(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년04월10일
(11) 등록번호 10-1379034
(24) 등록일자 2014년03월21일

(51) 국제특허분류(Int. Cl.)
H04N 5/76 (2006.01) *G11B 27/00* (2006.01)
(21) 출원번호 10-2008-7000499
(22) 출원일자(국제) 2007년05월10일
심사청구일자 2012년03월20일
(85) 번역문제출일자 2008년01월08일
(65) 공개번호 10-2009-0009769
(43) 공개일자 2009년01월23일
(86) 국제출원번호 PCT/JP2007/060081
(87) 국제공개번호 WO 2007/135932
국제공개일자 2007년11월29일
(30) 우선권주장
JP-P-2006-00138701 2006년05월18일 일본(JP)
(56) 선행기술조사문헌
WO2006030767 A1
JP2005027159 A
JP2005317076 A
전체 청구항 수 : 총 50 항

(73) 특허권자
소니 주식회사
일본국 도쿄도 미나토쿠 코난 1-7-1
(72) 발명자
마에 아츠시
일본 도쿄 미나토-쿠 코난 1-7-1 소니 가부시끼 가이샤내
아리도메 켄이치로
일본 도쿄 미나토-쿠 코난 1-7-1 소니 가부시끼 가이샤내
(뒷면에 계속)
(74) 대리인
신관호

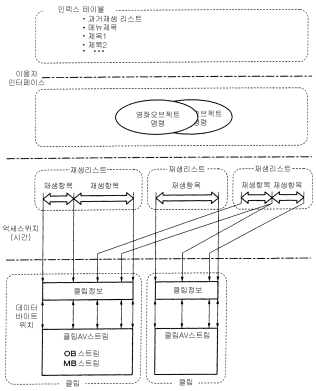
심사관 : 남옥우

(54) 발명의 명칭 **기록 장치, 기록 방법 및 기록 프로그램과 촬상 장치, 촬상방법 및 촬상 프로그램**

(57) 요 약

기록개시부터 정지동안에 생성되는 AV데이터를 파일로서 기록할 때의 이용자의 편리성을 향상시킨다. 기록개시부터 정지동안에 생성되는 AV데이터를 파일로서 기록하고, 재생 구간을 나타내는 정보와 점프 위치를 나타내는 마크 정보를 재생 리스트 파일에 추기한다. 이 때, 추기 후보의 재생 리스트 파일과 AV데이터의 재생 시각과 주소를 대응시킨 속성 파일과에 근거하여, 추기의 경우에, 재생 리스트 파일이나 속성 파일에 미리 설정된 제약을 만족하는지 아닌지가 판단된다.만족한다고 판단되면, 재생 구간 정보와 마크 정보가 추기 후보의 재생 리스트 파일에 추기된다. 만족하지 않는다고 판단되면, 신규로 재생 리스트 파일이 생성된다. 이용자는, 재생 리스트 파일이나 속성 파일에 미리 설정된 제약을 의식하지 않고, AV데이터의 기록 조작을 실시할 수 있다.

대 표 도 - 도1



(72) 발명자

이소베 유키오

일본 도쿄 미나토-쿠 코난 1-7-1 소니 가부시끼 가이샤내

모리모토 나오키

일본 도쿄 미나토-쿠 코난 1-7-1 소니 가부시끼 가이샤내

특허청구의 범위

청구항 1

비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 장치에 있어서,

비디오 데이터 및 오디오 데이터가 입력되는 데이터 입력부와,

상기 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력부와,

상기 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록부와,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성부를 가지며,

상기 관리 정보 생성부는,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보에 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 2

제 1항에 있어서,

상기 관리 정보 생성부는,

상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라, 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보에 포함되는 상기 소정의 제 2의 관리 정보에 대해서 추기하지 않는다고 판단한 경우, 상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를 포함하는 상기 제 2의 관리 정보를 신규로 생성하는 것을 특징으로 하는 기록 장치.

청구항 3

제 1항에 있어서,

상기 관리 정보 생성부는,

상기 스트림 파일의 기록에 따라 갱신되는 상기 제 2의 관리 정보 가운데, 가장 새롭게 갱신된 상기 제 2의 관리 정보를, 상기 소정의 제 2의 관리 정보로 하는 것을 특징으로 하는 기록 장치.

청구항 4

제 1항에 있어서,

상기 제 1의 관리 정보는,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 상기 스트림 파일의 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보가 저장된 파일로 구성되며,

상기 제 2의 관리 정보는,

상기 스트림 파일에 대해서 재생 개시점과 재생 종료점을 설정함으로써, 재생 구간을 지정하는 한 개 이상의 재생 구간 정보가 저장되며, 상기 스트림 파일에 대한 재생 시각 정보를 나타내는 마크 정보가 저장될 수 있는 파일로 구성되며,

상기 제 2의 관리 정보는, 상기 재생 구간 정보로부터 상기 제 1의 관리 정보를 참조하여, 상기 스트림 파일의 재생 방법을 나타내는 것을 특징으로 하는 기록 장치.

청구항 5

제 4항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의제 2의 관리 정보에 저장되는 상기 재생 구간 정보의 수에 근거하는 것을 특징으로 하는 기록 장치.

청구항 6

제 5항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 이미 저장된 상기 재생 구간 정보의 수가 미리 설정된 상한치의 미만이 된다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 7

제 4항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 마크 정보의 수에 근거하는 것을 특징으로 하는 기록 장치.

청구항 8

제 7항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 이미 저장된 상기 마크 정보의 수가 미리 설정된 상한치 미만이라고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 9

제 4항에 있어서,

상기 제 1의 관리 정보는, 대응하는 상기 스트림 파일에 저장되는 상기 비디오 데이터의 속성을 나타내는 비디오 속성 정보를 추가로 저장하며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 제 1의 관리 정보에 저장되는 상기 비디오 속성 정보에 근거하는 것을 특징으로 하는 기록 장치.

청구항 10

제 9항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

현재의 기록 모드에 근거하는 비디오 속성과, 상기 소정의 제 2의 관리 정보로부터 참조되는 상기 제 1의 관리 정보중 적어도 한 개의 상기 관리 정보에 저장되는 비디오 속성에 근거하여 이루어지는 것을 특징으로 하는 기록 장치.

청구항 11

제 10항에 있어서,

상기 관리 정보 생성부는,

상기 기록 모드에 근거하는 비디오 속성과, 상기 제 1의 관리 정보에 저장되는 비디오 속성의 사이에, 화상 범위 사이즈, 애스펙트비, 프레임 속도 및 주사 방식중 적어도 한 개가 일치하지 않는 경우에,

상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기 하지 않도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 12

제 4항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보의 파일 사이즈에 근거하는 것을 특징으로 하는 기록 장치.

청구항 13

제 12항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 기록에 따라 생성되는 상기 재생 구간 정보 및 상기 마크 정보의 데이터 사이즈에 근거하여, 한 개의 상기 스트림 파일이 기록되어, 상기 제 2의 관리 정보의 파일 사이즈가 미리 설정된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 14

제 4항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계 파일 사이즈에 근거하는 것을 특징으로 하는 기록 장치.

청구항 15

제 14항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 보증된 최저 연속시간의 기록에 따라 생성되는 상기 제 1의 관리 정보의 데이터 사이즈에 근거하여, 한 개의 상기 스트림 파일이 기록되며, 상기 소정의 제 2의 관리 정보로부터 참조되는 상기 제 1의 관리 정보의 합계 파일 사이즈가 미리 설정된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 16

제 4항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보에 포함되는 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수에 근거하는 것을 특징으로 하는 기록 장치.

청구항 17

제 16항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 보증된 최저 연속시간의 기록에 따라 생성되는 상기 제 1의 관리 정보로 포함되는 상기 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수에 근거하여, 한 개의 상기 스트림 파일이 기록되며, 상기 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계가 상기 제 2의 관리 정보에 대해서 미리 설치된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 18

제 4항에 있어서,

상기 제 2의 관리 정보는, 기기 고유 정보를 추가로 저장 가능하게 되며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 기기 고유 정보에 근거하는 것을 특징으로 하는 기록 장치.

청구항 19

제 18항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 상기 기기 고유 정보가 저장되는 경우에, 상기 기기 고유 정보가 자기 또는 자기와 동등의 기기에 의해 생성되었다고 판단되면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 20

제 4항에 있어서,

상기 제 2의 관리 정보는, 상기 제 2의 관리 정보의 최종 갱신자를 나타내는 정보를 추가로 저장 가능하게 되며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 최종 갱신자 정보에 근거하는 것을 특징으로 하는 기록 장치.

청구항 21

제 20항에 있어서,

상기 소정의 제 2의 관리 정보에 상기 최종 갱신자 정보가 저장되는 경우에, 상기 최종 갱신자 정보가 자신의 기기를 나타내고 있다고 판단되면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 22

제 1항에 있어서,

상기 관리 정보 생성부는,

한 개 또는 복수의 항목에 대해 상기 추기 여부 판단을 실시하며, 상기 추기 여부 판단의 결과, 상기 한 개 또는 복수의 항목중 한 개에서도 추기 불가를 나타내고 있으면, 상기 기록 매체에 기록되는 상기 스트림 파일에 대응하는 상기 제 2의 관리 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하지 않도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 장치.

청구항 23

제 22항에 있어서,

상기 한 개 또는 복수의 항목은,

상기 소정의 제 2의 관리 정보에 저장되는 재생 구간 정보의 수와, 상기 소정의 제 2의 관리 정보에 저장되는 마크 정보의 수와, 상기 제 1의 관리 정보에 저장되는 비디오 속성 정보와, 상기 소정의 제 2의 관리 정보의 파일 사이즈와, 상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계 파일 사이즈와, 상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보에 포함되는 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수와, 상기 소정의 제 2의 관리 정보에 저장되는 기기 고유 정보와, 상기 소정의 제 2의 관리 정보에 저장되는 최종 갱신자 정보중, 적어도 한 개의 항목을 포함하는 것을 특징으로 하는 기록 장치.

청구항 24

비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 방법에 있어서,

데이터 입력부에 입력된 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과,

상기 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며,

상기 관리 정보 생성의 스텝은,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력의 스텝에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보로 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 방법.

청구항 25

비디오 데이터와 오디오 데이터를 다중화하여, 기록 매체에 기록하는 기록 방법을, 컴퓨터 장치로 실행시키는 프로그램을 가지는 기록 기기에 있어서,

상기 프로그램은,

데이터 입력부에 입력된 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과,

상기 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며,

상기 관리 정보 생성의 스텝은,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력의 스텝에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보로 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 기기.

