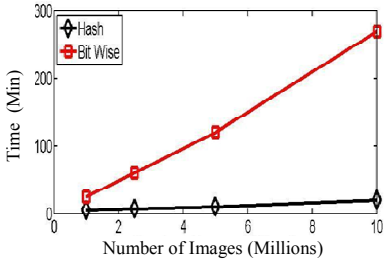

Figure 1. Execution time versus number of images for the hash and bit-wise approaches.

In the next test, we have improved the performance of the hash algorithm by using a parallel technique. This technique allows running multiple copies of the developed software at the same time. Figure 2 shows the execution time versus number of images for 4, 8, 12, and 16 copies using hash technique. It can be seen from Figure 2 that the parallel technique decreases the search time and hence improves th efficiency of the algorithm.

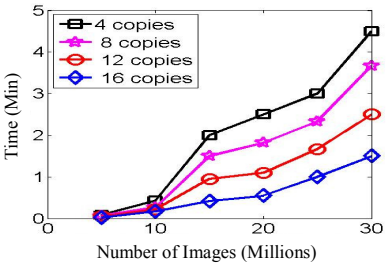

Figure 2. Execution time versus number of images for 4, 8, 12, and 16 copies as obtained by using the hash technique.

4.2. Working with Video and Audio Database

The performance of the developed software for dealing with video and audio databases has also been studied. In this test, multiple copies of the developed software are allowed to work in parallel. Figure 3 shows the execution time versus number of video and audio files for 4, 8, 12, and 16 copies as obtained by using the Hash algorithm.

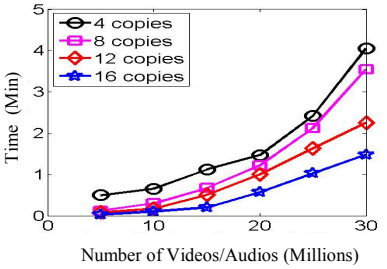

Figure 3. Execution time versus number of video and audio files for 4, 8, 12, and 16 copies as obtained by using the hash algorithms.

It is clear from Figure 3 that the software is capable of dealing with video and audio databases efficiently. The performance of the low-level feature extraction and the Hashing approaches has also been studied. Figure 4 shows that the time required for creating the database using the Hash technique is larger than that of the low level extraction technique.

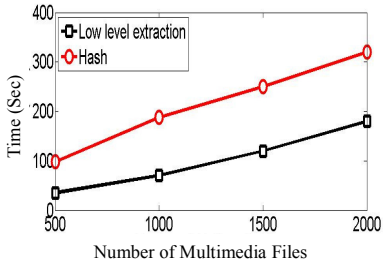

Figure 4. Execution time required for creating databases by using the Hash and the low level extraction approaches.

On the other hand, and during the comparison process, the Hash technique was found to be faster in the comparison process than the low level extraction technique as can be seen from Figure 5.

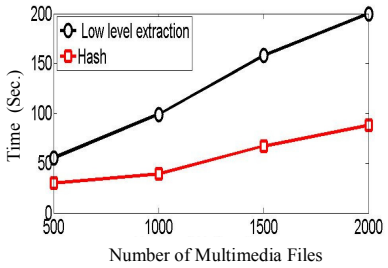

Figure 5. Execution time required for the comparison process by using the hash and low level extraction approaches.

Figure 6 shows the execution time versus number of files for the Hash and the low level extraction algorithms. It is clear that both techniques have approximately the same performance when the total time is compared. Moreover, it should be noted that both of these techniques are capable of dealing with large number of files in the database within an acceptable time.

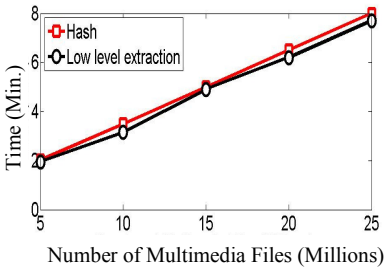

Figure 6. Execution time versus number of file using the hash algorithms and low level extraction.

Figure 7 outlines the results of the first and the second version of the software. It is clear that the performance of the developed software has significantly improved through the second version.

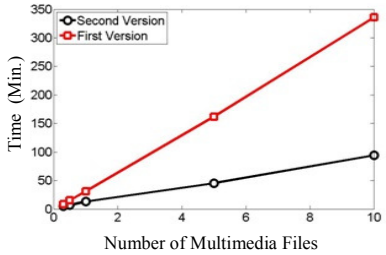

Figure 7. Execution time for the first and second version of the developed software.

Table 1 shows the execution time for the hash and the low level extraction approaches for different number of multimedia files using second version of the developed software described in section 3.

Table 1. Execution time for the hash and the low level extraction approaches by using the second version of the developed software.

