



US 20070067570A1

(19) **United States**

(12) **Patent Application Publication**
Su et al.

(10) **Pub. No.: US 2007/0067570 A1**

(43) **Pub. Date: Mar. 22, 2007**

(54) **DATA PROCESSING METHODS AND SYSTEMS**

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/230,402, filed on Sep. 20, 2005.

(75) Inventors: **Chan-Hung Su**, Kaohsiung City (TW);
Yu-Cheng Hsieh, Hsinchu City (TW);
Jf Yeh, Miaoli Hsien (TW)

Publication Classification

(51) **Int. Cl.**
G06F 12/00 (2006.01)
(52) **U.S. Cl.** 711/118

Correspondence Address:

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, LLP
100 GALLERIA PARKWAY, NW
STE 1750
ATLANTA, GA 30339-5948 (US)

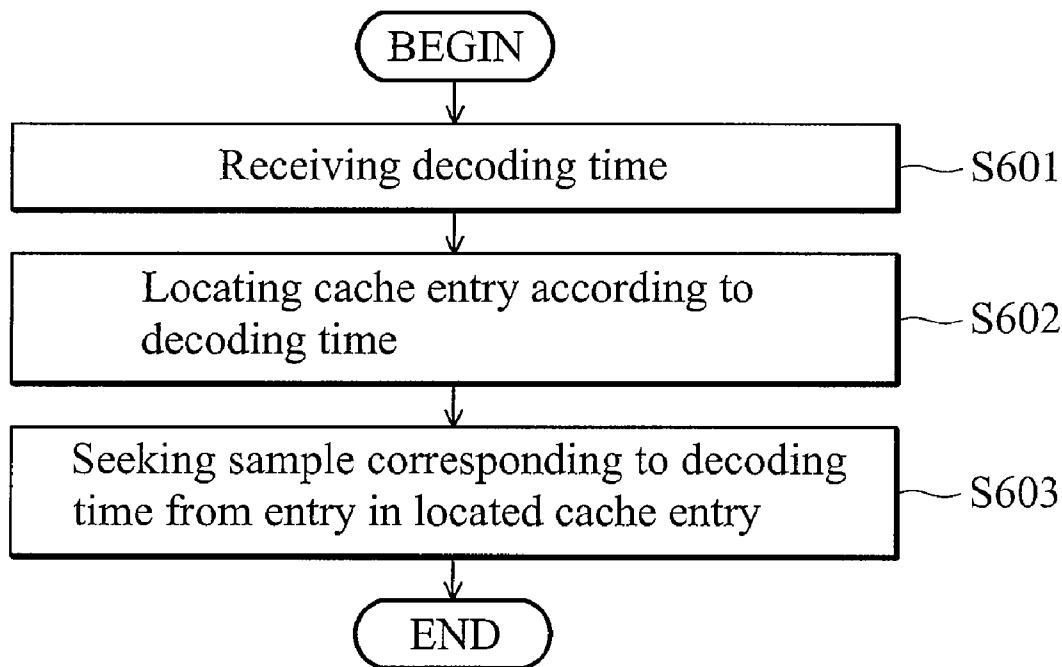
(57) **ABSTRACT**

Data processing methods and systems for accessing a target sample in a media data stream. The media data stream has a plurality of samples and corresponding sample information recorded in at least one entry, wherein the sample information for a predetermined number of the entries is calculated and the calculated result is stored in at least one cache entry. Target sample information corresponding to the target sample is provided. One of the cache entries is located by comparing the target sample information with the calculated result of respective cache entries. After locating the cache entry, the target sample from the entries is located corresponding to the located cache entry.

(73) Assignee: **MEDIATEK INC.**, Hsin-Chu (TW)

(21) Appl. No.: **11/465,469**

(22) Filed: **Aug. 18, 2006**



BOX TYPE	'stts'
Entry Count	5
Sample Count	SAMPLE-DELTA
6	66
14	67
11	63
2	64
12	66

100

FIG. 1 (RELATED ART)

BOX TYPE	'stss'
Entry Count	6
SAMPLE NUMBER	
1	
8	
16	
24	
31	
38	

200

FIG. 2 (RELATED ART)

BOX TYPE	'stsc'	
Entry Count	4	
FIRST CHUNK	SAMPLE PER CHUNK	SAMPLE DESCRIPTION INDEX
1	3	1
3	4	2
6	7	3
9	6	4

300

FIG. 3 (RELATED ART)