청구항 26

촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치에 있어서,

피사체를 촬상하여 비디오 데이터를 출력하는 촬상부와,

음성을 수음하여 오디오 데이터를 출력하는 수음부와,

상기 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록부와,

상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에 대한 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 조작부와,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성부를 가지며,

상기 관리 정보 생성부는,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보에 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는 지의 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 27

제 26항에 있어서,

상기 관리 정보 생성부는,

상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라, 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보에 포함되는 상기 소정의 제 2의 관리 정보에 대해서 추기하지 않는다고 판단했을 경우, 상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를 포함하는 상기 제 2의 관리 정보를 신규로 생성하는 것을 특징으로 하는 촬상 장치.

청구항 28

제 26항에 있어서,

상기 관리 정보 생성부는,

상기 스트림 파일의 기록에 따라 갱신되는 상기 제 2의 관리 정보 가운데, 가장 새롭게 갱신된 상기 제 2의 관리 정보를, 상기 소정의 제 2의 관리 정보로 하는 것을 특징으로 하는 촬상 장치.

청구항 29

제 26항에 있어서,

상기 제 1의 관리 정보는,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 상기 스트림 파일의 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보가 저장된 파일로 구성되며,

상기 제 2의 관리 정보는,

상기 스트림 파일에 대해서 재생 개시점과 재생 종료점을 설정하여, 재생 구간을 지정하는 한 개 이상의 재생 구간 정보가 저장되며, 상기 스트림 파일에 대한 재생 시각 정보를 나타내는 마크 정보가 저장될 수 있는 파일로 구성되며,

상기 제 2의 관리 정보는, 상기 재생 구간 정보로부터 상기 제 1의 관리 정보를 참조하여, 상기 스트림 파일의 재생 방법을 나타내는 것을 특징으로 하는 촬상 장치.

청구항 30

제 29항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의제 2의 관리 정보에 저장되는 상기 재생 구간 정보의 수에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 31

제 30항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 이미 저장된 상기 재생 구간 정보의 수가 미리 설정된 상한치의 미만이 된다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 32

제 29항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 마크 정보의 수에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 33

제 32항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 이미 저장된 상기 마크 정보의 수가 미리 설정된 상한치 미만이라고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 34

제 29항에 있어서,

상기 제 1의 관리 정보는, 대응하는 상기 스트림 파일에 저장되는 상기 비디오 데이터의 속성을 나타내는 비디오 속성 정보를 추가로 저장하며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 제 1의 관리 정보에 저장되는 상기 비디오 속성 정보에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 35

제 34항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

현재의 기록 모드에 근거하는 비디오 속성과, 상기 소정의 제 2의 관리 정보로부터 참조되는 상기 제 1의 관리 정보중 적어도 한 개의 상기 제 1의 관리 정보에 저장되는 비디오 속성에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 36

제 35항에 있어서,

상기 관리 정보 생성부는,

상기 기록 모드에 근거하는 비디오 속성과, 상기 제 1의 관리 정보에 저장되는 비디오 속성의 사이에, 화상 범위 사이즈, 애스펙트비, 프레임 속도 및 주사 방식중 적어도 한 개가 일치하지 않는 경우에,

상기 기록 지시 입력부에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에

대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기 하지 않도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 37

제 29항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보의 파일 사이즈에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 38

제 37항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 기록에 따라, 생성되는 상기 재생 구간 정보 및 상기 마크 정보의 데이터 사이즈에 근거하여, 한 개의 상기 스트림 파일이 기록되어, 상기 제 2의 관리 정보의 파일 사이즈가 미리 설정된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 39

제 29항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계 파일 사이즈에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 40

제 39항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 보증된 최저 연속시간의 기록에 따라 생성되는 상기 제 1의 관리 정보의 데이터 사이즈에 근거하여, 한 개의 상기 스트림 파일이 기록되며, 상기 소정의 제 2의 관리 정보로부터 참조되는 상기 제 1의 관리 정보의 합계 파일 사이즈가 미리 설정된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 41

제 29항에 있어서,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보에 포함되는 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 42

제 41항에 있어서,

상기 관리 정보 생성부는,

한 개의 상기 스트림 파일의 보증된 최저 연속시간의 기록에 따라 생성되는 상기 제 1의 관리 정보로 포함되는 상기 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수에 근거하여, 한 개의 상기 스트림 파일이 기록되며, 상기 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계가 상기 제 2의 관리

정보에 대해서 미리 설치된 상한을 넘지 않는다고 판단하면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 43

제 29항에 있어서,

상기 제 2의 관리 정보는, 기기 고유 정보를 추가로 저장 가능하며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 기기 고유 정보에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 44

제 43항에 있어서,

상기 관리 정보 생성부는,

상기 소정의 제 2의 관리 정보에 상기 기기 고유 정보가 저장되는 경우에, 상기 기기 고유 정보가 자기 또는 자기와 동등의 기기에 의해 생성되었다고 판단되면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 45

제 29항에 있어서,

상기 제 2의 관리 정보는, 상기 제 2의 관리 정보의 최종 갱신자를 나타내는 정보를 추가로 저장 가능하며,

상기 관리 정보 생성부에 의한 상기 추기 여부 판단은,

상기 소정의 제 2의 관리 정보에 저장되는 상기 최종 갱신자 정보에 근거하는 것을 특징으로 하는 촬상 장치.

청구항 46

제 45항에 있어서,

상기 소정의 제 2의 관리 정보에 상기 최종 갱신자 정보가 저장되는 경우에, 상기 최종 갱신자 정보가 자신의 기기를 나타내고 있다고 판단되면,

상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 47

제 26항에 있어서,

상기 관리 정보 생성부는,

한 개 또는 복수의 항목에 대해 상기 추기 여부 판단을 실시하며, 상기 추기 여부 판단의 결과, 상기 한 개 또는 복수의 항목중 한 개에서도 추기 불가를 나타내고 있으면, 상기 기록 매체에 기록되는 상기 스트림 파일에 대응하는 상기 제 2의 관리 정보를, 상기 소정의 제 2의 관리 정보에 대해서 추기하지 않도록 상기 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치.

청구항 48

제 47항에 있어서,

상기 한 개 또는 복수의 항목은,

상기 소정의 제 2의 관리 정보에 저장되는 재생 구간 정보의 수와, 상기 소정의 제 2의 관리 정보에 저장되는

마크 정보의 수와, 상기 제 1의 관리 정보에 저장되는 비디오 속성 정보와, 상기 소정의 제 2의 관리 정보의 파일 사이즈와, 상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보의 합계 파일 사이즈와, 상기 소정의 제 2의 관리 정보로부터 참조되는 모든 상기 제 1의 관리 정보에 포함되는 재생 시각 정보와 주소 정보와의 대응 관계를 나타내는 정보의 수와, 상기 소정의 제 2의 관리 정보에 저장되는 기기 고유 정보와, 상기 소정의 제 2의 관리 정보에 저장되는 최종 갱신자 정보중, 적어도 한 개의 항목을 포함하는 것을 특징으로 하는 촬상 장치.

청구항 49

촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치의 촬상 방법에 있어서,

상기 촬상 방법은,

촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과,

조작부에 대한 상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에의 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 스텝과,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며,

상기 관리 정보 생성의 스텝은,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력의 스텝에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보로 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 방법.

청구항 50

촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치의 촬상 방법을, 컴퓨터 장치에 실행시키는 촬상 프로그램을 가지는 촬상 기기에 있어서,

상기 촬상 프로그램은,

촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과,

조작부에 대한 상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에의 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 스텝과,

상기 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 상기 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 상기 기록 매체에 기록되는 상기 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며,

상기 관리 정보 생성의 스텝은,

기존의 상기 재생 관리 정보에 근거하여, 상기 기록 지시 입력의 스텝에 의한 상기 기록 개시의 지시에 따라 상기 기록 매체에 기록되는 상기 스트림 파일에 대한 상기 재생 방법을 나타내는 정보를, 상기 기존의 재생 관리 정보로 포함되는 소정의 상기 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 기기.

명 세 서

기 술 분 야

[0001] 이 발명은, 비디오 데이터와 오디오 데이터를 다중화한 스트림 데이터를 기록 매체에 기록하는데 적합한 기록 장치, 기록 방법 및 기록 프로그램 및 촬상 장치, 촬상 방법 및 촬상 프로그램에 관한 것이다.

배 경 기 술

[0002] 종래부터, 기록 가능하고 기록 재생장치로부터 분리가능한 동시에, 기록 용량이 비교적 크고, 비디오 데이터와 오디오 데이터로 구성되는 AV(Audio/Video) 데이터를 기록하는데 적합한 기록 매체로서, 4.7GB(Giga Byte) 이상의 기록 용량을 가지는 DVD(Digital Versatile Disc)가 보급되어 있다. 특허 문헌 「특개 2004-350251」에는, 기록 가능한 타입의 DVD에 대해서 DVD-Video 포맷으로 기록하는 촬상 장치가 기재되어 있다.

[0003] 그런데, 기록 매체에 대한 비디오 데이터 및 오디오 데이터의 기록은, 일반적으로는, 기록 개시 조작으로부터 기록 정지 조작 동안에 생성되는 비디오 데이터를 단위로 하여 행해진다. 그리고, 기록 개시 조작으로부터 기록 정지 조작동안에 생성되는 비디오 데이터에 대해서, 재생 리스트 정보를 이용하여 재생 구간이나 재생 순서를 지정하는 것이 일반적으로 행해지고 있다. 기록된 AV스트림을, 기록 개시 조작으로부터 기록 정지 조작 동안에 생성되는 비디오 데이터를 단위로 하여, 재생 리스트 정보를 이용하여 관리하고, 기록 매체상의 AV스트림을 가공하지 않고, 해당 AV스트림의 재생 구간이나 재생 순서를 자유롭게 설정한 편집을 용이하게 실시할 수 있다.

[0004] 여기서, 비디오 데이터 및 오디오 데이터를 파일로서 기록 매체에 기록하는 것을 생각한다. 예를 들면, 비디오 데이터 및 오디오 데이터를, 기록 개시 조작으로부터 기록 정지 조작동안에 생성되는 데이터를 단위로 하여 파일로서 기록한다. 비디오 데이터 및 오디오 데이터를 파일로서 기록하므로, 컴퓨터 장치 등의 다른 장치와의 친화성이 증가하며, 기록된 데이터를 보다 효과적으로 활용할 수 있는 것이 기대된다.

[0005] 이와 같이, 비디오 데이터 및 오디오 데이터를 파일로서 기록하는 경우, 일련의 기록 개시 및 정지의 조작에 따라 한 개의 기록 매체에 차례차례 기록되어 가는 복수 단위의 비디오 데이터 및 오디오 데이터를 기록순서에 따라 연속적으로 재생하기 위해서는, 어떠한 대책이 필요하다. 예를 들면, 일련의 기록 개시 및 정지의 조작에 따라 한 개의 기록 매체에 차례차례 기록되어 가는 복수 단위의 비디오 데이터 및 오디오 데이터의 정보를, 재생 리스트 정보가 저장되는 한 개의 재생 리스트 파일에 대해서 차례차례, 추기해 나가도록 제어하는 것이 생각된다. 이와 같이 기록 제어를 실시하여, 기록 매체에 단위마다 기록된 일련의 복수 단위의 비디오 데이터 및 오디오 데이터를 연속적으로 재생하는 것이 용이해진다.

