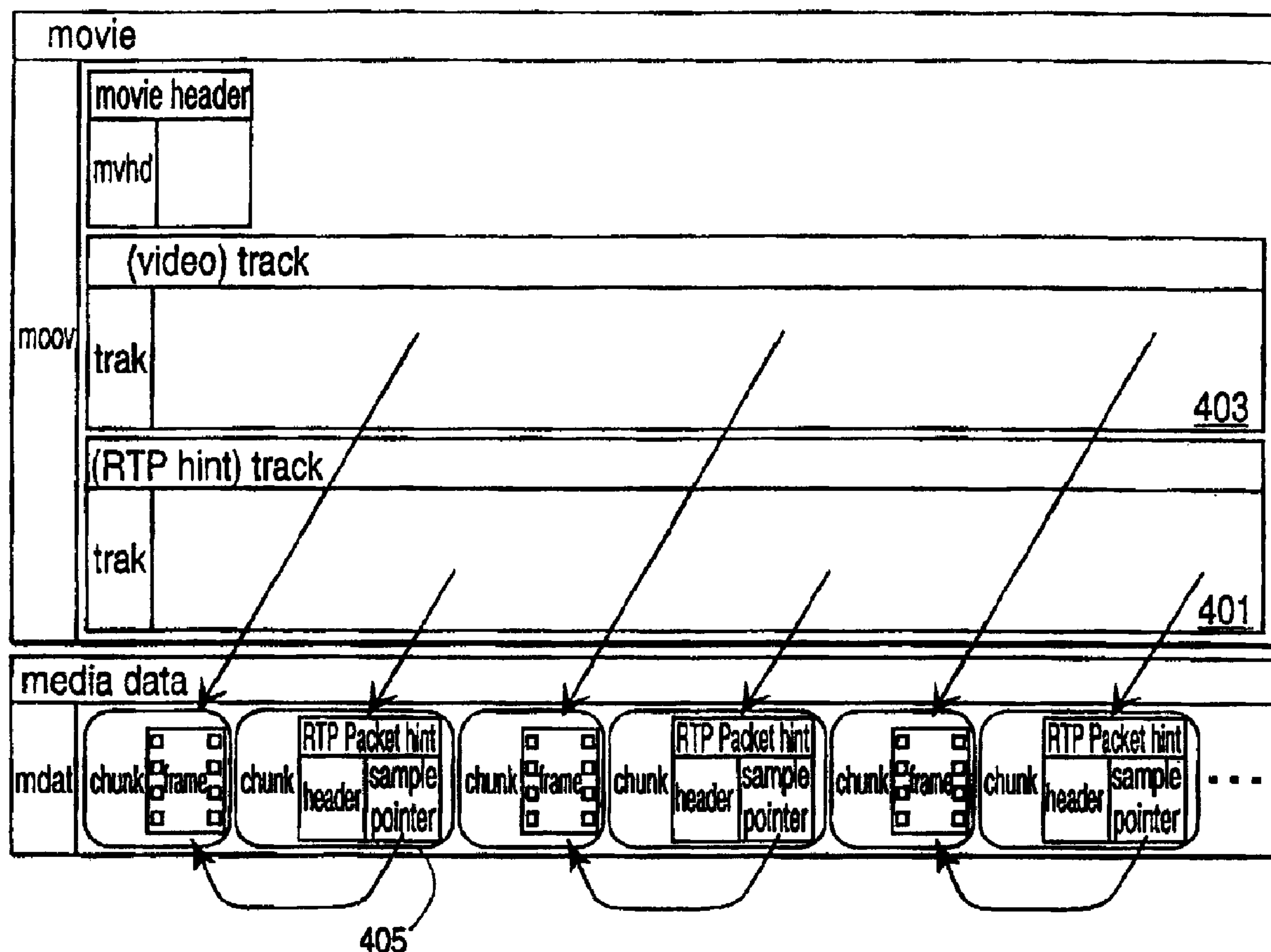




(86) Date de dépôt PCT/PCT Filing Date: 1999/01/13
(87) Date publication PCT/PCT Publication Date: 1999/07/22
(45) Date de délivrance/Issue Date: 2012/03/20
(85) Entrée phase nationale/National Entry: 2000/07/07
(86) N° demande PCT/PCT Application No.: US 1999/000954
(87) N° publication PCT/PCT Publication No.: 1999/037057
(30) Priorités/Priorities: 1998/01/15 (US60/071,566);
1998/08/25 (US09/140,173)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01),
H04L 29/08 (2006.01)
(72) Inventeurs/Inventors:
JONES, ANNE, US;
GEAGAN, JAY, US;
GONG, KEVIN L., US;
PERIYANNAN, ALAGU, US;
SINGER, DAVID W., US
(73) Propriétaire/Owner:
APPLE INC., US
(74) Agent: RICHES, MCKENZIE & HERBERT LLP

(54) Titre : PROCEDE ET EQUIPEMENT DE TRANSMISSION DE DONNEES MEDIA
(54) Title: METHOD AND APPARATUS FOR MEDIA DATA TRANSMISSION



(57) Abrégé/Abstract:

Methods and apparatuses for processing media data for transmission in a data communication medium. A set of data indicates how to transmit a time related sequence of media data according to a transmission protocol. The set of data, includes a time

(57) **Abrégé(suite)/Abstract(continued):**

related sequence of data which is associated with the time related sequence of media data. The set of data may be utilized by a digital processing system to transmit the time related sequence of media data (e.g., by packets generated according to the transmission protocol and the set of data).

