IMAGE INDEXING SEARCH SYSTEM AND METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Serial No. - 60/454,315 filed on March 14, 2003 by inventors Tianlong Chen, Yi Rui, Yingbin Wang, and Yinong Wei.

[0002] Other related applications include the following:

[0003] Memory-Resident Database Management System and Implementation Thereof; U.S. Patent Application Serial Number 10/347,678; Filed on January 22, 2003; Inventors: Tianlong Chen, Jonathan Vu.

[0004] Distributed Memory Computing Environment and Implementation Thereof; U.S. Patent Application Serial Number 10/347,677, Filed on January 22, 2003; Inventors: Tianlong Chen, Jonathan Vu, Yingbin Wang.

[0005] Invariant Memory Page Pool and Implementation Thereof; U.S. Patent Application Serial Number 10/425,730; Filed on April 30, 2003; Inventors: Tianlong Chen, Yingbin Wang, Yinong Wei.

[0006] Central Linked List Data Structure and Methods of Use; Filed January 7, 2002, U.S. Patent Application Serial Number 10/346,260; Inventor: Jonathan Vu.

[0007] A Method and or System to Perform Automated Facial Recognition and Comparison Using Multiple 2D Facial Images Parsed from a Captured 3D Facial Image; U.S. Provisional Application No. 60/440,338 filed on January 16, 2003; Inventors: Donald A. Milne, III and Jonathon Vu.

[0008] Method and Apparatus for Facial Identification Enhancement; U.S. Patent Application Serial Number 10/635,565, filed on August 5, 2003; Inventors: Tianlong Chen,

Donald A. Milne, III, Yi Rui, Yingbin Wang, Jonathan Vu, And Yinong Wei.

[0009] Single Computer Distributed Computing Environment and Implementation Thereof; U.S. Patent Application Serial Number 10/425,729; filed on April 30, 2003; Inventors: Jonathon Vu, Tianlong Chen, Yingbin Wang, and Yinong Wei.

[0010] Integrated Portable Identification and Verification Device; U.S. Patent Application Serial Number 10/635,516 filed on August 5, 2003; Inventors: Donald Milne, III and Tianlong Chen.

[0011] The entirety of each of the aforementioned patent applications is incorporated by reference herein.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0012] Not applicable.

FIELD OF THE INVENTION

[0013] The present invention relates to devices, systems, and processes useful for fast one-to-many image matching and more particularly to an indexing method to dramatically improve image matching processing and its implementation.

BACKGROUND OF THE INVENTION

[0014] Image processing has been advancing dramatically. One imaging process involves 'one-to-many' matching, i.e., given one image, this process finds its matches from an image archive or image database of a large number of images. In order to maintain acceptably high accuracy, many one-to-many image matching methods use the given image to one-by-one compare the given image to every image in the image database or archive. This process requires an expensive server or server cluster to achieve fast matching speed if the image database has a very large volume of images, such as more than 5 million images. The current industrial record for one-to-many matching is about one million comparisons per second.

SUMMARY OF THE INVENTION

[0015] One aspect of the present invention relates to a method which is capable of achieving high speed without sacrificing accuracy, and can be implemented with less expensive hardware than required with prior systems and methods. Theoretically, systems and methods according to the present invention can achieve high speeds, for example finding a set of 'top matches' of a given image (if any) from an image archive including up to billions of images in about one second.

[0016] Aspects, features, and attendant advantages of the present invention will become apparent to those skilled in the art from a reading of the following detailed description of embodiments constructed in accordance therewith, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] The invention of the present application will now be described in more detail with reference to preferred embodiments of the architecture and method, given only by way of example, and with reference to the accompanying drawings, in which:

[0018] FIG. 1 illustrates a conceptual block diagram of an exemplary one-to-many image matching search.

[0019] FIG. 2 illustrates a conceptual block diagram of an embodiment of an exemplary one-to-many image indexing matching architecture.

[0020] FIG. 3 illustrates a conceptual block diagram of an exemplary embodiment of image indexing method.

[0021] FIG. 4 illustrates a conceptual block diagram of another exemplary embodiment of image indexing method.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0022] The present invention relates generally to architectures and methods for fast one-to-many image matching. Another aspect of the present invention includes architectures and methods of image indexing. Accordingly, the drawings and descriptions are to be regarded as illustrative in nature, and not as restrictive.