발명의 상세한 설명

[0006] 기록 개시 조작 및 정지 조작마다 차례차례 기록되는 비디오 데이터 및 오디오 데이터의 정보를 재생 리스트 파일에 대해서 추기해 나가도록 제어하는 경우, 기록 개시 및 정지의 조작이 반복될 때마다, 재생 리스트 파일의 데이터 사이즈가 증대하게 된다. 한편, 기록 장치나 재생장치에서는, 재생 리스트 파일을 일단 메모리에 읽어들이고, 파일로서 기록된 비디오 데이터 및 오디오 데이터의 재생 제어를 실시하는 것이 생각된다. 메모리에는, 기록 매체에 기록되는 비디오 데이터 및 오디오 데이터를 관리하는 다른 관리 정보도 읽기 위해, 재생 리스트 파일의 사이즈가 증대하면 메모리의 용량을 압박하므로, 다른 처리에 지장이 되는 우려가 있다고 하는 문제점이 있었다.

[0007] 또한, 비디오 데이터 및 오디오 데이터를 파일로서 기록하는 경우, 재생시에 비디오 데이터내의 프레임을 검색하기도 하며, 프레임 단위로 편집을 용이하게 실시할 수 있는 구조를 제공할 필요가 있다. 이를 위해서는, 예를 들면 비디오 데이터에 있어서의 재생 시간 정보와, 비디오 데이터가 저장되는 파일내의 주소를 대응시키는 것이 생각된다. 이 재생 시간 정보와 파일내 주소와의 대응 관계를 나타내는 데이터는, 기록에 따라 증가하기 때문에, 이 경우에 있어서도, 기록시나 재생시에 있어서 메모리 용량을 압박해 버려, 다른 처리에 지장이 나올 우려가 있다는 문제점이 있었다.

[0008] 게다가, 메모리 용량이 압박받는 상태에서 예를 들면 기록을 속행하려고 한경우, 시스템에 의해서 강제적으로 비디오 데이터 및 오디오 데이터의 기록이 정지되거나 시스템이 행업(hangup)이 될 가능성이 있다는 문제점이 있었다.

[0009] 또, 최근에는, 일반적으로 이용되는 비디오 포맷의 종류가 보다 다채롭게 되어 오고 있다. 예를 들면, 라인수와 주사 방법에 관해서는, 인터레이스(interlace) 주사의 포맷에서는, 라인수가 480개의 480i, 라인수가 576개의 576i, 라인수가 1080개의 1080i등이 있으며, 프로그래시브 주사(progressive scanning)의 포맷에서는, 라

인수가 각각 480개, 576개, 720개 및 1080개의, 480p, 576p, 720p 및 1080p등이 규정되어 있다.

[0010] 기록 장치에 있어서도, 상술과 같은 다종류의 포맷에 대응하는 기록이 가능한 것이 요구된다. 그와 함께, 한 개의 기록 매체에 대해서 복수 종류의 포맷에 의한 비디오 데이터를 혼재하여 기록 가능한 것이 요구된다. 그렇지만, 상술한 것처럼, 기록 매체에 파일로서 차례차례 기록되는 비디오 데이터 및 오디오 데이터의 정보를, 한 개의 재생 리스트 파일에 대해서 추기해 나가도록 기록 제어를 실시하는 경우, 해당 기록 매체를 재생하는 재생장치측에서 문제가 발생할 가능성이 있다. 예를 들면, 재생 리스트 파일에 따라 차례차례, 파일로서 기록된 비디오 데이터 및 오디오 데이터를 재생해 나갈 때에, 지금까지 재생하고 있던 비디오 데이터와 포맷이 다른 비디오 데이터를 다음에 연속적으로 재생하려고 해도, 디코더의 처리의 변경이 늦거나, 디코더가 행업하여 동작이 정지해 버릴 우려가 있다는 문제점이 있었다.

[0011] 따라서, 이 발명의 목적은, 기록 개시 조작으로부터 기록 정지 조작동안에 생성되는 비디오 데이터 및 오디오 데이터를 파일로서 기록할 때의 이용자의 편리성을 향상시킬 수 있는 기록 장치, 기록 방법 및 기록 프로그램과 촬상 장치, 촬상 방법 및 촬상 프로그램을 제공하는 것에 있다.

[0012] 상술한 과제를 해결하기 위해서, 제 1의 발명은, 비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 장치에 있어서, 비디오 데이터 및 오디오 데이터가 입력되는 데이터 입력부와 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력부와 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록부와 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성부를 가지며, 관리 정보 생성부는, 기존의 재생 관리 정보에 근거하여, 기록 지시 입력부에 의한 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는 지 아닌지의 여부 판단을 실시하는 것을 특징으로 하는 기록 장치이다.

[0013] 또, 제 2의 발명은, 비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 방법에 있어서, 데이터 입력부에 입력된 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과, 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과, 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며, 관리 정보 생성의 스텝은, 기존의 재생 관리 정보에 근거하여, 기록 지시 입력의 스텝에 의한 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보로 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 방법이다.

[0014] 또한, 제 3의 발명은, 비디오 데이터와 오디오 데이터를 다중화하여, 기록 매체에 기록하는 기록 방법을 컴퓨터 장치에 실행시키는 기록 프로그램에 있어서, 기록 방법은, 데이터 입력부에 입력된 비디오 데이터 및 오디오 데이터의 기록 개시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과, 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과, 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며, 관리 정보 생성의 스텝은, 기존의 재생 관리 정보에 근거하여, 기록 지시 입력의 스텝에 의한 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보로 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 기록 프로그램이다.

[0015] 또한, 제 4의 발명은, 촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치에 있어서, 피사체를 촬상하여 비디오 데이터를 출력하는 촬상부와, 음성을 수음하여 오디오 데이터를 출력하는 수음부와, 비디오 데이터 및 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록부와, 비디오 데이터 및 오디오 데이터의 기록 매체에 대한 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 조작부와, 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성

하는 관리 정보 생성부를 가지며, 관리 정보 생성부는, 기존의 재생 관리 정보에 근거하여, 조작부에 대한 이용자 조작에 따르는 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는 지의 여부 판단을 실시하는 것을 특징으로 하는 촬상 장치이다.

[0016] 또, 제 5의 발명은, 촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치의 촬상 방법에 있어서, 촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과, 조작부에 대한 비디오 데이터 및 오디오 데이터의 기록 매체로의 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 스텝과, 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며, 관리 정보 생성의 스텝은, 기존의 재생 관리 정보에 근거하여, 조작부에 대한 이용자 조작에 따르는 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보로 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 방법이다.

[0017] 또한 제 6의 발명은, 촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하여 기록 매체에 기록하는 촬상 장치의 촬상 방법을 컴퓨터 장치에 실행시키는 촬상 프로그램에 있어서, 촬상부에서 피사체를 촬상하여 얻어지는 비디오 데이터와, 수음부에서 음성을 수음하여 얻어지는 오디오 데이터를 다중화하고, 다중화된 스트림을 스트림 파일로서 기록 매체에 기록하는 기록의 스텝과, 조작부에 대한 비디오 데이터 및 오디오 데이터의 기록 매체로의 기록 개시 및 기록 정지를 지시하는 이용자 조작을 받아들이는 스텝과, 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 스트림 파일의 재생 방법을 나타내는 정보를 포함한 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하는 관리 정보 생성의 스텝을 가지며, 관리 정보 생성의 스텝은, 기존의 재생 관리 정보에 근거하여, 조작부에 대한 이용자 조작에 따르는 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해서 추기하는지 아닌지의 추기 여부 판단을 실시하는 것을 특징으로 하는 촬상 프로그램이다.

실 시 예

[0071] 이하, 이 발명의 실시의 한 형태를, 도면을 참조하면서 설명한다. 먼저, 이해를 용이하게 하기 위해서, 이 발명에 적용 가능한 일례의 포맷(이하, AVCHD 포맷이라고 부른다)에 대해 설명한다. AVCHD 포맷은, 비디오 데이터와 오디오 데이터가 소정의 형태로 다중화된 AV(Audio/Video) 스트림을 기록 가능한 기록 매체에 기록하는 기록 포맷으로서 현재 제안되어 있는 것이며, 기록 매체에 기록된 AV스트림을, 클립 단위로 플레이 리스트를 이용하여 관리 가능하게 하고 있다.

[0072] 예를 들면 ITU-T(International Telecommunication Union-Telecommunication Standarization Sector) 권고 H.264혹은 ISO(International Organization for Standarization)/IEC(International Electrotechnical Commission) 국제 표준 14496-10(MPEG-4 파트 10) Advanced Video Coding(이하, H.264|AVC로 약칭한다)에 규정되어 있는 부호화 방식으로 부호화되고, MPEG2 시스템에 따라 다중화된 비트 스트림은, 클립 AV스트림(또는 AV스트림)이라고 칭해진다. 클립 AV스트림은, 소정의 파일 시스템에 의해 파일로서 디스크에 기록된다. 이 파일을, 클립 AV스트림 파일(또는 AV스트림 파일)로 칭한다.

[0073] 클립 AV스트림 파일은, 파일 시스템상에서의 관리 단위이며, 이용자에게 있어서 반드시 알기 쉬운 관리 단위라고 할 수 없다. 이용자의 편리성을 생각했을 경우, 복수의 클립 AV스트림 파일로 분할된 영상 콘텐츠를 하나로 정리하여 재생하는 구조나, 클립 AV스트림 파일의 일부만을 재생하는 구조, 또는, 특수 재생이나 선두 부분 재생을 매끄럽게 실시하기 위한 정보 등을 데이터 베이스로서 디스크에 기록해 둘 필요가 있다.

[0074] 도 1은, 이 발명에 적용 가능한 AVCHD 포맷에 규정되는 데이터 모델을 개략적으로 도시하고 있다. 이 AVCHD 포맷에 의하면, 데이터 구조는, 도 1에 도시한 바와 같이 4층의 레이어(layer)로 구성된다. 가장 최하층의 레이어는, 클립 AV스트림이 배치되는 레이어이다(편의상, 클립 레이어라고 부른다). 그 위의 레이어는, 클립 AV스트림에 대한 재생 위치를 지정하기 위한, 플레이 리스트(PlayList)와 플레이 아이템(PlayItem)이 배치되는 레이어이다(편의상, 플레이 리스트 레이어라고 부른다). 게다가 그 위의 레이어는, 플레이 리스트에 대해서 재생 순서 등을 지정하는 명령(command)으로 구성되는 무비 오브젝트(Movie Object)등이 배치되는 레이

어이다(편의상, 오브젝트 레이어라고 부른다). 최상층의 레이어는, 기록 매체에 저장되는 타이틀 등을 관리하는 인덱스 테이블이 배치된다(편의상, 인덱스 레이어라고 부른다.).