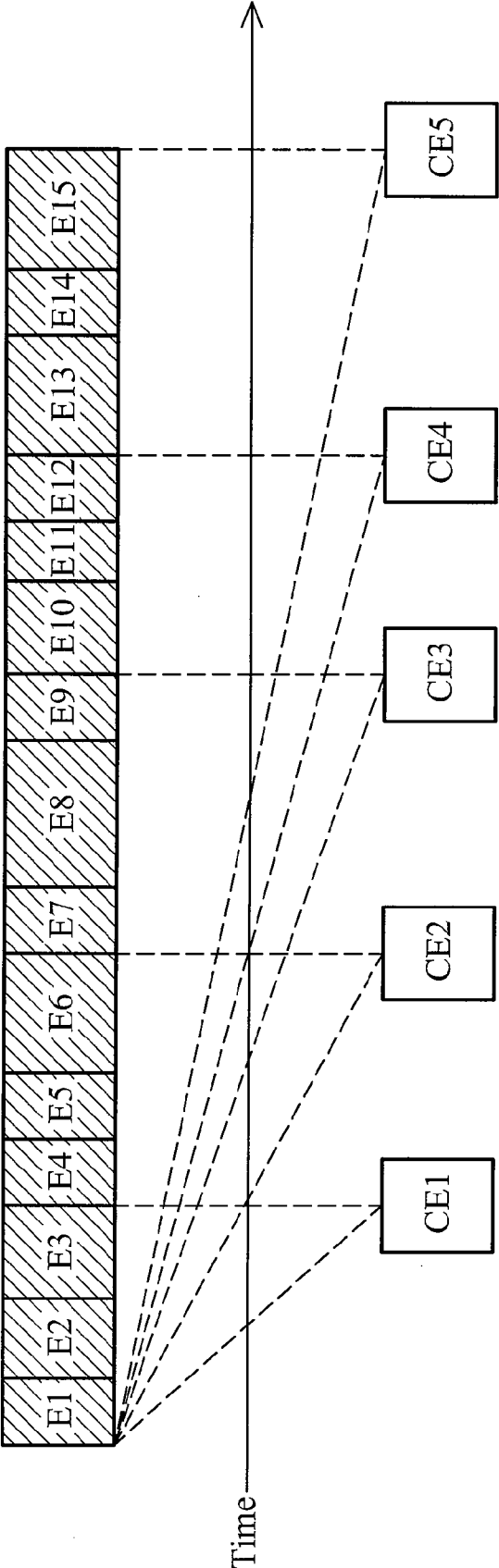


FIG. 4

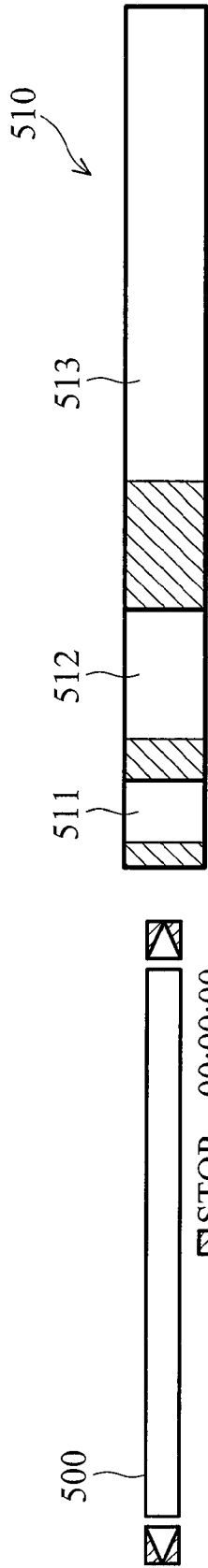


FIG. 5A

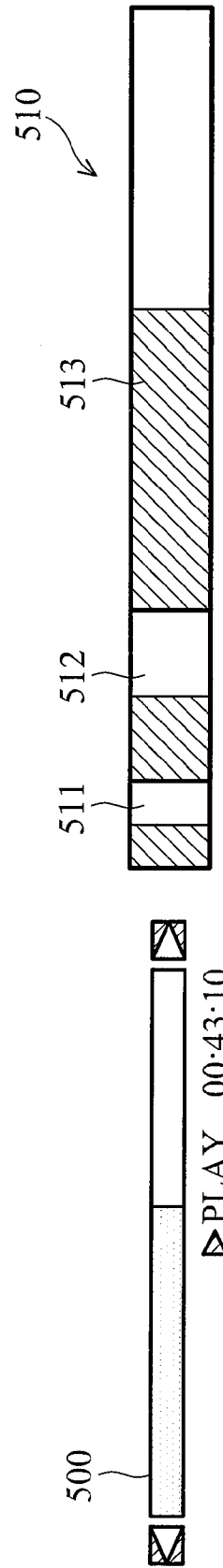


FIG. 5B

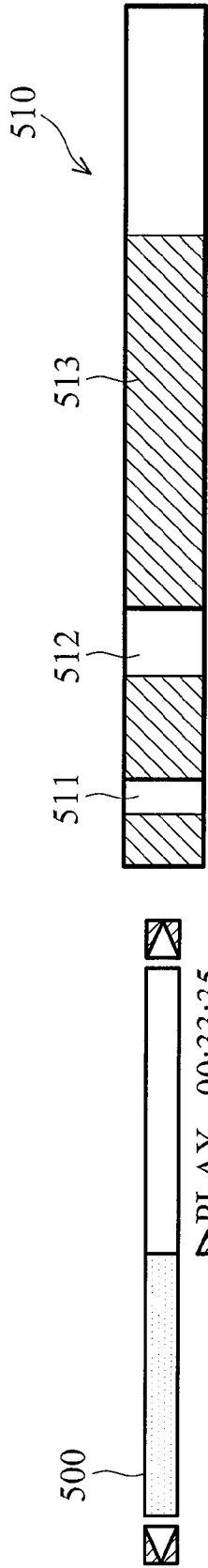


FIG. 5C

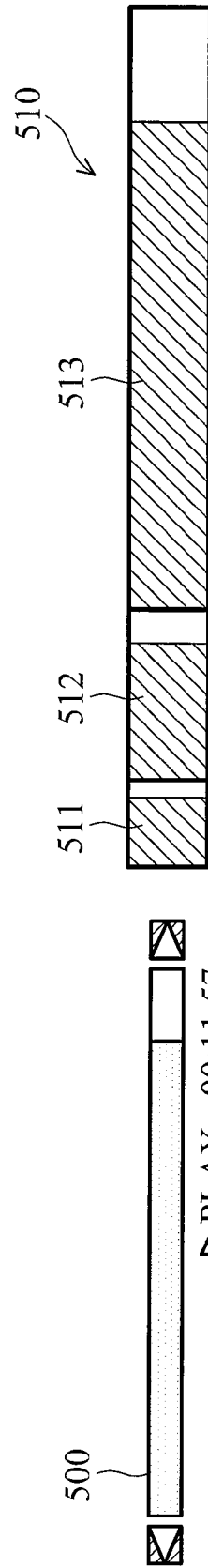


FIG. 5D

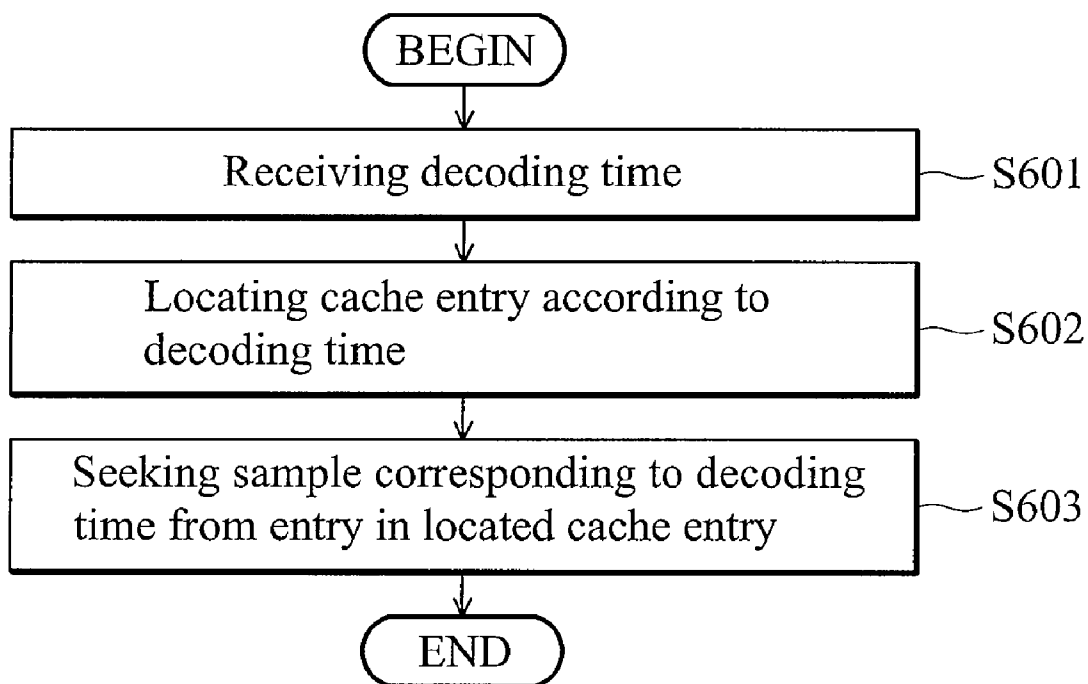


FIG. 6

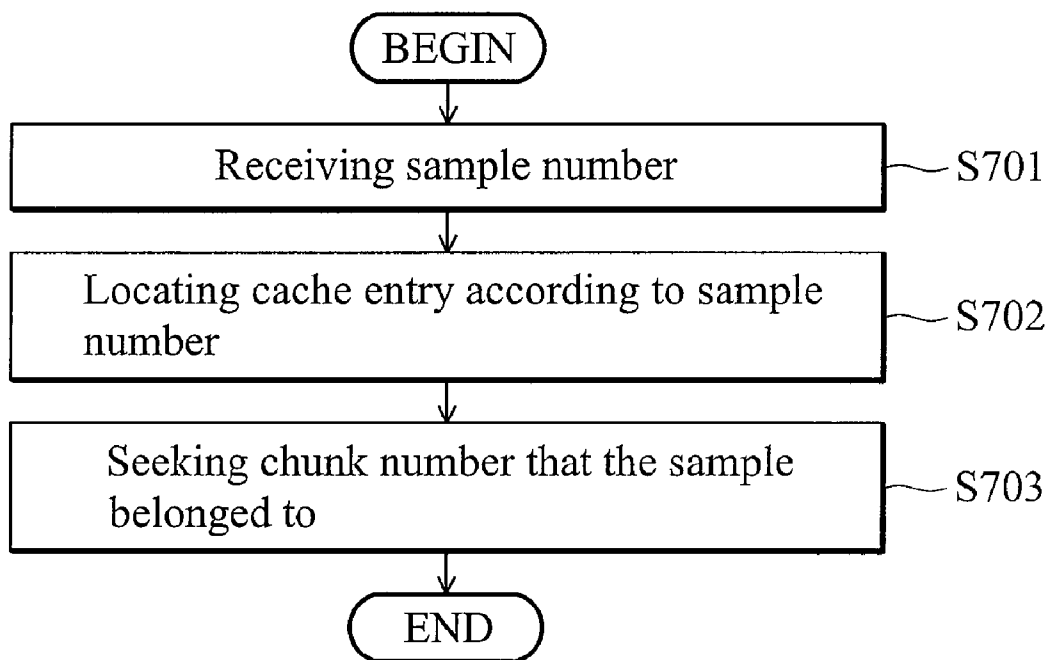


FIG. 7

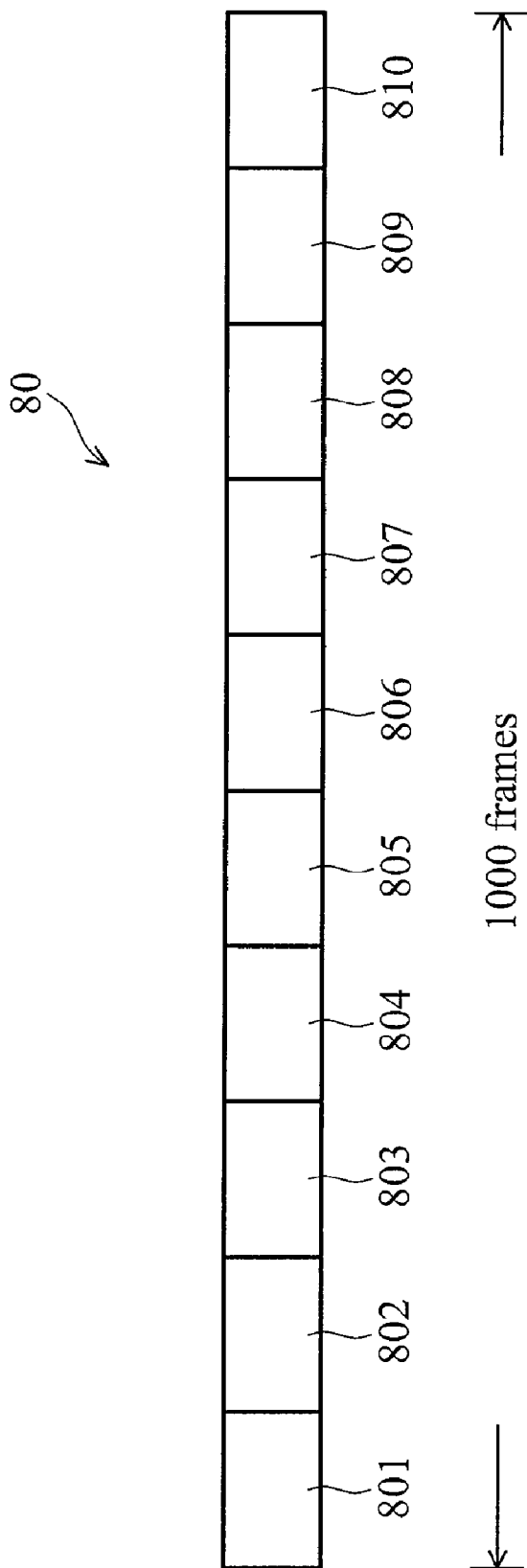


FIG. 8

900
↙

	Index (accumulated frames)	Offset (Bytes)
901	100	32796
902	200	63484
903	300	99876
904	400	123450
905	500	151000
906	600	185762
907	700	211135
908	800	248998
909	900	288553
910	1000	319788

FIG. 9

DATA PROCESSING METHODS AND SYSTEMS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This is a continuation-in-part application of copending application Ser. No. 11/230,402 filed Sep. 20, 2005, entitled Data Processing Methods and Systems.

BACKGROUND

[0002] The present disclosure relates generally to data processing, and more particularly to methods and systems for processing and accessing metadata in a media file.

[0003] Media files, such as MPEG-4 files, comprise media data and metadata of the media data. The metadata provides data sample information to media applications for processing media data in the media file. Generally, media files are defined to be organized with several structural elements. For example, MPEG-4 files are composed of structural elements called boxes. Each box may comprise media data, metadata or other