[0023] The present invention is particularly useful when all of the images that are to be compared can be digitized into an array of roughly independent, and preferably entirely independent, descriptors (or called eigenvalues for certain digitization algorithms), and each descriptor may represent one feature of the targeted object (the object, such as a human face, is the matching target) in images. This is true for most currently used recognition algorithms, in particular facial recognition applications, such as various variations of Principal Component Analysis (PCA) based facial recognition algorithms, including those variations that have Local Feature Analysis (LFA) or combination of PCA and LFA. Examples of such algorithms include, but the present invention is not limited to: Hancock, P.R. Baddeley, and L.S. Smith (1992), "The principal components of natural images," Network: Computation in Neural Systems 3, 61-70; U.S. Patent No. 5,164,992, Turk et al., "Face Recognition System"; and Kim, K. I., et al., "Facial Recognition Using Kernel Principal Component Analysis," IEEE Signal Processing Letters 9:2, pp. 40-42 (IEEE, Feb. 2002). Those of skill in the art are familiar with facial image recognition algorithms, including facial image recognition algorithms, and their implementations in computer executable formats, and therefore further discussion of such algorithms is not included herein.

[0024] Referring to the drawing figures, like reference numerals designate identical or corresponding elements throughout the several figures.

[0025] Turning now to the drawing figures, and referring to FIG. 1, a first problem to which the present invention presents a solution, as embodied in the functionality of the exemplary Image Matching Engine 2053, is to find several 'top matches' of a given image 2052 ("target image"). One target object, such as a human face 2056, from the target image is sought in an image database 2051 ("source image database"; images in the "source image database" are "source images") of a large number of images, in which each image may have the targeted object. A match is defined when the targeted object exists in an image from the image database 2051 and is "similar" (meaning that it has similarity above a threshold of, e.g., a certain percentage) to the targeted object in the target image. If no match within the similarity threshold is found in a source image, then "No match" 2054 is returned as the result of the comparison. If multiple matches are found, then those matches 2055 with the highest similarity (called "top matches") are returned. The number of desirable top matches is configurable, for example, by changing the similarity threshold.

[0026] As an example of similarity or proximity of a match, assume that one image with a human face on it can be digitized into an array of 100 descriptors. If one image from the source image database has 80 descriptors out of 100 which are close enough, meaning that the absolute difference of each corresponding descriptor from the target image and a source image is smaller than a predefined value, then this source image is said to be of "80% similarity" to the target image. This can also be referred to as the similarity of the targeted object, such as a human face. The actual definitions of "similarity" or "proximity" of different matching algorithms can be different within the present invention, because the present invention is not limited to the specific matching algorithm utilized.

[0027] "Similarity" is defined and can be calculated, as will be readily appreciated by those of ordinary skill in the art, and all images can be digitized into an array of a certain number of descriptors, such that the descriptors are at least roughly independent of each other, and each descriptor may roughly correspond to one feature (such as a right eye, a left eye, nose, etc., of a human being) of the targeted object in the images. Even for those descriptors which are not completely independent of each other, the present invention is still applicable.

[0028] In order to achieve high speed one-to-many matching, architectures and methods of the present invention preferably pre-process all images and digitize them into arrays of descriptors. Thereafter, all the descriptors are loaded into computer system memory for further processing. Therefore, if the source image database is large, the system memory requirement will be huge. Another optional aspect of the present invention is the use of an Extended Memory Management Unit to exceed any limitation that the underlying Operating System may impose, as described in the aforementioned patent applications "Distributed Memory Computing Environment and Implementation Thereof" and "Memory-Resident Database Management System and Implementation Thereof", above.

[0029] Another aspect of the present invention is pre-separation of images into different groups based on the characteristics of the digitized descriptors of the images, and preferably only to conduct one-by-one matching searches in the qualified groups that have similarity in higher levels.

[0030] One observation in process image matching is that if the targeted human face from two images are from same person, a large percentage of the digitized descriptors from the two images are close to each other, meaning that the absolute differences of a large percentage of

corresponding descriptors are less than a pre-selected small number.