[0075] 클립 레이어에 대해 설명한다. 클립 AV스트림은, 비디오 데이터나 오디오 데이터가 MPEG2TS(트랜스포트 스트림)의 형식 등에 다중화된 비트 스트림이다.이 클립 AV스트림에 관한 정보가 클립 정보(Clip Information)로서 파일에 기록된다.

[0076] 또, 클립 AV스트림에는, 자막을 표시하는 그래픽스 스트림인 OB스트림(Overlay Bitmap stream)이나, 메뉴 표시 등에 이용되는 데이터(버튼 화상 데이터등)를 스트림으로 하는 MB스트림(Menu Bitmap stream)을 다중화할 수 있다.

[0077] 클립 AV스트림 파일과 대응하는 클립 정보가 기록된 클립 정보 파일을 하나로 모은 오브젝트라고 보고, 클립(Clip)으로 칭한다. 즉, 클립은, 클립 AV스트림과 클립 정보로 구성되는, 하나의 오브젝트이다.

[0078] 파일은, 일반적으로, 바이트열로 취급된다. 클립 AV스트림 파일의 콘텐츠는, 시간축상에 전개되며, 클립중의 엔트리 포인트는, 주로 시간 베이스로 지정된다. 소정의 클립에 대한 액세스 포인트의 타임 스탬프가 부여된 경우, 클립 AV스트림 파일중에서 데이터의 읽기를 개시해야 할 주소 정보를 찾아내기 위해서, 클립 정보 파일을 이용할 수 있다.

[0079] 플레이 리스트 레이어에 대해 설명한다. 플레이 리스트는, 재생하는 AV스트림 파일의 지정과 지정된 AV스트림 파일의 재생 위치를 지정하는 재생 개시점(IN점)과 재생 종료점(OUT점)의 집합으로 구성된다. 이 재생 개시점과 재생 종료점의 정보를 1조로 한 것은, 플레이 아이템(PlayItem)이라고 칭해진다. 플레이 리스트는, 플레이 아이템의 집합으로 구성된다. 플레이 아이템을 재생한다고 하는 것은, 그 플레이 아이템에 참조되는 AV스트림 파일의 일부분을 재생한다고 하는 것이 된다. 즉, 플레이 아이템중의 IN점 및 OUT점 정보에 근거하여, 클립중의 대응하는 구간이 재생된다.

[0080] 오브젝트 레이어에 대해 설명한다. 무비 오브젝트는, 네비게이션 명령 프로그램과 무비 오브젝트를 연합시키는 터미널 정보를 포함한다. 네비게이션 프로그램은, 플레이 리스트의 재생을 제어하기 위한 명령(네비게이션 명령 : navigation command)이다.

[0081] 인덱스 레이어에 대해 설명한다. 인덱스 레이어는, 인덱스 테이블(Index Table)로 구성된다. 인덱스 테이블은, 기록 매체에 기록된 콘텐츠의 타이틀을 정의하는, 톱 레벨의 테이블이다. 인덱스 테이블에 저장되어 있는 타이틀 정보에 근거하여, 재생기에 항상 내장되어 있는 시스템 소프트웨어중의 모듈 매니저에 의해 기록 매체의 재생이 제어된다.

[0082] 즉, 도 2에 개략적으로 도시된 바와 같이, 인덱스 테이블중의 임의의 엔트리는, 타이틀이라고 칭해지며, 인덱스 테이블에 엔트리되는 퍼스트 플레이백 타이틀(First PlaybackTitle), 메뉴 타이틀(MenuTitle) 및 무비 타이틀(MovieTitle)(#1,#2,???)은, 모두 타이틀이다. 각 타이틀은, 무비 오브젝트에 대한 링크를 나타낸다.

[0083] 이해를 용이하게 하기 위해 재생 전용의 기록 매체를 예로 들면, 예를 들면, 퍼스트 플레이백 타이틀은, 해당 기록 매체에 저장되는 콘텐츠가 영화이면, 영화 본편에 앞서 상영되는 영화 회사의 선전용 영상(트레일러 : trailer)에 대응한다.메뉴 타이틀은, 예를 들면 콘텐츠가 영화인 경우, 본편 재생, 챕터 검색, 자막이나 언어 설정, 특전 영상 재생 등을 선택하기 위한 메뉴 화면에 대응한다. 또, 무비 타이틀은, 메뉴 타이틀로부터 선택되는 각 영상이다. 게다가 타이틀이 메뉴 화면인 구성도 가능하다.

[0084] 도 3은, 상술한 바와 같은 클립 AV스트림, 클립 정보(Stream Attributes), 클립, 플레이 아이템 및 플레이 리스트의 관계를 나타내는 UML(Unified Modeling Language) 도면이다. 플레이 리스트는, 한 개 또는 복수의 플레이 아이템에 대응되며, 플레이 아이템은, 한 개의 클립에 대응된다. 한 개의 클립에 대해서, 각각 개시점 및 / 또는 종료점이 다른 복수의 플레이 아이템을 대응시키는 것이 가능하다. 한 개의 클립으로부터 한 개의 클립 AV스트림 파일이 참조된다. 이와 같이, 한 개의 클립으로부터 한 개의 클립 정보 파일이 참조된다. 또, 클립 AV스트림 파일과 클립 정보 파일이란, 1 대 1의 대응 관계를 가진다. 이러한 구조를 정의함으로써, 클립 AV스트림 파일을 변경하지 않고, 임의의 부분만큼을 재생하는, 비파괴의 재생 순서 지정을 실시하는 것이 가능해진다.

[0085] 또, 도 4와 같이, 복수의 플레이 리스트로부터 동일한 클립을 참조할 수도 있다. 또, 한 개의 플레이 리스트로부터 복수의 클립을 지정할 수도 있다. 클립은, 플레이 리스트중의 플레이 아이템에 나타나는 IN점 및 OUT점에 의해, 참조된다. 도 4의 예에서는, 클립(300)은, 플레이 리스트(310)의 플레이 아이템(320)으로부터

참조됨과 동시에, 플레이 리스트(311)를 구성하는 플레이 아이템(321 및 322)중 플레이 아이템(321)으로부터, IN점 및 OUT점으로 나타나는 구간이 참조된다. 또, 클립 (301)은, 플레이 리스트(311)의 플레이 아이템(322)으로부터 IN점 및 OUT점으로 나타나는 구간이 참조됨과 동시에, 플레이 리스트(312)의 플레이 아이템(323 및 324) 가운데, 플레이 아이템(323)의 IN점 및 OUT점으로 나타나는 구간이 참조된다. 도 4의 예에서는, 클립(301)은, 또 다른 플레이 리스트로부터도 참조되어 있다.

[0086] 다음에, AVCHD 포맷에 의한, 기록 매체에 기록되는 파일의 관리 구조에 대해서, 도 5를 이용하여 설명한다. 파일은, 디렉토리 구조에 의해 계층적으로 관리된다. 기록 매체상에는, 먼저, 한 개의 디렉토리(도 5의 예에서는 루트(root) 디렉토리)가 작성된다. 이 디렉토리의 아래가, 한 개의 기록 재생 시스템으로 관리되는 범위로 설정된다.

[0087] 루트 디렉토리의 아래에, 디렉토리 "BDMV"가 배치된다. 또한 필요에 따라서, 루트 디렉토리의 아래에 디렉토리 "AVCHDTN"가 배치된다. 디렉토리 "AVCHDTN"에는, 예를 들면 클립의 대표 화상을 소정 사이즈로 축소한 섬네일(thumbnail) 파일이 배치된다. 디렉토리 "BDMV"에, 도 1을 이용하여 설명한 데이터 구조가 저장된다.

[0088] 디렉토리 "BDMV"의 직하에는, 파일 즉, 파일 "index.bdmv" 및 파일"MovieObject.bdmv"의 2개만을 둘 수 있다. 또, 디렉토리 "BDMV" 아래에, 디렉토리 "PLAYLIST", 디렉토리 "CLIPINF", 디렉토리 "STREAM" 및 디렉토리 "BACKUP"이 배치된다. 디렉토리 "BACKUP"은, 각 디렉토리 및 파일의 백업이 저장된다.

[0089] 파일 "index.bdmv"는, 디렉토리 BDMV의 내용에 대해 기술된다. 즉, 이 파일 "index.bdmv"가 상술한 최상층의 레이어인 인덱스 레이어에 있어서의 인덱스 테이블에 대응한다. 또, 파일 MovieObject.bdmv는, 한 개 이상의 무비 오브젝트의 정보가 저장된다. 즉, 이 파일 "MovieObject.bdmv"가 상술한 오브젝트 레이어에 대응한다.

[0090] 디렉토리 "PLAYLIST"는, 플레이 리스트의 데이터 베이스가 배치되는 디렉토리이다. 즉, 디렉토리 "PLAYLIST"는, 플레이 리스트에 관한 파일인 파일"xxxxx.mpls"를 포함한다. 파일 "xxxxx.mpls"는, 플레이 리스트의 각각에 대해 작성되는 파일이다. 파일명에 있어서, "."(피리어드)의 이전의 "xxxxx"는, 5자리수의 숫자로 되며, 피리어드의 뒤의 "mpls"는, 이 타입의 파일에 고정적으로 형성되는 확장자(extension)이다.

[0091] 디렉토리 "CLIPINF"는, 클립의 데이터 베이스가 놓여지는 디렉토리이다. 즉, 디렉토리 CLIPINF"는, 클립 AV스트림 파일의 각각에 대응하는 클립 정보 파일인 파일"zzzzz.clpi"를 포함한다. 파일명에 있어서, "."(피리어드)의 이전의"zzzzz"는, 5자리수의 숫자로 되며, 피리어드의 뒤의 "clpi"는, 이 타입의 파일에 고정적으로 형성되는 확장자(extension)이다.

[0092] 디렉토리 "STREAM"은, 실체로서의 AV스트림 파일이 놓여지는 디렉토리이다.즉, 디렉토리 "STREAM"은, 클립 정보 파일의 각각에 대응하는 클립 AV스트림 파일을 포함한다. 클립 AV스트림 파일은, MPEG2(Moving Pictures Experts Group 2)의 트랜스포트 스트림(이하, MPEG2TS로 약칭한다)으로 구성되어, 파일명이"zzzzz.m2ts"로 된다. 파일명에 있어서, 피리어드의 이전의 "zzzzz"는, 대응하는 클립 정보 파일과 동일한 것이므로, 클립 정보 파일과 이 클립 AV스트림 파일과의 대응 관계를 용이하게 파악할 수 있다.