Number of Files (Million)	Total Time (Minutes)	
	Low Level Extraction	Hash Technique
3	2	6
5	5	14
10	9	25
50	30	115
100	61	220

As it is known, the websites have dynamic information process which means that the contents of a website is updated and modified in a routine manner. Consequently, the search engines have to update their databases in a similar manner. In this paper, the developed software works in a routine manner to deal with databases updates. We have done an experiment to find the required time for updating the database. Three databases have been created for this purpose: the first one contains ten million multimedia files, the second contains one hundred million files and the third one contains one billion files. Table 2 shows the execution time for updating the databases. It is clear from Table 2 that the developed software can deal with large database and keep updating the database to eliminate the repeated occurrences of multimedia files within acceptable execution time.

Table 2. Execution time for updating the database.

Number of Files in the Database (Million)	Number of Files in Updated Process (Million)	Time (Minutes)				
		One Copy	32 Copies	64 Copies	128 Copies	256 Copies
10	5	5	1	0.45	0.23	0.15
	7,5	8	1.89	0.98	0.51	0.27
	10	11.5	2.2	1.23	0.69	0.41
	15	16.7	3.2	1.7	0.91	0.57
	50	48.1	8.91	4.76	2.67	1.41
100	5	22	4	2	1	0.5
	7,5	33	6	3	1.6	0.8
	10	47	9	4.8	2.5	1.8
	15	70	16	8.7	4.7	2.5
	50	191	81	42.5	23	11
1000	5	185	35.3	18	10	4.9
	7,5	285	52.7	27.1	14.6	8
	10	371	68.2	37.9	18.4	10.1
	15	542	101	50.5	26.1	14
	50	1235	220.7	113	59	30.8

Also, we have done an experiment to compare the efficiency of current mechanism of Google with and without using the developed software. In this experiment, we have used the most frequently used search keywords as shown in Table 3, which we have obtained from [10, 12] in the first week of July/2011. After executing every query, we have tried to find the percentage of relevant files, the percentage of repeated occurrences and the query execution time. Two humans found the percentage of relevant files of the results. The percentage of repeated occurrences and the query execution time were found using the developed software. For more information, the second part of the experiment was done to test the efficiency of the developed software with Yahoo and Bing search engines. Unfortunately, Yahoo and Bing search engines do not show the required time for the search process. Hence, we have compared only the percentage of relevant files and the percentage of repeated occurrences. Table 3 shows time required for the current mechanism of Google with and without using the developed software. Table 4 shows the efficiency for current mechanism of Yahoo and Bing Search engines with and without using the developed software.

Table 3. Time required for current mechanism of Google with and without using the developed software.

Keywords	Google (without using the Developed Software)				Google (by using the Developed Software)		
	Number of Results (Millions)	Percent of Repeated Occurrences	Required Time (Seconds)	Percent of Relevant Files	Number of Results (Millions)	Required Time (Seconds)	Percent of Relevant Files
Images							
Hotels	140	35%	0.39	88%	91	0.25	91%
Cars	158	32%	0.20	92%	108	0.14	95%
Dog	130	39%	0.10	89%	80	0.06	94%
Girls	209	40%	0.36	91%	126	0.24	97%
Weather	93,6	39%	0.76	88%	58	0.46	92%
Videos							
Videos	78,6	30%	0.08	92%	55	0.06	96%
Funny videos	16,2	33%	0.09	80%	11	0.56	85%
Dance movies	3,7	32%	0.06	82%	2.5	0.04	87%
Movie trailers	3,7	29%	0.19	80%	2.7	0.14	84%
Weight loss	0.42	25%	0.08	80%	0.32	0.06	88%
Audios							
Music	14,8	38%	0.10	85%	9.2	0.08	89%
Songs	6,31	41%	0.15	88%	3.8	0.09	92%
Albums	4,51	40%	0.23	80%	2.8	0.14	86%
MP3	5,52	39%	0.14	84%	3.4	0.09	89%
Clips	5,6	35%	0.15	82%	3.7	0.10	86%

Table 4. The efficiency of current mechanism of Yahoo and bing search engines with and without using the developed software.

Keywords	Yahoo (without using the Developed Software)			Bing (without using the Developed Software)			Yahoo and Bing (by using the Developed Software)	
	Number of Results (Millions)	Percent of Repeated Occurrences	Percent of Relevant Files	Number of Results (Millions)	Percent of Repeated Occurrences	Percent of Relevant Files	Number of Results (Millions)	Percent of Relevant Files
Images								
Hotels	29,7	33%	86%	24,1	30%	85%	18,3	89%
Cars	59,8	34%	92%	53,2	37%	95%	35,8	96%
Dog	31,3	35%	87%	28,2	39%	89%	18,6	93%
Girls	41,6	36%	90%	41,3	29%	93%	26,8	94%
Weather	26,7	31%	82%	24,3	33%	80%	17,5	88%
Videos								
Videos	99,5	32%	91%	111	33%	90%	67,5	95%
Funny videos	12,8	31%	82%	12,7	30%	85%	8,4	87%
Dance movies	0.188	35%	80%	0.185	36%	82%	0.124	84%
Movie trailers	0.851	25%	80%	0.911	27%	82%	0.65	84%
Weight loss	0.242	29%	82%	0.236	25%	83%	0.18	85%
Audios								
Music	36,5	40%	84%	34,6	37%	80%	21,7	86%
Songs	2,25	37%	85%	2,2	39%	83%	1,3	88%
Albums	76,8	35%	79%	63	30%	81%	48,3	82%
MP3	0.77	36%	80%	0.75	32%	87%	0.49	88%
Clips	6,3	33%	81%	5,2	31%	80%	4	86%

It can be seen from Tables 3 and 4 that the output list contains a very large number of files and around 40% of those files are duplicated. On the other hand, by using the developed software, the percentage of duplicated files becomes zero. In addition, the required time during the query execution is decreased. Finally, it is important to note that by removing the repeated files, the chance of getting more relevant file will be increased.