sub-boxes. For example, a sample table box (STBL) records time and file information of media data. According to the time and file information in the STBL box, applications can obtain the time, type, data size, and position in the media file and further perform the playback or random-seek functions toward the media file, accordingly. The STBL box also includes several sub-boxes comprising a decoding time to sample box (STTS) as shown in FIG. 1, a sample size box (STSZ), a sample to chunk box (STSC) as shown in FIG. 3, a chunk offset box (STCO), a sync sample box (STSS) as shown in FIG. 2, sample description table (STSD), and others.

[0004] The STTS box contains at least one entry for recording the time duration of samples of media data. FIG. 1 shows a STTS box 100, comprising 5 entries with total 45 samples in the media data. It should be noted that data in the STTS box is recorded using the concept of run-length coding to reduce the size thereof. That is, the time duration for media sample #1~#6, #7~#20, #21~#31, #32~#33, and #34~#45 is 66, 67, 63, 64, and 66 time units, respectively, and the total time duration of the 45 sample of the media data is 2947 time units.

[0005] Traditionally, a linear search through these boxes is performed to help locating a specific sample with a target decoding time. A linear search means that the time duration of respective samples from the very first media sample is accumulated until the accumulated time duration equals or exceeds the target decoding time. For example, to locate a specific sample with a specific decoding time 2000, a linear search is performed by accumulating the time duration of the first thirty samples. Since the total time duration of the first thirty samples is 1964 ($6*66+14*67+10*63=1964$) and the total time duration of the first thirty-one samples is 2027 ($6*66+14*67+11*63=2027$) which exceeds the specific decoding time 2000, sample #30 is then located. The linear search calculation is time-consuming if there are large number of entries in these boxes.

[0006] In addition, the media samples in the media data are grouped into chunks. The STSC box records the mapping relationship between samples and chunks. Based on the mapping relationship recorded in the STSC box, one can identify in which chunk a target sample resides, and further

obtain other related data using the chunk information. FIG. 3 shows a STSC 300, in which the number of samples in chunk #1~#2, chunk #3~#5, chunk #6~#8, and chunk #9 and the latter is 3, 4, 7, and 6, respectively. Similarly, data in the STSC is recorded using the concept of run-length coding to reduce the size thereof. To locate a specific sample, a specific chunk corresponding to the specific sample is sought by performing a linear search by accumulating the number of samples from the first chunk. The calculation is time-consuming if the number of entries in the STSC box is large.