[0093] 한편, 디렉토리 "AVCHDTN"는, 2종류의 섬네일파일(thumbnail.tidx 및 thumbnail.tdt2)을 둘 수 있다. 섬네일파일(thumbnail.tidx)에는, 소정의 방식으로 암호화된 섬네일화상이 저장된다. 섬네일파일(thumbnail.tdt2)에는, 암호화되어 있지 않은 섬네일화상이 저장된다. 예를 들면, 비디오 카메라로 이용자가 촬영한 클립에 대응하는 섬네일화상은, 복사가 자유롭고 암호화할 필요가 없다고 생각되기 때문에, 이 섬네일파일(thumbnail.tdt2)에 저장된다.

[0094] 도 5에 도시된 각 파일 가운데, 이 발명에 관계가 깊은 것에 대해서, 보다 상세하게 설명한다. 먼저, 디렉토리 "BDMV"의 직하에 놓여지는 파일 "index.bdmv"에 대해 설명한다. 도 6은, 이 파일 "index.bdmv"의 일례의 구조를 나타내는 문맥을 나타낸다. 여기에서는, 문맥(syntax)을 컴퓨터 장치 등의 프로그램의 기술 언어로서 이용되는 C언어의 기술법에 근거하여 나타내고 있다. 이것은, 다른 문맥을 나타내는 도면에 있어서도, 동일하다.

[0095] 도 6에 있어서, 필드 TypeIndicator는, 32비트의 데이터 길이를 가지며, 이 파일이 인덱스 테이블인 것을 나타낸다. 필드 TypeIndicator2는, 32비트의 데이터 길이를 가지며, 이 파일 "index.bdmv"의 버젼(version)을 나타낸다. 필드 IndexesStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥내에 있는 블록 blkIndexes()의 개시 주소를 나타낸다.

[0096] 필드 ExtensionDataStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥내에 있는 블록

blkExtensionData()의 개시 주소를 나타낸다. 블록 blkExtensionData()는, 소정의 확장 데이터를 저장 가능하게 하기 위한 블록이다.필드 ExtensionDataStartAddress는, 이 파일"index.bdmv"의 최초의 바이트로부터의 상대 바이트 수로서, 블록 blkExtensionData()의 개시 주소를 나타낸다. 상대 바이트 수는, "0"으로부터 개시된다. 만약, 이 필드 ExtensionDataStartAddress의 값이 "0"이면, 이 파일 "index.bdmv"내에, 블록 blkExtensionData()가 존재하지 않는 것을 나타낸다.

[0097] 필드 ExtensionDataStartAddress에 이어서, 데이터 길이가 192바이트의 영역 reserved가 배치된다. 한편, 영역 reserved는, 바이트 얼라이먼트(alignment)나, 장래적인 필드의 추가 등을 위한 영역이다. 이것은, 이하의 설명에 있어서도 동일하다. 블록 blkAppInfoBDMV()는, 콘텐츠 제작자가 임의의 정보를 기술할 수 있는 블록이며, 재생기의 동작 등에는 영향을 주지 않는다.

[0098] 블록 blkIndexes()는, 이 파일 "index.bdmv"의 실질적인 내용이며, 이 파일"index.bdmv"에 기술된 내용에 의해, 디스크를 재생기에 장전했을 때에 재생되는 퍼스트 플레이백이나, 톱 메뉴로부터 호출되는 타이틀(무비 오브젝트)이 지정된다.인덱스 테이블에 의해 호출된 무비 오브젝트 등에 기술된 명령에 근거하여, 후술 하는 플레이 리스트 파일이 읽혀진다.

[0099] 도 7은, 블록 blkIndexes()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length 직후부터 이 블록 blkIndexes()의 마지막까지의 데이터 길이를 나타낸다. 이어서, 블록 FirstPlaybackTitle() 및 블록 MenuTitle()이 배치된다.

[0100] 블록 FirstPlaybackTitle()는, 퍼스트 플레이백으로 이용되는 오브젝트에 관한 정보가 기술된다. 블록 FirstPlaybackTitle()는, 1비트의 데이터 길이를 가지는 영역 reserved에 이어 고정치 "1"이 기술된다. 게다가 31비트의 데이터 길이를 가지는 영역 reserved를 통해, 고정치 "1"이 기술된다. 그리고, 14비트의 데이터 길이를 가지는 영역 reserved를 통해, 16비트의 데이터 길이를 가지는 필드 FirstPlaybackTitleMobjIDRef가 배치된다. 이 필드 FirstPlaybackTitleMobjIDRef에 의해, 퍼스트 플레이백 타이틀로 이용되는 무비 오브젝트의 ID를 나타낸다.

[0101] 무비 오브젝트의 ID는, 예를 들면, 도 8 및 도 9를 이용하여 후술하는 무비 오브젝트의 문맥에 근거하여, 무비 오브젝트의 for 루프문에 대해 루프 변수로서 이용되는 값(mobj_id)으로 나타낸다. 이 예에서는, 필드 FirstPlaybackTitleMobjIDRef는, 참조하는 무비 오브젝트에 대응하는 값(mobj_id)이 저장된다.

[0102] 한편, 블록 blkIndexes()에 있어서의 블록 FirstPlaybackTitle()내의 필드 FirstPlaybackTitleMobjIDRef는, 톱 메뉴의 무비 오브젝트를 가리키고 있어도 좋고, 타이틀을 가리키고 있어도 좋다.

[0103] 블록 MenuTitle()에는, 톱 메뉴로 이용되는 오브젝트에 관한 정보가 기술된다. 블록 MenuTitle()에는, 1비트의 데이터 길이를 가지는 영역 reserved에 이어 고정치 "1"이 기술된다. 게다가 31비트의 데이터 길이를 가지는 영역 reserved를 통해 고정치"1"이 기술된다. 그리고, 14비트의 데이터 길이를 가지는 영역 reserved를 통해, 16비트의 데이터 길이를 가지는 필드 MenuTitleMobjIDRef가 배치된다. 필드 MenuTitleMobjIDRef는, 메뉴 타이틀로 이용되는 무비 오브젝트의 ID를 나타낸다.

[0104] 블록 MenuTitle()의 다음의 필드 NumberOfTitles는, 16비트의 데이터 길이를 가지며, 이용자가 선택, 재생 가능한 타이틀의 수를 나타낸다. 다음의 for 루프문에 따라, 이 필드 NumberOfTitles에 나타나는 회수만큼, 값 title_id를 인수로서 블록 MovieTitle[title_id]()이 기술된다. 블록 MovieTitle[title_id]()는, 타이틀 마다의 정보가 기술된다. 값 title_id는, "0"으로부터 필드 NumberOfTitles로 나타나는 값까지의 수치이며, 타이틀을 식별한다.

[0105] 블록 MovieTitle[title_id]()에 있어서, 1비트의 데이터 길이를 가지는 영역 reserved를 통해 고정치 "1"이 기술되며, 게다가, 46비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 MovieTitleMobjIDRef가 기술된다. 필드 MovieTitleMobjIDRef는, 16비트의 데이터 길이를 가지며, 이 타이틀에서 이용되는 무비 오브젝트의 ID를 나타낸다. 필드 MovieTitleMobjIDRef의 뒤에는, 32비트의 데이터 길이를 가지는 영역 reserved가 배치된다.

[0106] 도 8은, 디렉토리 "BDMV"의 직하에 놓여지는 파일 "MovieObject.bdmv"의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 TypeIndicator는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 파일"MovieObject.bdmv"인 것을 나타낸다. 필드 TypeIndicator에는, ISO(International Organization for Standarization) 646에 규정된 부호화 방식으로 부호화된 4 문자로 구성되는 문자열이 기술된다. 이 도 8의 예에서는, 필드 TypeIndicator에 ISO646에 규정의 방식으로 부호화된 4문자의 문자열 "MOBJ"가 기술되며, 이 파일이 파일 "MovieObject.bdmv"인

것이 표시된다.

[0107] 필드 TypeIndicator2는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일"MovieObject.bdmv"의 버전 번호를 나타낸다. 이 파일 "MovieObject.bdmv"에서는, 필드 TypeIndicator2는, ISO646에 규정된 부호하 방식으로 부호화된 4문자의 문자열 "0100"이 아니면 안된다.

[0108] 필드 ExtensionDataStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥내에 있는 블록 blkExtensionData()의 개시 주소를 나타낸다. 필드 ExtensionDataStartAddress는, 이 파일 "MovieObject.bdmv"의 최초의 바이트로부터의 상대 바이트 수이며, 블록 blkExtensionData()의 개시 주소를 나타낸다. 상대 바이트 수는, "0"으로부터 개시된다. 만약, 이 필드 ExtensionDataStartAddress의 값이 "0"이면, 이 파일 "MovieObject.bdmv"내에, 블록 blkExtensionData()가 존재하지 않는 것을 나타낸다.

[0109] 한편, 이 도 8에 도시한 문맥내의 필드 padding_word는, 16비트의 데이터 길이를 가지며, 이 파일 "MovieObject.bdmv"의 문맥에 따라 for 루프문에 값 N1 또는 값 N2로 나타나는 회수만큼 삽입된다. 값 N1 또는 값 N2는,"0"또는 임의의 양의 정수이다. 또, 필드 padding_word는, 임의의 값을 이용할 수 있다.

[0110] 필드 ExtensionDataStartAddress에 이어 데이터 길이가 224비트의 영역 reserved가 배치되며, 그 다음에, 이 파일 "MovieObject.bdmv"의 본체인 블록 blkMovieObjects()가 저장된다.

[0111] 도 9는, 블록 blkMovieObjects()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 이 블록 blkMovieObjects()의 마지막까지의 데이터 길이를 나타낸다. 32비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 NumberOfMobjs이 배치된다. 필드 NumberOfMobjs는, 직후의 for 루프문에 따라 저장되는 무비 오브젝트의 수를 나타낸다. for 루프문의 루프 변수로서 이용되는 값 mobj_id에서, 무비 오브젝트가 일의적으로 특정된다. 값 mobj_id는,"0"부터 시작되는 값이며, 무비 오브젝트는, for 루프문중에 기술되는 순서에 의해 정의된다.

[0112] for 루프 문중의 블록 TerminalInfo()에는, 고정치 "1"이 기술되며, 다음에 15비트의 데이터 길이를 가지는 영역 reserved가 배치된다. 그 다음에, 16비트의 데이터 길이를 가지는 필드 NumberOfNavigationCommands[mobj_id]가 배치된다. 이 필드 NumberOfNavigationCommands[mobj_id]는, 값 mobj_id에 의해서 지시되는 무비 오브젝트 MovieObject[mobj_id]()에 포함되는 네비게이션 명령(NavigationCommand)의 수를 나타낸다.

[0113] 다음의, 값command_id를 루프 변수로 하는 for 루프문에 의해, 필드 NumberOfNavigationCommands[mobj_id]에 나타나는 수만큼, 네비게이션 명령이 기술된다. 즉, 이 for 루프문중에 배치되는 필드 NavigationCommand[mobj_id][command_id]는, 값 mobj_id에 의해서 지시되는 블록 MovieObject[mobj_id]()에 포함되는, 값 command_id로 나타내는 순번의 네비게이션 명령 NavigationCommand를 저장한다. 값 command_id는, 0으로부터 시작되는 값으로, 네비게이션 명령 NavigationCommand는, 이 for 루프문중에 기술되는 순서로 정의된다.