[0031] Assume the number of descriptors from an image is M, the number of descriptors from the target image is Ai (with $1 \leq i \leq M$), the number of descriptors from the source image is Bi (with $1 \leq i \leq M$), and δ is a pre-selected small positive number. Then, if two images are similar, the following equation holds for a large percentage of descriptors from the two images,

$$|A_i - B_i| < \delta \tag{1}$$

[0032] As discussed above, the "difference" can be defined differently for different matching algorithms; however, equation (1) holds essentially true for most matching algorithms with independent or roughly independent descriptors, and each descriptor corresponds to one feature of the targeted object, e.g., a human face.

[0033] Another observation from matching algorithms found by the present inventors is that the average value of a group of selected descriptors have a similar equation for two "similar" images, as below:

$$\left| \frac{\sum_i A_i}{X} - \frac{\sum_i B_i}{X} \right| < \delta \tag{2}$$

where 'i' are indexes of selected descriptors, δ is a pre-selected small positive number, and X is the number of selected descriptors in a group.

[0034] The present inventors have found that a one-to-many matching algorithm can achieve a speed of one million matchings per second on a 2.4 GHz P4 (Pentium™ 4, Intel) server; therefore, the algorighm can finish a one-to-35,000 image matching (actually, 35,000 one-to-one matchings) in 35 milliseconds using entirely system memory-based matching. With more sophisticated and complicated matching algorithms, a one-to-35,000 image matching can be finished faster yet, for example in less than 35 milliseconds. However, if the image database is of millions of images, the matching speed will be long, and furthermore may require large system memory.

[0035] According to the present invention, however, image indexing can be used to select images with high similarity, from millions or even billions of images, in short time periods,

e.g., 500 milliseconds, for fine one-to-one matching, and limit the final one-to-one matching to about a small number of images, e.g., 35,000 images. This is because it has already been established that this latter comparison can be finished in about 500 milliseconds, and thus the whole matching process can still be finished in a short time period, e.g., one second. Of course, these time values are dependent on processing speed and other parameters, as well known to those of skill in the art, and the present invention is not limited to these values. As also well known to those of skill in the art, an indexing process in a database is a relatively fast process, provided that the indexing method is good, and thus the present invention is capable of achieving one to up-to-billions image matching searching with reasonably high accuracy in one second on a readily commercially available regular powerful server, without requiring the use of a supercomputer.

[0036] Referring to FIG. 2, benefiting from the above observations, one aspect of the present invention includes an indexing image matching search architecture capable of achieving one-to-billion image matching with reasonably high accuracy.

[0037] For ease of explanation, facial recognition matching algorithms will be used herein as an example for description of the invention. As will be readily appreciated by those of skill in the art, the present invention is not limited to facial recognition, and applies equally to other image types. Current facial recognition algorithms are mostly based on various variations of PCA (Principal Component Analysis), which represent human face features with eigenvalues or descriptors. The number of descriptors typically ranges from about 40 up to several thousands. Two similar faces have a high percentage of corresponding descriptors close to each other, as discussed above.

[0038] In a simple case, each descriptor is chosen as one index in the image matching search. However, there are two reasons for normally not choosing each descriptor as an index: the number of indexes will become too large and degrade the speed of indexing; and, the noise in each image can result in unusual fluctuation of descriptor value.

[0039] The above equation (2) provides a method of indexing, using the average value of a group of selected descriptors as index, as illustrated in FIGs. 3 and 4 as examples. In FIG. 3, the average value of every four consecutive descriptors is selected as one index 2031, 2032, and 2033, resulting in $N = M/4$ indexes, when M is the number of all descriptors. In FIG. 4, the average value of every fourth descriptor is selected as an index 2041, 2042, resulting in a

total of four indexes. As these two examples illustrate, the number of descriptors in a group is configurable, and the number should be selected to be able to separate images as evenly distributed as possible, and the number of index levels is reasonable. Those of skill in the art will appreciate that the present invention is not restricted to these exemplary descriptor indexing methods.