5. Conclusions

In this paper, new software has been developed to improve the efficiency of multimedia searching by eliminating the repeated occurrences of multimedia files. The developed software can work with any search engine, and can work periodically on multimedia databases. Moreover, it allows multiple copies to be executed in parallel, and hence, it can improve the search time for multimedia files.

References

[1] Alqaraleh S., *Elimination of Repeated Occurrences in Image Search Engines*, *Technical Report*, Eastern Mediterranean University, North Cyprus, 2011.

[2] Aybay I. and Alqaraleh S., "Elimination of Repeated Occurrences in Image Search Engines," *in Proceedings of the 9th International Conference on Application of Fuzzy Systems and Soft Computing*, Prague, Czech Republic, pp. 145-149, 2010.

[3] Cheddad A., Condell J., Curran K., and McKevitt P., "A Hash-Based Image Encryption Algorithm," *Optics Communications*, vol. 283, no. 6, pp. 879-893, 2010.

[4] Doulaverakis C., Nidelkou E., and Kompatsiaris A., "A Hybrid Ontology and Content-Based Search Engine for Multimedia Retrieval," *in Proceedings of CiteSeerX-Scientific Literature Digital Library and Search Engine*, USA, pp. 1-13, 2008.

[5] Gonzalez R. and Woods R., *Digital Image Processing*, Prentice Hall, USA, 2008.

[6] Jin H., He R., Liao Z., Tao W., and Zhang Q., "A Flexible and Extensible Framework for Web Image Retrieval System," *in Proceedings of International Conference on Internet and Web Applications and Services/ Advanced*, China, pp. 193, 2006.

[7] Manjunath B., Salembier P., and Sikora T., *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley, New York, 2002.

[8] Manning C., Raghavan P., and Schutze H., *Introduction to Information Retrieval*, Cambridge University Press, USA, 2008.

[9] Maredj A. and Tonkin N., "Semantic Adaptation of Multimedia Documents," *the International Arab Journal of Information Technology*, vol. 10, no. 6, 2013.

[10] Multimedia Search Keywords, available at: https://www.wordtracker.com/, last visited 2011.

[11] Papastefanos D., Mateevitsi V., Andritsopoulos F., Achilleopoulos N., and Mikhalev V., "Video Index and Search Services Based on Content Identification Features," *in Proceedings of Broadband Multimedia Systems and Broadcasting*, Las Vegas, USA, pp. 1-4, 2008 .

[12] Search Keywords, available at: http://www.wordstream.com, last visited 2011.

[13] Stallings W., *Cryptography and Network Security: Principles and Practice*, Pearson Education, USA, 2003.

[14] Vermilyer R., "Intelligent User Interface Agents in Content-Based Image Retrieval," *in Proceedings of the IEEE*, Memphis, USA, pp. 136-142, 2006.

[15] Watai Y., Yamasaki T., and Aizawa K., "View-Based Web Page Retrieval using Interactive Sketch Query," *in Proceedings of International Conference on Image Processing*, San Antonio, USA, vol. 6, pp. 357-360, 2007.

[16] Wu X., Ngo C., Hauptmann A., and Tan H., "Real-Time Near-Duplicate Elimination for Web Video Search with Content and Context," *in Proceedings of IEEE Transaction on Multimedia*, Hong Kong, pp. 196-207, 2009.

[17] YuJie L., Feng B., ZongMin L., and Hua L., "3D Model Retrieval Based on 3D Fractional Fourier Transform," *the International Arab Journal of Information Technology*, vol. 10, no. 5, 2013.

Saed Alqaraleh received his BSc degree in software engineering from Al-Hussein Bin Talal University, Jordan, in 2008, and his MSc degree in computer engineering, Eastern Mediterranean University, Northern Cyprus, in 2011. He is currently a PhD student in computer engineering, Eastern Mediterranean University, Northern Cyprus.

Omar Ramadan received his BS MS and PhD degrees in electrical and electronic engineering, Eastern Mediterranean University, Northern Cyprus, in 1992, 1994, and 1999, respectively. He is currently working as a professor in Computer Engineering Department at the Eastern Mediterranean University. He has authored or co-authored more than 50 journal articles, 15 conference papers, and 5 book chapters. He has supervised one PhD and two MSc students and currently he is supervising two PhD and two MSc students.