[0114] 도 10은, 플레이 리스트 파일 "xxxxx.mpls"의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 TypeIndicator는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 플레이 리스트 파일인 것을 나타낸다. 필드 TypeIndicator2는, 32비트(4바이트)의 데이터 길이를 가지며, 이 플레이 리스트 파일의 버젼을 나타낸다. 필드 PlayListStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥중의 블록 blkPlayList()의 개시 주소를 나타낸다.

[0115] 필드 PlayListMarkStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥중의 블록 blkPlayListMark()의 개시 주소를 나타낸다. 필드 ExtensionDataStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥중의 블록 blkExtensionData()의 개시 주소를 나타낸다. 필드 ExtensionDataStartAddress는, 블록 blkExtensionData()의 개시 주소를, 파일 "xxxxx.mpls"의 최초의 바이트로부터의 상대 바이트 수를 나타낸 값이다. 상대 바이트수는,"0"부터 개시된다. 만약, 이 필드 ExtensionDataStartAddress의 값이 "0"이면, 이 파일"xxxxx.mpls"내에, 블록 blkExtensionData()가 존재하지 않는 것을 나타낸다.

[0116] 160비트의 데이터 길이를 가지는 영역 reserved를 통해 블록 blkAppInfoPlayList()가 배치된다. 블록 blkAppInfoPlayList()에는, 다음의 블록 blkPlayList()에 기술되는 플레이 리스트의 타입, 재생 제한 등의 정보가 기술된다. 블록 blkPlayList()는, 플레이 리스트가 기술된다. 블록 blkPlayListMark()는, 챕터 등에서 점프되는 포인트가 기술된다. 블록 blkExtensionData()는, 소정의 확장 데이터를 저장 가능하게 하기 위한 블

록이다.

[0117] 도 11은, 블록 blkPlayList()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkPlayList()의 끝까지의 데이터 길이를 나타낸다. 필드 Length에 이어 16비트의 데이터 길이를 가지는 영역 reserved가 배치되며, 다음에 필드 NumberOfPlayItems가 배치된다. 필드 NumberOfPlayItems는, 16비트의 데이터 길이를 가지며, 이 블록 blkPlayList()에 포함되는 플레이 아이템의 수를 나타낸다. 필드 NumberOfSubPath는, 이 블록 blkPlayList()에 포함되는 서브 패스의 수를 나타낸다.

[0118] 다음의 for 루프문에 따라, 필드 NumberOfPlayItems로 나타내는 수만큼, 플레이 아이템이 기술되는 블록 blkPlayItem()이 기술된다. for 루프문에 근거하는 카운트 수가 블록 blkPlayItem()의 식별자 PlayItem_id가 된다. 게다가 다음의 for 루프문에 따라, 필드 NumberOfSubPath로 나타나는 수만큼, 블록 blkSubPath()가 기술된다. for 루프문에 근거하는 카운트 수가 블록 blkSubPath()의 식별자 SubPath_id가 된다.

[0119] 한편, 서브 패스(Subpath)는, 주로 재생되는 플레이 아이템에 대응하는 메인 패스에 대해서, 서브 플레이 아이템에 대응하여 가질 수 있다. 서브 패스는, 예를 들면, 애프터 레코딩(after recording)용의 오디오 데이터의 지정이나, 2매의 영상을 합성할 때에, 플레이 아이템으로 지정되는 클립과 동기하여 재생하는 부영상을 지정하는 목적으로 이용된다.

[0120] 도 12는, 블록 blkPlayItem()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 16비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkPlayItem()의 끝까지의 데이터 길이를 나타낸다.

[0121] 필드 ClipInformationFileName는, 40비트(5바이트)의 데이터 길이를 가지며, 이 블록 blkPlayItem()이 참조하는 클립 정보 파일의 파일명이 표시된다. 이 플레이 아이템에 있어서, 필드 ClipInformationFileName로 나타나는 파일명의 클립 정보 파일이 읽혀진다. 필드 ClipCodecIdentifier는, 32비트(4바이트)의 데이터 길이를 가지며, 이 블록 blkPlayItem()에 의한 플레이 아이템에 대해 이용되는 클립 AV스트림의 코덱 방식을 나타낸다.

[0122] 12비트의 데이터 길이를 가지는 영역 reserved를 통해, 필드 ConnectionCondition가 배치된다. 필드 ConnectionCondition는, 4비트의 데이터 길이를 가지며, 클립간의 접속 상태에 관한 정보를 나타낸다. 기록 용도의 기록 매체에 대해서는, 필드 ConnectionCondition의 값으로 "1", "5" 또는 "6"이 이용된다. 필드 ConnectionCondition의 값이 "1"이면, 그 플레이 아이템으로부터 참조되어 있는 클립과 앞의 플레이 아이템으로부터 참조되어 있는 클립이 심리스 접속하지 않는 것을 나타내며, 필드 ConnectionCondition의 값이 "5" 또는 "6"이면, 그 플레이 아이템으로부터 참조되어 있는 클립과 앞의 플레이 아이템으로부터 참조되고 있는 클립이 심리스 접속하는 것을 나타낸다. 한편 심리스 접속이란, 클립과 다음의 클립이 프레임 타이밍에 연속적으로 재생되도록, 클립간의 재생 제어를 실시하는 것을 말한다.

[0123] 필드 ConnectionCondition의 값이 "5"가 되면, 해당 플레이 아이템이 참조하는 클립에 있어서, 오디오 데이터의 기록 길이가 비디오 데이터의 기록 길이에 대해서 길게 된다(도 13a참조). 이에 의해, 클립과 클립을 접속할 때에, 오디오 데이터의 페이드 아웃(fade out) 처리가 가능하게 된다. 예를 들면, 이용자에 의한 기록 정지 조작에 의해 클립이 클로우즈(close) 되는 경우에, 필드 ConnectionCondition의 값이 "5"로 된다. 이하, 이 필드 ConnectionCondition의 값이 "5"로 표시되는 클립의 접속 방법을, 제 1의 심리스 접속이라고 부른다.

[0124] 필드 ConnectionCondition의 값이 "6"이라면, 해당 플레이 아이템이 참조하는 클립에 있어서, 오디오 데이터의 기록 길이가 비디오 데이터의 기록 길이에 대해서 동일하게 된다(도 13b 참조). 이에 의해, 클립과 클립과의 접속을 심리스하게 실시하는 것이 가능하게 된다. 예를 들면, 이용자 조작에 대응하는 기록 정지 이외의 이유, 예를 들면 시스템 요인에 근거하여 클립이 클로우즈 되는 경우에, 필드 ConnectionCondition의 값이 "6"으로 된다. 이하, 이 필드 ConnectionCondition의 값이 "6"으로 나타나는 클립의 접속 방법을, 제 2의 심리스 접속이라고 부른다.

[0125] 필드 RefToSTCID는, 8비트의 데이터 길이를 가지며, 시스템 타임 베이스(STC)의 불연속점에 관한 정보를 나타낸다. 필드 INTime 및 필드 OUTTime는, 각각 32비트의 데이터 길이를 가지며, 메인 클립 AV스트림의 재생 범위를 나타낸다. 필드 INTime이 개시점(IN점)을 나타내며, 필드 OUTTime가 종료점(OUT점)을 나타낸다.

[0126] 블록 blkUOMaskTable()은, 이용자 입력의 접수 제한이 설정되는 테이블이다. 1비트의 데이터 길이를 가지는 플래그 PlayItemRandomAccessFlag는, 이 블록 blkPlayItem()에 의한 플레이 아이템에 대해서 랜덤 억세스를 허가하는지 아닌지를 규정한다. 이어서, 7비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 StillMode가 배치된다. 필드 StillMode는, 8비트의 데이터 길이를 가지며, 블록 blkPlayItem()에 의한 플레이 아이템에

대해서, 마지막으로 표시한 영상을 정지화면으로서 표시시키는지 아닌지를 나타낸다. 필드 StillMode의 값이 "0x01"(2진수)이면, if문에 근거하여, 16비트의 데이터 길이를 가지는 필드 StillTime에 의해 정지 시간이 나타난다. 필드 StillMode의 값이 "0x01"이외이면, 해당 16비트의 데이터 길이를 가지는 영역이 영역 reserved로 여겨진다.

[0127] 한편, 수치의 기술에 있어서 "0x"는, 그 수치가 16진 표기로 되어 있는 것을 나타낸다. 이것은, 이하의 동일한 표기에 대해 공통이다.

[0128] 블록 blkSTNTable()에서는, 이 블록 blkPlayItem()에 의한 플레이 아이템이 관리하고 있는 클립 AV스트림의 속성, PID 번호, 기록 매체상에서의 기록 위치 등이 관리된다.

[0129] 도 14는, 블록 blkPlayListMark()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkPlayListMark()의 끝까지의 데이터 길이를 나타낸다.

[0130] 필드 NumberOfPlayListMarks는, 16비트의 데이터 길이를 가지며, 이 블록 blkPlayListMark()에 포함되는 플레이 리스트 마크의 수를 나타낸다. 다음의 for 루프문에 따라, 필드 NumberOfPlayListMarks로 나타나는 수만큼 플레이 리스트 마크의 정보가 기술된다.

[0131] for 루프문내에 있어서, 8비트의 데이터 길이를 가지는 영역 reserve에 이어 필드 MarkType가 배치된다. 필드 MarkType는, 8비트의 데이터 길이를 가지며, 마크의 타입을 나타낸다. 플레이 리스트 마크에는, 엔트리 마크(Entry Mark) 및 링크 포인트(Link Point)의 두 개의 타입이 정의되어 있으며, 이 필드 MarkType에 의해, 어느 쪽의 타입이 되는 지가 표시된다. 챕터를 정의하기 위해서는, 엔트리 마크를 이용한다. 링크 포인트는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다. 상술한 필드 NumberOfPlayListMarks는, 엔트리 마크 및 링크 포인트를 합계한 값을 나타낸다.

[0132] 필드 RefToPlayItemID는, 16비트의 데이터 길이를 가지며, 마크가 일치하는 플레이 아이템을 참조하는 식별 정보 PlayItem_id가 기술된다. 필드 MarkTimeStamp는, 32비트의 데이터 길이를 가지며, 마크가 맞는 포인트를 나타내는 타임 스탬프가 기술된다. 필드 EntryESPID는, 16비트의 데이터 길이를 가지며, 마크에 의해서 지시되는 기초(elementary) 스트림을 포함하고 있는 TS패킷의 PID의 값을 나타낸다. 필드 Duration은, 45 kHz의 클락을 단위로 하는 계측에 의한, 32비트의 데이터 길이를 가지는 부호 없는 정수이다. 이 필드 Duration에 저장되는 값이 "0"이면, 이 필드 Duration은, 의미를 이루지 않는다.