-5-

The movie header provides basic information about the overall presentation (its creation date, overall timescale, and so on). In the sequence of contained objects there is typically at least one track, which describes temporally presented data.

```
class Track {
    int(32)      size;
    char         type[4] = 'trak';
    TrackHeader th;
    contents     Atom[];
}
```

The track header provides relatively basic information about the track (its ID, timescale, and so on). Objects contained in the track might be references to other tracks (e.g. for complex compositing), or edit lists. In this sequence of contained objects there may be a media object, which describes the media which is presented when the track is played.

The media object contains declarations relating to the presentation required by the track (e.g. that it is sampled audio, or MIDI, or orientation information for a 3Dscene). The type of track is declared by its handler:

```
class handler {
    int(32)      size;
    char         type[4] = 'hdlr';
    int(8)       version;
    bit(24)      flags;
    char         handlertype[4];    -- mhlr for media handlers
    char         handlersubtype[4] -- vide for video, soun for
audio          char         manufacturer[4];
    bit(32)      handlerflags;
    bit(32)      handlerflagsmask;
    string       componentname;
}
```

Within the media information there is likewise a handler declaration for the data handler (which fetches media data), and a data information declaration, which defines which files contain the media data for the associated track. By using this declaration, movies may be built which span several files.

-6-

At the lowest level, a sample table is used which relates the temporal aspect of the track to the data stored in the file:

```
class sampletable {
    int(32)      size;
    char        type[4] = 'stbl';
    sampleddescription sd;
    timetosample    tts;
    syncsampletable syncs;
    sampletochunk   stoc;
    samplesize      ssize;
    chunkoffset     coffset;
    shadowsync      ssync;
}
```

The sample description contains information about the media (e.g. the compression formats used in video). The time-to-sample table relates time in the track, to the sample (by index) which should be displayed at that time. The sync sample table declares which of these are sync (key) samples, not dependent on other samples.

The sample-to-chunk object declares how to find the media data for a given sample, and its description given its index:

```
class sampletochunk {
    int(32)      size;
    char        type[4] = 'stsc';
    int(8)      version;
    bits(24)    flags;
    int(32)     entrycount;
    for (int i=0; i<entrycount; i++) {
        int(32)    firstchunk;
        int(32)    samplesperchunk;
        int(32)    sampledescriptionindex;
    }
}
```

The sample size table indicates the size of each sample. The chunkoffset table indicates the offset into the containing file of the start of each chunk.

Walking the above-described structure to find the appropriate data to display for a given time is fairly straightforward, generally involving indexing and

10c

In a further aspect, the present invention resides in a machine-readable medium containing executable computer program instructions, which when executed on a first digital processing system cause the first digital processing system to perform a method comprising: generating a set of data to indicate how to packetize a time related sequence of media data according to defined packetizing characteristics, wherein said set of data varies with different packetizing characteristics, wherein said set of data is a time related sequence of data associated with said time related sequence of media data; and storing said set of data.

Table 5 - Additional Data Table Entries

Entry length	Data type	Data description
8	0x00000000	Indicates there are no more entries in the table
9	'otyp'	Describes how offsets are described in the hints. The one byte of data has values described below in figure B.4. <i>This entry is mandatory in the additional data table.</i>
9	'msns'	Describes the size of media sample numbers. The one byte of data indicates how many bytes are used to specify media sample numbers. If this is not present, and media sample numbers are present in the sample data, the default value is 4 bytes.
9	'msos'	Describes the size of media sample offsets. The one byte of data indicates how many bytes are used to specify media sample offsets. If this is not present, and media sample offsets are present in the sample data, the default value is 4 bytes.
9	'fosz'	Describes the size of file offsets. The one byte of data indicates how many bytes are used to specify file offsets within samples. If this is not present, and file offsets are present in the sample data, the default value is 4 bytes.
Variable	'tmap'	Describes an abbreviated mapping of media tracks. Each 5 byte entry maps a 4 byte track ID to a 1 byte track reference number. This limits any given transport mux to containing no more than 256 media tracks, but this should not be a limiting factor, and this compression is useful in limiting the size of the hint track. The format of these 5 byte entries is specified below in figure B.5. <i>This entry is mandatory in the additional data table.</i>

Table 6- 'otyp' Values In the Additional Data Table

Value	Description
0	Samples are described in terms of media samples
1	Samples are described in terms of file offsets

Table 7 - Format of Entries in the 'tmap' Additional Data Entry

Length	Description
4	Original Track ID
1	Abbreviated track reference number used in samples