SUMMARY

[0007] Data processing methods and systems are provided.

[0008] In an embodiment of a data processing method for accessing a target sample in a media data stream, target sample information corresponding to the target sample is provided. The media data stream has a plurality of samples and corresponding sample information recorded in at least one entry, wherein the sample information for a predetermined number of the entries is calculated and the calculated result is stored in at least one cache entry. One of the cache entries is located by comparing the target sample information with the calculated result of respective cache entries. After locating the cache entries, the target sample from the entries is located according to the located cache entry.

[0009] An embodiment of a data processing system comprises a media data stream, at least one cache entry, and a processing unit. The media data stream comprises a plurality of samples and corresponding sample information recorded in at least one entry. The sample information for a predetermined number of the entries is calculated and the calculated result is stored in the cache entry. The processing unit receives target sample information corresponding to a target sample, locates one of the cache entries by comparing the target sample information with the calculated result of respective cache entries, and after locating the cache entries, locates the target sample from the entries according to the located cache entry.

[0010] Data processing methods may take the form of program code embodied in a tangible media. When the program code is loaded into and executed by a machine, the machine becomes an apparatus for practicing the disclosed method.

DESCRIPTION OF THE DRAWINGS

[0011] The invention will become more fully understood by referring to the following detailed description with reference to the accompanying drawings, wherein:

[0012] FIG. 1 shows an example of a STTS;

[0013] FIG. 2 shows an example of a STSS;

[0014] FIG. 3 shows an example of a STSC;

[0015] FIG. 4 is a schematic diagram illustrating an embodiment of entries in the STTS box and its corresponding pre-calculated cache entries;

[0016] FIGS. 5A~5D shows an example of incremental processing;

[0017] FIG. 6 is a flowchart showing an embodiment of a data processing method; and

[0018] FIG. 7 is a flowchart showing another embodiment of a data processing method.

[0019] FIG. 8 is a schematic diagram of an audio file.

[0020] FIG. 9 is an embodiment of a pre-calculated cache table.

DESCRIPTION

[0021] Data processing methods and systems are provided, in which pre-calculated cache entry, shared cache pool and incremental processing are employed. In some embodiments, the data processing methods and systems are used to decode MPEG-4 files, which is well-defined in ISO standard and is however not limited in the present invention. An overview of the MPEG-4 files is provided in a standard specification of ISO 14496-12, which is incorporated herein by reference in its entirety for all purposes. In addition, in some embodiments, the data processing methods and systems are used to decode MP3 data. The data processing methods and systems can be implemented to decode other types of audio data without metadata, such as AAC, AMR/AWB).

Pre-Calculated Cache Entry:

[0022] First, the data processing methods and systems used to decode MPEG-4 files are described.

[0023] FIG. 4 is a schematic diagram illustrating an embodiment of entries in the STTS box and its corresponding pre-calculated cache entries. Each cache entry records the accumulated total time duration of a group of STTS entries. The former group of STTS entries is a sub-set of the latter groups. As shown in FIG. 4, the accumulated time duration and sample count of the STTS entries E1 to E3 are recorded by cache entry CE1, the accumulated time duration and sample count of the STTS entries E1 to E6 are recorded by cache entry CE2, the accumulated time duration and sample count of the STTS entries E1 to E9 are recorded by cache entry CE3, the accumulated time duration and sample count of the STTS entries E1 to E12 are recorded by cache entry CE4, and the accumulated time duration and sample count of the STTS entries E1 to E15 are recorded by cache entry CE5. For example, if STTS entry E4 records 3 samples and the time duration of each is 5, STTS entry E5 records 5 samples and the time duration of each is 10, and STTS entry E6 records 7 samples and the time duration of each is 8, the time duration recorded in cache entry CE2 is the time duration recorded in CE1 plus the total time duration of 15 samples (121), and the sample count recorded in cache entry CE2 is the sample count recorded in CE1 plus 15. The calculation for other cache entries is similar thereto. It should be noted that the calculation of the cache entries could be calculated in advance.

[0024] To locate the sample data with the specific decoding time T, the cache entry CE_N is checked, such that the duration recorded in CE_N is greater than T and the duration recorded in CE_{N-1} is smaller than T. In some embodiments, the cache entry can be located by a binary search. If CE_{N-1} records the time duration from E₁ to E_x, and CE_N records the time duration from E₁ to E_y, it is trivial that the sample data which we are looking for must resides in the STTS entries

E_x to E_y. Then a linear search is applied in the STTS entries E_x to E_y to locate the wanted sample data.

[0025] To obtain the decoding time of a specific sample data S_K, the similar method applies. The cache entry CE_N is checked, such that the accumulated sample count recorded in CE_N is greater than K and the accumulated sample count recorded in CE_{N-1} is smaller than K. In some embodiments, the cache entry can be located by a binary search. If CE_{N-1} records the sample count from E₁ to E_x, and CE_N records the sample count from E₁ to E_y, it is trivial that the sample data which we are looking for must resides in the STTS entries E_x to E_y. Then a linear search is applied in the STTS entries E_x to E_y to locate the wanted sample data and its decoding time.

[0026] With the use of the pre-calculated cache, the frequencies of accessing media files can be greatly reduced since the pre-calculated data can be stored in memory. It is also understood that the use of pre-calculated cache entry can be also applied to STSS and STSC. In STSC, the cache entry records the total number of samples in the chunks covered by the current and prior cache entries.