[0133] 도 15는, 클립 정보 파일의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 TypeIndicator는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 클립 정보 파일인 것을 나타낸다.

[0134] 필드 TypeIndicator2는, 32비트(4바이트)의 데이터 길이를 가지며, 이 클립 정보 파일의 버젼을 나타낸다. 이 클립 정보 파일은, 블록 blkClipInfo(), 블록 blkSequenceInfo(), 블록 blkProgramInfo(), 블록 blkCPI(), 블록 blkClipMark() 및 블록 blkExtensionData()를 가지며, 각각 32비트의 데이터 길이를 가지는 필드 SequenceInfoStartAddress, 필드 ProgramInfoStartAddress, 필드 CPIStartAddress, 필드 ClipMarkStartAddress 및 필드 ExtensionDataStartAddress는, 각각 대응하는 블록의 개시 주소를 나타낸다.

[0135] 필드 ExtensionDataStartAddress는, 이 클립 정보 파일의 최초의 바이트로부터의 상대 바이트수이며, 블록 blkExtensionData()의 개시 주소를 나타낸다. 상대 바이트수는,"0"으로부터 개시된다. 만약, 이 필드 ExtensionDataStartAddress의 값이 "0"이면, 이 파일 "index.bdmv"내에, 블록 blkExtensionData()가 존재하지 않는 것을 가리킨다.

[0136] 블록 blkClipInfo()는, 이러한 개시 주소를 나타내는 필드에 이어서, 96비트의 데이터 길이를 가지는 영역 reserved의 다음으로부터 개시된다. 블록 blkClipInfo()에는, 이 클립 정보 파일이 관리하는 클립 AV스트림에 관한 정보가 기술된다. 블록 blkSequenceInfo()에는, STC나 ATC(어라이벌 타임 베이스)가 연속하고 있는 순서를 정리하여 관리하는 정보가 기술된다. 블록 blkProgramInfo()에는, 이 클립 정보 파일에 관리되는 클립 AV스트림의 부호하 방식, 클립 AV스트림중의 비디오 데이터의 애스펙트비 등의 정보가 기술된다. 블록 blkCPI()에는, 랜덤 억세스 개시점 등의, AV스트림중의 특징적인 위치를 나타내는 특징점정보 CPI에 관한 정보가 저장된다.

[0137] 또한, 블록 blkClipMark()에는, 챕터 위치 등의, 클립에 첨부된 선두 부분의 인덱스점(점프 포인트)이 기술된다. 블록 blkExtensionData()는, 확장 데이터를 저장할 수 있는 영역이다. 한편, 클립 정보 파일내의 블록 blkClipMark() 및 블록 blkSequenceInfo()는, 이 발명과의 관련성이 별로 없기 때문에, 상세한 설명을 생

략한다.

[0138] 도 16은, 블록 blkClipInfo()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkClipInfo()의 끝까지의 데이터 길이를 나타낸다. 16비트의 데이터 길이를 가지는 영역 reserved를 통해, 필드 ClipStreamType가 배치된다.

[0139] 필드 ClipStreamType는, 8비트의 데이터 길이를 가지며, 클립 AV스트림의 종별을 나타낸다. 이 필드 ClipStreamType의 값은, 예를 들면 "1"로 고정된다. 필드 ApplicationType는, 8비트의 데이터 길이를 가지며, 클립 AV스트림(확장자(extension)가 [m2ts]의 파일)이 어떠한 다중화에 의해서 만들어지고 있는지를 나타낸다. 필드 ApplicationType의 값이 "1"이면, 대응하는 클립 AV스트림은, 통상의 동영상이 재생된다. 이어서, 31비트의 데이터 길이를 가지는 영역 reserved가 배치된다.

[0140] 데이터 길이가 1비트의 플래그 IsCC5는, 플레이 리스트에 있어서의 블록 blkPlayItem()에 의해서, 대응하는 클립과 다음의 클립과의 접속을, 상술한 제 1의 심리스 접속, 즉 필드 ConnectionCondition의 값이 "5"로 나타나는 방법으로 실시하는지 아닌지를 나타낸다. 플래그 IsCC5의 값이 "1"(이진 값)이면, 클립간의 접속이 제 1의 심리스 접속에 의해 이루어지고 있는 것을 나타낸다.

[0141] 필드 TSRecordingRate는, 클립 AV스트림 파일의 기록 레이트를 바이트/초로 표시한 것이다. 필드 NumberOfSourcePackets는, 클립 AV스트림에 포함되는 소스 패킷수를 나타낸다. 1024비트의 데이터 길이를 가지는 영역 reserved를 통해 블록 TSTypeInfoBlock()이 배치된다. 블록 TSTypeInfoBlock()는, 클립 AV스트림이 저장되는 패킷의 타입을 나타내는 정보가 저장된다. 이 블록 TSTypeInfoBlock()는, 이 발명과의 관련성이 별로 없기 때문에, 상세한 설명을 생략한다.

[0142] 다음의 if문 이하의 정보는, 상술의 플래그 IsCC5의 값이 "1"인 경우에 기술된다. if문의 다음의 8비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 FollowingClipStreamType가 배치되는 필드 FollowingClipStreamType는, 8비트의 데이터 길이를 가지며, 이 클립 정보 파일에 대응하는 클립의 다음의 클립의 타입이 기술된다. 32비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 FollowingClipInformationFileName가 배치된다.

[0143] 필드 FollowingClipInformationFileName는, 40비트(5바이트)의 데이터 길이를 가지며, 이 클립 정보 파일에 대응하는 클립의 다음의 클립에 대응하는 클립 정보 파일의 파일명이 기술된다. 다음의 필드 ClipCodecIdentifier는, 32비트(4바이트)의 데이터 길이를 가지며, 상기 다음의 클립의 부호화 방식을 나타낸다. 이 예에서는, 필드 ClipCodecIdentifier는, ISO646에 규정의 방식으로 부호화된 4문자의 문자열치 "M2TS"로 고정된다. 다음에 8비트의 데이터 길이를 가지는 영역 reserved가 배치된다.

[0144] 도 17은, 블록 blkProgramInfo()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkProgramInfo()의 끝까지의 데이터 길이를 나타낸다. 15비트의 데이터 길이를 가지는 영역 reserved를 통해, 데이터 길이가 1비트, 고정치 "1"이 기술된다.

[0145] 필드 SPNProgramSequenceStart에는, 32비트의 데이터 길이를 가지며, 대응하는 클립 AV스트림 파일에 있어서, 프로그램 순서가 개시되는 소스 패킷의 번호가 기술된다. 필드 ProgramMapPID는, 16비트의 데이터 길이를 가지며, 프로그램 순서에 적용 가능한 프로그램 맵 섹션을 포함되어 있는 TS패킷의 PID의 값을 나타낸다. 필드 NumberOfStreamsInPS는, 8비트의 데이터 길이를 가지며, 프로그램 순서에 정의되는 기초 스트림의 수를 나타낸다. 필드 NumberOfStreamsInPS에 이어서, 8비트의 데이터 길이를 가지는 영역 reserved가 배치된다.

[0146] 다음의 for 루프문에 따라, 값[stream_index]을 루프 변수로서, 필드 NumberOfStreamsInPS로 표시되는 수만큼, 필드 StreamPID[stream_index] 및 블록 blkStreamCodingInfo(stream_index)의 그룹이 저장된다. 필드 StreamPID[stream_index]는, 프로그램 순서에 의해서 참조된 PMT(Program Map Table)에 기술된 기초 스트림에 대응하는 PID의 값을 나타낸다. 다음의 블록 blkStreamCodingInfo(stream_index)에는, 대응하는 필드 StreamPID[stream_index]로 표시되는 기초 스트림의 부호화 방식에 관한 정보가 기술된다.

[0147] 도 18은, 블록 blkStreamCodingInfo(stream_index)의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 8비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkStreamCodingInfo(stream_index)의 마지막까지의 데이터 길이를 나타낸다.

[0148] 필드 Length의 다음에, 8비트의 데이터 길이를 가지는 필드 StreamCodingType가 배치된다. 필드 StreamCodingType에서는, 값[stream_index]로 나타나는 기초 스트림의 부호화 타입이 나타난다. 일례로서 필

드 StreamCodingType의 값으로 값 "0x1B", 값 "0x80", 값 "0x81", 값 "0x90" 및 값"0x91"가 정의되며, 해당 스트림의 부호화 타입은, 값 "0x1B"로 비디오 스트림, 값"0x80" 또는 값 "0x81"로 오디오 스트림, 값 "0x90"로 OB 스트림, 값 "0x91"로 MB스트림이라는 것이 각각 표시된다. 다음의 if문에 따라, 이 필드 StreamCodingType의 값에 대응하는 기재가 서술된다.

[0149] 필드 StreamCodingType의 값이 예를 들면 "0x1B"이며, 값[stream_index]으로 나타나는 기초 스트림이 비디오 스트림이라는 것이 나타나면, if문에 따라 필드 VideoFormat, 필드 FrameRate 및 필드 AspectRatio가 기술되며, 2비트의 데이터 길이를 가지는 영역 reserved를 통해 추가로 플래그 CCFlag가 기술된다. 플래그 CCFlag의 뒤에는, 17비트의 데이터 길이를 가지는 영역 reserved와 128비트의 데이터 길이를 가지는 영역 reserved가 배치된다.

[0150] 필드 VideoFormat는, 4비트의 데이터 길이를 가지며, 값[stream_index]로 나타나는 비디오 데이터의 포맷을 나타낸다. 도 19는, 필드 VideoFormat로 나타나는 비디오 데이터의 일례의 포맷을 일람(一覽)으로 나타내고 있다. 도 19에 예시되어 있는 바와 같이, 비디오 데이터의 포맷은, 4비트로 표현 가능한 값 "0"~ 값 "15"로 식별되며, 값 "0" 및 값 "8"~ 값 "15"는, 예약으로 되어 있다. 값 "1"~ 값 "7"에 서는, 각각 비디오 데이터의 포맷의 480i, 576i, 480p, 1080i, 720p, 1080p 및 576 p를 나타낸다.

[0151] 한편, 이 비디오 포맷의 표기에 있어서, 말미의 「i」 및 「p」는, 각각 인터레이스 주사 및 프로그래시브 주사를 나타낸다. 또, 숫자는 라인수를 나타낸다. 이러한 비디오 포맷은, ITU(International Telecommunication Union)-RBT.601-4(480i 및 576i), ITU-RBT.1358(576p), SMPTE(Society of Motion Picture and Television Engineers) 293 M(480p), SMPTE274M(1080i 및 1080p) 및 SMPTE296M(720 p)에 의해 표준화된 포맷이다.