[0040] Some current image matching algorithms use a floating point number for the descriptor; in the present invention, this is preferably normalized into a range of [0, 1], then multiplied by 65535, resulting in the descriptors being on the range of [0, 65535]. As will be readily appreciated by those of skill in the art, the above range is provided merely by example, and other ranges, smaller or larger, can be used without departing from the present invention. This range is preferable, although not necessary, for hash-based indexing.

[0041] Turning back to the drawing figures, assuming the number of final fine one-to-one matching is selected to be a value F, FIG. 2 illustrates a block diagram of an exemplary indexing matching architecture in accordance with the present invention. Before matching can proceed, the hierarchical indexing hashing tables are built. All images in the source image database are digitized, then the index values are pre-calculated as described above for index groups, for each image. Then, one index group is selected as the level 1 index, and a hash table 2002 is built of, e.g., 65536 (depending on the descriptor value, as discussed above) hash buckets, such that each possible index value of the level 1 index will be indexed to one of the hash buckets. Each hash bucket includes one counter that is the sum of the numbers of the underlying linked images. Then each source image is first hashed into the level 1 index hash table 2002.

[0042] When one image descriptor is hashed into one hash bucket 2006, the associated counter will increase by one. This process is repeated until all the images are inserted into the level 1 index hash table 2002. After the level 1 index hash table 2002 is finished, the process loops through each hash bucket of the level 1 hash table 2002 and checks the counters: any hash bucket 2006 with its counter value larger than value F is subject to further indexing. Then the second level index hash table 2003 can be built within those buckets which have counters larger than the value F in the level 1 hash table, by using the level 2 index. This is repeated, either until the level N indexing hash table 2008 (for a total N index groups) is reached, or no single hash bucket in any indexing level has its counter larger than

the value F. Those index levels with counter values less than the value F include the actual linked list (or array) 2010 of images 2011.

[0043] Further preferably, the index group should be picked such that a hierarchical indexing hash tree can be built such that any hash bucket in any indexing level has its counter less than or equal to the value F, so that the number of final fine one-to-one matching is limited to about value F. As discussed above, this can result in finishing the matching process within a desirable time, such as 500 millisecond.

[0044] Still referring to FIG. 2, for the given target image, the index values are calculated as done for the source images. The level 1 index of the target image is used as an Initial Index 2001 to index 2004 ("center index", in this level) into one hash bucket 2006 of the level 1 index hash table 2002. If the counter is smaller than the value F, then the neighboring two hash buckets 2005, 2007 are selected. If the sum of the counters from 2004 and its neighbors 2005, 2007 is still smaller than the value F, further neighbors are picked until the sum of all picked hash bucket counters are about the value F (the sum may be slightly larger or smaller than the value F).

[0045] Alternatively, when the counter at the center index 2004 is larger than the value F, then no additional neighbors of 2004 are picked, and the process proceeds to level 2 indexing 2003 for further refinement using the level 2 index from the target image to index into the level 2 index hash table 2003. Steps similar to those described above with respect to the level 1 indexing hash table 2002 are performed until an indexing level is found such that the center index and its neighbors has a counter sum approximately that of the value F. Then all image descriptors 2011 in those picked hash buckets 2009 are subject to a final, fine one-to-one matching search. If the second index level still does not have a center index with a counter smaller than the value F, then the process is continued for additional levels until a level is reached in which the hit center index has its counter smaller than the value F.

[0046] The foregoing example shows that the indexing value is calculated by averaging values of descriptors in a group. The present invention is not limited to this specific methodology. Without departing from the scope of the present invention, numerous other methodologies can be utilized, for example an average of square values of descriptors in an index group, neural network indexing or any other indexing values based on characteristics of matching algorithms.

[0047] While the invention has been described in detail with reference to preferred embodiments thereof, it will be apparent to one skilled in the art that various changes can be made, and equivalents employed, without departing from the scope of the invention. Each of the aforementioned documents is incorporated by reference herein in its entirety.

WHAT IS CLAIMED IS:

1. A method for indexing a plurality of images in a database, each image in said database having within it an object of the same type, comprising the steps of:

digitizing each of said plurality of images in said database into an array of descriptors;

separating said descriptors into a plurality of index groups;

calculating for at least one of said plurality of images in said database an index value corresponding to one of said index groups.