[0027] In addition, the data processing methods and systems used to decode MP3 files are described.

[0028] FIG. 8 is a schematic diagram of an audio file. In the following embodiment, the audio file contains several frames, i.e. X frames. In addition, the audio file is evenly divided into Y segments. The end point of each of the Y segments is referred to as a recording point. The value X can vary for different audio files. The value Y is preset in accordance to a preserved storage space for a pre-calculated cache table. The number of segments and the size of the pre-calculated cache table are pre-determined and fixed where the size of each of the Y segments varies according to the value X of the audio file. For example, it is assumed that Y is 100 for each audio file to be decoded. In other words, the pre-calculated cache table would comprise 100 entries. In this case, for an audio file comprising 10000 frames, each segment would comprise 100 frames. For an audio file comprising 5000 frames, each segment would comprise 50 frames. As shown in FIG. 8, audio file 80 comprises 1000 frames. Here, Y is set to be 10. Audio file 80 is divided into segments 801~810, each of them comprises 100 frames. Accordingly, segment 801 comprises frames 1~100, and segment 802 comprises frames 101~200. Frames comprised in any of segments 801~810 can be determined on the analogy of this.

[0029] FIG. 9 is an embodiment of a pre-calculated cache table. Each cache entry records an offset value corresponding to an end point of a particular segment. The number of entries comprised in the cache table is the same with the preset value Y. For example, when Y equals to 10, cache table 900 would comprise 10 entries, wherein entry 901 corresponds to segment 801, entry 902 corresponds to segment 802. Each entry of the cache table 90 corresponds to a segment of audio file 800. In an entry, the left column specifies accumulated number of frames at the end point of a segment corresponding to the entry, and the right column specifies an offset of the end point specified at the left column. It should be noted that the offset values recorded in the cache entries could be calculated in advance.

[0030] In MP3 file format, the frame duration is fixed; that is, given a decoding time T, we can easily find the corre-

sponding frame F. To locate the frame F with the specific decoding time T, one of the cache entries is first located. For example, as shown in FIG. 9, the present invention first determines in which cache entry the audio frame resides by comparing F with the accumulated frame count. If the wanted frame F is the 205th frame, it is clearly that F resides in the file with offset range from 63484 to 99876. That is, after the cache entry is located, the segment 902 corresponding to the located cache entry is further searched to locate the offset corresponding to the specific decoding time.

Shared Cache Pool:

[0031] In some embodiments, the number of corresponding cache entries to each box might be either fixed or dynamically decided according to each box size.

[0032] In some embodiments, the total size of cache memory might be limited and the allocation of the cache entries for each box can be determined using a shared cache pool model. In the shared cache pool, the number of cache entries for a box is determined according to the number of entries in the boxes requiring cache entries in the order of initialization. For example, to allocate a certain amount of cache entries to the STTS box, the STSS box and the STSC box, the system will firstly allocate a first ratio of the cache entries to the STSS box. The first ratio could be the number of STSS entries divided by the total entry count of the 3 tables. In some embodiments, the number of cache entries allocated to the STSS box can be further limited to a pre-defined maximum number. Then, the system will allocate a second ratio of the cache entries to the STSC box. The second ratio could be the number of STSC entries divided by the total entry count of the 3 tables. In some embodiments, the number of cache entries allocated to the STSC box can also be further limited to a pre-defined maximum number. Finally, the system will allocate the rest of the cache entries to the STTS box. It should be noted that the order of cache allocation is not limited in the present invention. Any order of cache allocation should be covered in the present invention.

Incremental Processing:

[0033] First, the data processing methods and systems used to decode MPEG-4 files are described.

[0034] As mentioned above, with the pre-calculated cache entries, lots of time spent on redundant computation can be reduced. However, if the boxes (STTS, STSS and STSC) are very large, the calculation of cache entries will need a certain amount of time. For example, if complete cache entries for the STTS box must be prepared before a playback, the accumulated total time duration of a group of STTS entries for respective cache entries must be calculated. To reduce the response time caused by calculation of cache entries before playback, the calculation of cache entries is performed with the concept of incremental processing. In incremental processing, a part of the cache entries is calculated before the playback of media data, and the rest of the cache entries are calculated progressively during playback. If the data sought by users is not found in the currently ready cache entries, the calculation of cache entries is performed forward until the data is covered by the cache entries.

[0035] FIGS. 5A~5D shows an example of incremental processing. In this example, the memory space 511 of the memory 510 is allocated to the STSS box for its correspond-

ing cache entries, the memory space 512 of the memory 510 is allocated to the STSC box for its corresponding cache entries, and the memory space 513 of the memory 510 is allocated to the STTS box for its corresponding cache entries. As shown in FIG. 5A, before the playback of media data (see playback status 500), only a part of the cache entries are calculated. The rest of the cache entries for respective boxes are calculated progressively during playback, as shown in FIG. 5B. Since the calculation is accomplished progressively during playback, each time only a small part of the cache entries are calculated (as shown in FIG. 5C), no time delay in playback will be aware by users. If a target decoding time prior to current decoding time is sought, since the required data can be found in the calculated cache entries, no time delay will be aware by users. If a target decoding time is sought and the required data is not found in the calculated cache entries, the calculation of cache entries is performed forward until the data is covered by the cache entries, as shown in FIG. 5D.

[0036] FIG. 6 is a flowchart showing an embodiment of a data processing method. In this embodiment, a STTS box is used as an example, rather a limitation to the invention. To locate a sample date with the specific decoding time received in step S601, in step S602, one of the cache entries is firstly located. In step S603, after the cache entry is located, a sample corresponding to the decoding time can be easily found by a linear search. Here, after locating the cache entry, the method of the present invention only needs to search through the group corresponding to the cache entry for the target sample by accumulating the time durations of respective samples in the STTS entries until the sum of the accumulated time duration of samples equals or exceeds the target decoding time. Therefore, total search time can be reduced. It is understood that if the sample is sought for playback, it is also determined whether the sample can be randomly accessed. As described, there are random accessible samples and non-random accessible samples among the media data. A random accessible sample means the sample can be sought and decoded without regard to other samples. If the desired sample is not a random accessible one, a random access point (random accessible sample) closest to (equal to or less than) the sample is sought from the STSS, and a decoding process begins from the random accessible sample, until the desired sample is encountered.

[0037] FIG. 7 is a flowchart showing another embodiment of a data processing method. In this embodiment, a STSC box is used as an example, rather a limitation to the invention. To locate a sample with a sample number received in step S701, in step S702, one of the cache entries is firstly located. In step S703, after the cache entry is located, a chunk number that the sample belonged to is located by a linear search. Here, after locating the cache entry, the method of the present invention only needs to search through the group corresponding to the cache entry for the target chunk number. It is understood that if the sample is sought for playback, related information of the sample is further retrieved from a sample description table (STSD) according to a corresponding sample description index of the located chunk number.

[0038] Data processing methods, or certain aspects or portions thereof, may take the form of program code (i.e., executable instructions) embodied in tangible media, such as products, floppy diskettes, CD-ROMS, hard drives, or any

other machine-readable storage medium, wherein, when the program code is loaded into and executed by a machine, such as a computer, the machine thereby becomes an apparatus for practicing the methods. The methods may also be embodied in the form of program code transmitted over some transmission medium, such as electrical wiring or cabling, through fiber optics, or via any other form of transmission, wherein, when the program code is received and loaded into and executed by a machine, such as a computer, the machine becomes an apparatus for practicing the disclosed methods. When implemented on a general-purpose processor, the program code combines with the processor to provide a unique apparatus that operates analogously to application specific logic circuits.

[0039] In addition, the data processing methods and systems used to decode MP3 files are described.

[0040] As mentioned above, with the pre-calculated cache entries, lots of time spent on computation to locate a specific decoding time can be reduced. However, if the audio file is very large, the calculation of cache entries requires a certain amount of time. To reduce the response time caused by calculation of cache entries before playback, the calculation of cache entries is performed with the concept of incremental processing.

[0041] In incremental processing, a certain portion of the cache entries are calculated before the playback, and the rest of the cache entries are calculated progressively during playback. If the data sought by users is not found in the currently ready cache entries, the calculation of cache entries is performed forward until the data is covered by the cache entries.

[0042] For example, when an audio file is to be decoded, the first few entries of the cache table are established before the playback. During the playback, other entries are established. The establishment of cache table can be in the lead from the playback operation. If a user wants to rush the playback operation forward, the established cache table can be used to facilitate locating of the specific forward target.

[0043] The established cache table can be overwritten when another audio file is to be decoded. In addition, the established cache table can be copied to another storage position when another audio file is to be decoded. For the later example, the copied cache table can be stored for further use. When the same audio file is played next time, the stored cache table can be utilized, and the establishment of the cache table needs not to be repeated.

[0044] While the invention has been described by way of example and in terms of preferred embodiment, it is to be understood that the invention is not limited thereto. Those skilled in this technology can still make various alterations and modifications without departing from the scope and spirit of this invention. Therefore, the scope of the present invention shall be defined and protected by the following claims and their equivalents.

What is claimed is:

1. A data processing method for accessing a target sample in a media data stream, the media data stream having a plurality of samples and corresponding sample information recorded in at least one entry, the method comprising:

providing target sample information corresponding to the target sample, wherein the sample information for a predetermined number of entries is calculated and the calculated result is stored in at least one cache entry;

locating one of the cache entries by comparing the target sample information with the calculated result of respective cache entries; and

after locating the cache entry, locating the target sample from the entries corresponding to the located cache entry.

2. The method of claim 1 further comprising determining the number of the cache entries according to the number of the entries.

3. The method of claim 1 further comprising calculating the sample information for the respective predetermined number of the entries during playback.

4. The method of claim 1 wherein at least one table is provided, wherein the table is a time to sample box (STTS) listing the entries and the corresponding sample information, the sample information comprises a decoding time of samples in the corresponding entry.

5. The method of claim 4 wherein the target sample information comprises a specific decoding time.

6. The method of claim 5 wherein the specific decoding time is between the accumulated time duration recorded in the cache entries prior to the located cache entry, and the accumulated time duration recorded in the located cache entry.

7. The method of claim 1 wherein at least one table is provided, wherein the table comprises a sample to chunk box (STSC), comprising chunks of the media data stream, and the sample information comprises the number of samples in respective chunks.

8. The method of claim 7 wherein the target sample number is between the accumulated number of samples recorded in the cache entries prior to the located cache entry and the accumulated number of samples recorded in the located cache entry.

9. The method of claim 1 wherein the media data stream is of a MPEG-4 file.

10. The method of claim 1, wherein the media data stream is comprised in an audio file with no metadata.

11. The method of claim 10, wherein the media data stream is comprised in an MP3, AAC, or AMR/AWB file.

12. The method of claim 10, further dividing the media data stream into segments, and determining the number of the cache entries according to the number of the segments.

13. The method of claim 10, wherein the sample information comprises offset information of an end point of the corresponding segment.