2. A method for indexing a plurality of images in a database according to claim 1 further comprising the step of identifying a group of images in said database having an index value corresponding to a particular index group greater than a threshold value.

3. A method for indexing a plurality of images in a database according to claim 1 wherein said index value is a function of at least two of said descriptors.

4. A method for indexing a plurality of images in a database according to claim 3 wherein said function comprises averaging at least two of said descriptors.

5. A method for indexing a plurality of images in a database according to claim 3 wherein said function comprises averaging four of said descriptors.

6. A method in accordance with claim 1, wherein said object type comprises a human face.

7. A method of comparing a target image to a database of a plurality of digitized images comprising the steps of:

digitizing a target object in said target image into an array of descriptors;

calculating an average of at least a first descriptor and a second descriptor from said target object; and

comparing said calculated average to a threshold value.

8. A method for image indexing comprising:

digitizing a target image having an object within it into an array of descriptors;

digitizing at least on source image in a database into an array of descriptors, each said source imaging having an object within it and said object in said first image and said object in said at least one source image being of same type;

wherein said array of descriptors from said first image and said array of descriptors from said at least one source image comprise the same type and the same number of descriptors; and

separating said descriptors into index groups, wherein said index groups each have an index value.

9. A method for image indexing comprising:

providing a first image, said first image having an object;

providing an image database having at least one source image, said at least one source image in said image database having an object;

wherein said object in said first image and said object in said at least one source image in said image database being of same type;

digitizing said first image into an array of descriptors;

digitizing said at least one source image into an array of descriptors;

wherein said array of descriptors from said first image and said array of descriptors from said at least one source image comprise the same type and the same number of descriptors, and wherein a descriptor corresponds to one feature in said object; and

separating said descriptors into index groups, wherein said index groups each have an index value.

10. A method in accordance with claim 9, wherein said object type comprises a human face.

11. A method according to claim 9, wherein said index value is the average value of said descriptors in said index group.

12. A method according to claim 9, wherein said index group comprises a consecutive number Y of said descriptors.

13. A method according to claim 12, further comprising selecting Y.

14. A method according to claim 9, wherein said index group comprises every other Z descriptors, wherein Z is an integer.

15. A method according to claim 14, further comprising selecting Z.

16. A method of image indexing comprising:

providing a first index level, a second index level, and a pre-set number;

providing a image database having a source image;

wherein said first index level has an index value, said second index level has an index value, said first index level having a list of buckets, said second index level having a list of buckets, and said buckets each having a counter;

if said counter of said bucket in said first index level is larger than said pre-set number, then said bucket in said first index level has a second index level, and said counter is the sum of counters of said buckets in said second index level;

if said counter of said bucket in said first index level is less than or equal to said pre-set number, then said bucket in said first index level has a list of image templates, and the number of said image templates is said counter;

wherein said image template comprising said digitized descriptors of said source image according to claim 1.

17. A method according to claim 16, wherein said first index level comprises a hash table, and comprising using said index value as a hash value.

18. A method according to claim 16, wherein said second index level comprises a hash table, and comprising using said index value as a hash value.

19. A method of finding top matches from an image database having source images, given a target image, and using image indexing architecture with a one-by-one matching algorithm, a pre-set number as an upper limit of a final fine group, and a pre-set percentage threshold for similarity matching for said matching algorithm, said finding process comprising:

providing an image indexing architecture according to claim 8;

digitizing said target image into an image template of an array of descriptors;

calculating index groups of said target image;

use said first index of said target image to index into said first index level of said image indexing architecture;

if the counter of said first index of said first index level is larger than or equal to the said pre-set number, then using said second index of said target image to index into said second index level of said image indexing architecture;

if the counter of said first index of said first index level is less than the said pre-set number, then picking neighboring buckets of said first index in said first index level until the sum of the counters of said neighbor buckets of said first index is larger than or equal to the said pre-set number;

picking the image templates from said neighbor buckets of said first index as final one-by-one matching group;

thereafter applying said one-by-one matching algorithm to said target image and each of said source images of said picked final one-by-one matching group.

20. A method according to claim 19, wherein said matching algorithm provides a list of matches with higher than said pre-set percentage threshold, or gives no match if there are no matches with higher than said pre-set percentage threshold.