14. The method of claim 13, further comprising calculating the offset for the entries during playback.

15. The method of claim 10, wherein the target sample information comprises a specific decoding time.

16. The method of claim 10, wherein the target sample is located by locating one of the cache entries by a binary search, and further locating the target sample within the located segment by a linear search.

17. The method of claim 10, further copying the established cache entries.

18. The method of claim 10, further rewriting the cache entries when another audio file is to be processed.

19. A machine-readable storage medium comprising a computer program, which, when executed, causes a device to perform a data processing method for accessing a target sample in a media data stream, the media data stream having a plurality of samples and corresponding sample information recorded in at least one entry, the method comprising:

providing target sample information corresponding to the target sample, wherein the sample information for a predetermined number of entries is calculated and the calculated result is stored in at least one cache entry;

locating one of the cache entries by comparing the target sample information with the calculated result of respective cache entries; and

after locating the cache entry, locating the target sample from the entries corresponding to the located cache entry.

20. The storage medium of claim 19 wherein the method further comprising determining the number of the cache entries according to the number of the entries.

21. The storage medium of claim 19 wherein the method further comprising calculating the sample information for the respective predetermined number of the entries during playback.

22. The storage medium of claim 19 wherein at least one table is provided, wherein the table is a time to sample box (STTS) listing the entries and the corresponding sample information, the sample information comprises a decoding time of samples in the corresponding entry.

23. The storage medium of claim 22 wherein the target sample information comprises a specific decoding time.

24. The storage medium of claim 23 wherein the specific decoding time is between the accumulated time duration recorded in the cache entries prior to the located cache entry, and the accumulated time duration recorded in the located cache entry.

25. The storage medium of claim 19 wherein at least one table is provided, wherein the table comprises a sample to chunk box (STSC), comprising chunks of the media data stream, and the sample information comprises the number of samples in respective chunks.

26. The storage medium of claim 25 wherein the target sample number is between the accumulated number of samples recorded in the cache entries prior to the located cache entry and the accumulated number of samples recorded in the located cache entry.

27. The storage medium of claim 19 wherein the media data stream is of a MPEG-4 file.

28. The storage medium of claim 19, wherein the media data stream is comprised in an audio file with no metadata.

29. The storage medium of claim 28, wherein the media data stream is comprised in an MP3, AAC, or AMR/AWB file.

30. The storage medium of claim 28, wherein the method further divides the media data stream into segments, and determines the number of the cache entries according to the number of the segments.

31. The storage medium of claim 28, wherein the sample information comprises offset information of an end point of the corresponding segment.

32. The storage medium of claim 31, wherein the method further comprises calculating the offset for the entries during playback.

33. The storage medium of claim 28, wherein the target sample information comprises a specific decoding time.

34. The storage medium of claim 28, wherein the method further locates the target sample by locating one of the cache entries by a binary search, and further locates the target sample within the located segment by a linear search.

35. The storage medium of claim 28, wherein the method further copies the established cache entries.

36. The storage medium of claim 28, wherein the method further rewrites the cache entries when another audio file is to be processed.

37. A data processing system, comprising:

a media data stream comprising a plurality of samples and corresponding sample information recorded in at least one entry;

at least one cache entry storing a calculated result of the sample information for a predetermined number of the entries; and

a processing unit receiving target sample information corresponding to a target sample, locating one of the cache entries by comparing the target sample information with the calculated result of respective cache entries, and after locating the cache entry, locating the target sample from the entries corresponding to the located cache entry.

38. The system of claim 37 wherein the processing unit further determines the number of the cache entries according to the number of the entries.

39. The system of claim 37 wherein the processing unit further calculates the sample information for the respective predetermined number of the entries during playback.

40. The system of claim 37 wherein at least one table is provided, wherein the table is a time to sample box (STTS) listing the entries and the corresponding sample information, the sample information comprises a decoding time of samples in the corresponding entry.

41. The system of claim 40 wherein the target sample information comprises a specific decoding time.

42. The system of claim 41 wherein the specific decoding time is between the accumulated time duration recorded in the cache entries prior to the located cache entry, and the accumulated time duration recorded in the located cache entry.

43. The system of claim 37 wherein at least one table is provided, wherein the table comprises a sample to chunk box (STSC), comprising chunks of the media data stream, and the sample information comprises the number of samples in respective chunks.

44. The system of claim 43 wherein the target sample number is between the accumulated number of samples recorded in the cache entries prior to the located cache entry and the accumulated number of samples recorded in the located cache entry.

45. The system of claim 37 wherein the media data stream is of a MPEG-4 file.

46. The system of claim 37 wherein the data processing system is a MPEG-4 file decoder.

47. The system of claim 37, wherein the media data stream is comprised in an audio file with no metadata.

48. The system of claim 47, wherein the media data stream is comprised in an MP3, AAC, or AMR/AWB file.

49. The system of claim 47, wherein the processing unit further divides the media data stream into segments, and determines the number of the cache entries according to the number of the segments.

50. The system of claim 47, wherein the sample information comprises offset information of an end point of the corresponding segment.

51. The system of claim 50, wherein the processing unit further calculates the offset for the entries during playback.

52. The system of claim 47, wherein the target sample information comprises a specific decoding time.

53. The system of claim 47, wherein the processing unit further locates the target sample by locating one of the cache entries by a binary search, and further locates the target sample within the located segment by a linear search.

54. The system of claim 47, wherein the processing unit further copies the established cache entries for further use.

55. The system of claim 47, wherein the processing unit further rewrites the cache entries when another audio file is to be processed.

* * * * *