[0152] 블록 blkStreamCodingInfo(stream_index)에 있어서, 필드 FrameRate는, 4비트의 데이터 길이를 가지며, 값[stream_index]으로 나타나는 비디오 데이터의 프레임 속도를 나타낸다. 도 20은, 필드 FrameRate로 나타나는 일례의 프레임 속도를 일람으로 가리킨다. 도 20에 예시되어 있는 바와 같이, 비디오 데이터의 프레임 속도는, 4비트로 표현 가능한 값 "0"~"15"로 식별되며, 값 "0", 값 "5" 및 값"8" ~ 값 "15"는, 예약으로 되어 있다. 값 "1" ~ 값 "4"로, 각각 프레임 속도(필드 주파수)가 (24000 / 1001)Hz, 즉 거의 23.97Hz, 24Hz, 25Hz 및 (30000 / 1001) Hz 즉, 거의 29.97 Hz를 나타낸다. 또, 값 6 및 값 7로, 각각 프레임 속도가 50 Hz 및(60000 / 1001) Hz 즉 거의 59.94 Hz를 나타낸다.

[0153] 블록 blkStreamCodingInfo(stream_index)에 있어서, 필드 AspectRatio는, 4비트의 데이터 길이를 가지며, 값[stream_index]로 나타나는 비디오 데이터의 애스펙트비를 나타낸다. 도 21은, 필드 AspectRatio로 나타나는 일례의 프레임 속도를 일람으로 가리킨다. 도 21에 예시되어 있는 바와 같이, 비디오 데이터의 애스펙트비는, 4비트로 표현 가능한 값 "0" ~ 값 "15"로 식별되며, 값 "0" 및 값"1", 및, 값 "4" ~ 값 "15"는, 예약으로 되어 있다. 값 "2"로 애스펙트비가 4 : 3을 나타낸다. 값 "3"으로 애스펙트비가 16 : 9를 나타낸다.

[0154] 블록 blkStreamCodingInfo(stream_index)에 있어서, 필드 StreamCodingType의 값이 예를 들면 "0x80", "0x81", "0x90"또는 "0x91"의 경우에도, if문에 있어서의 「else if」의 기술에 따라, 각각의 값이 가리키는 부호화 타입에 대응하는 기재가 이루어진다. 한편, 비디오 스트림 이외의 부호화 타입의 스트림에 관한 기재는, 이 발명과 관련성이 별로 없기 때문에, 상세한 설명을 생략한다.

[0155] 도 22는, 클립 정보 파일에 있어서의 블록 blkCPI()의 일례의 구조를 나타내는 문맥을 나타낸다. MPEG 스트림과 같은, 프레임간 압축을 실시하고 있는 부호화 스트림에 대해서는, 디코드 개시 가능한 위치는, GOP(Group Of Picture)의 선두 등 일부의 위치로 한정되어 있는 것이 많다. CPI(Characteristic Point Information)란, 그 디코드 가능한 개시점의 위치의 정보를 모은 데이터 베이스로, 재생 시각과 파일내 주소가 대응되어 있는 테이블이 된다. 즉, CPI는, 디코드 단위의 선두 위치를 나타내는 정보가 테이블화 되어 있다.

[0156] 이와 같이 데이터 베이스를 정하는 것으로, 예를 들면, 임의의 시각부터 재생하고 싶은 경우, 재생 시각을 바탕으로 CPI를 참조하는 것에 의해서 재생 위치의 파일내 주소를 알 수 있다. 이 주소는, 디코드 단위의 선두가 되므로, 재생기는, 그 곳으로부터 데이터를 읽어내 디코드하고, 신속하게 화상을 표시할 수 있다.

[0157] 한편, 이 CPI에 저장되는, 디코드 단위의 선두 위치(이 예에서는 GOP의 선두 위치)를, EP(Entry Point) 엔트리라고 칭한다.

[0158] 도 22에 있어서, 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkCPI()의

끝까지의 데이터 길이를 나타낸다. 다음의 if문에 따라, 필드 Length의 값이 0이 아니면, 12비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 CPIType가 배치된다. 필드 CPIType는, 4비트의 데이터 길이를 가지며, CPI의 종류를 나타낸다. 다음의 블록 blkEPMap()는, 대응하는 클립 AV스트림 파일에 있어서의 PTS치와 바이트 주소와의 관련성을 기재한 테이블이 저장된다.

[0159] 도 23은, 블록 blkEPMap()의 일례의 구조를 나타내는 문맥을 나타낸다. 8비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 NumberOfStreamPIDEntries가 배치된다. 필드 NumberOfStreamPIDEntries는, 8비트의 데이터 길이를 가지며, 블록 blkEPMap()에 있어서의 블록 blkEPMapForOneStreamPID의 엔트리수를 나타낸다.

[0160] for 루프문에 따라, 값 [k]를 루프 변수로서 필드 NumberOfStreamPIDEntries에 나타나는 수만큼, 엔트리 포인트에 관한 정보가 기술된다. for 루프문내에서, 필드 StreamPID[k]는, 16비트의 데이터 길이를 가지며, 블록 blkEPMap()안에서 [k]번째에 엔트리 되는 블록 blkEPMapForOneStreamPID(이하,[k]번째의 블록 blkEPMapForOneStreamPID라고 기술한다)에 의해서 참조되는 기초 스트림을 전송하는 트랜스포트 패킷의 PID의 값을 나타낸다.

[0161] 10비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 EPStreamType[k]가 배치된다. 필드 EPStreamType[k]는, 4비트의 데이터 길이를 가지며,[k]번째의 블록 blkEPMapForOneStreamPID에 의해서 참조되는 기초 스트림의 타입을 나타낸다.필드 NumberOfEPCoarseEntries[k]는, 16비트의 데이터 길이를 가지며,[k]번째의 블록 blkEPMapForOneStreamPID안에 있는 대략적인 검색용의 서브 테이블(EP coarse table)의 엔트리수를 나타낸다. 필드 NumberOfEPFineEntries[k]는, 18비트의 데이터 길이를 가지며, [k]번째의 블록 blkEPMapForOneStreamPID안에 있는 정밀한 검색용의 서브 테이블(EP fine table)의 엔트리수를 나타낸다. 필드 EPMapForOneStreamPIDStartAddress[k]는, 32비트의 데이터 길이를 가지며, 블록 blkEPMap()안에서 [k]번째의 블록 blkEPMapForOneStreamPID가 시작되는 상대 바이트 위치를 나타낸다. 이 값은, 블록 blkEPMap()의 제 1바이트로부터의 바이트수로 나타난다.

[0162] 상술의 for 루프문에 의한 기재가 되어진 후에, 16비트의 정수배의 데이터 길이를 가지는 패딩 워드를 사이에 두고 기술되는 for 루프문에 따라, 값 [k]를 루프 변수로서 필드 NumberOfStreamPIDEntries에 나타나는 수만큼, 블록 blkEPMapForOneStreamPID(EPStreamType[k], NumberOfEPCoarseEntries[k], NumberOfEPFineEntries[k])가 저장된다. 즉, 인수 NumberOfEPCoarseEntries[k]는, 서브 테이블(EP coarse table)에 저장되는 엔트리 PTSEPCoarse 및 엔트리 SPNEPCoarse의 수를 나타낸다. 이와 같이, 인수 NumberOfEPFineEntries[k]는, 서브 테이블(EP fine table)에 저장되는 엔트리 PTSEPFine 및 엔트리 SPNEPFine의 수를 나타낸다. 이하에서는, 인수 NumberOfEPCoarseEntries[k] 및 인수 NumberOfEPFineEntries[k]를, 각각 적절하게 엔트리수 Nc 및 엔트리수 Nf라고 부른다.

[0163] 도 24는, 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)의 일례의 구조를 나타내는 문맥을 나타낸다. 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)의 시멘틱스(semantics)를 설명하기 위해서, 먼저, 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)에 저장되는 데이터의 기초가 되는 엔트리인, 엔트리 PTSEPStart 및 엔트리 SPNEPStart의 의미에 대해 설명한다.

[0164] 엔트리 PTSEPStart와 엔트리 PTSEPStart와 관련되는 엔트리 SPNEPStart는, 각각 AV스트림상의 엔트리 포인트를 가리킨다. 그리고, 엔트리 PTSEPFine와 엔트리 PTSEPFine와 관련되는 엔트리 PTSEPCoarse는, 동일한 엔트리 PTSEPStart로부터 도출된다. 또, 엔트리 SPNEPFine와 엔트리 SPNEPFine와 관련되는 엔트리 SPNEPCoarse는, 동일한 엔트리 SPNEPStart로부터 도출된다.

[0165] 도 25는, 엔트리 PTSEPCoarse 및 엔트리 PTSEPFine의 일례의 포맷에 대해 도시하고 있다. PTS 즉 엔트리 PTSEPStart는, 데이터 길이가 33비트의 값이다. MSB의 비트를 제 32비트, LSB의 비트를 제 0비트로 할 때, 이 도 25의 예에서는, 큰 단위로 검색을 실시할 때에 이용되는 엔트리 PTSEPCoarse는, 엔트리 PTSEPStart의 제 32비트로부터 제 19비트까지의 14비트가 이용된다. 엔트리 PTSEPCoarse에 의해, 해상도가 5.8초로, 26.5시간까지의 범위에서 검색이 가능하다. 또, 보다 정밀한 검색을 행하기 위한 엔트리 PTSEPFine는, 엔트리 PTSEPStart의 제 19비트로부터 제 9비트까지의 11비트가 이용된다. 엔트리 PTSEPFine에 의해, 해상도가 5.7밀리 초에서, 11.5초까지의 범위까지의 검색이 가능하다. 한편, 제 19비트는, 엔트리 PTSEPCoarse와 엔트리 PTSEPFine에 의해 공통적으로 이용된다. 또, LSB측의 제 0비트로부터 제 8비트까지의 9비트는, 이용되지 않는다.

[0166] 도 26은, 엔트리 SPNEPCoarse 및 엔트리 SPNEPFine의 일례의 포맷에 대해 도시하고 있다. 소스 패킷 번호

즉 엔트리 SPNEPStart는, 데이터 길이가 32비트인 값이다. MSB의 비트를 제 31비트, LSB의 비트를 제 0비트로 할 때, 이 도 26의 예에서는, 큰 단위로 검색을 실시할 때에 이용되는 엔트리 SPNEPCoarse는, 엔트리 SPNEPStart의 제 31비트로부터 제 0비트까지의 모든 비트가 이용된다. 또, 보다 정밀한 검색을 행하기 위한 엔트리 SPNEPFine는, 엔트리 SPNEPStart의 제 16비트로부터 제 0비트까지의 17비트가 이용된다. 엔트리 SPNEPFine에 의해, 예를 들면 거의 25 MB(Mega Byte)의 AV스트림 파일까지의 범위에서, 검색이 가능하다.

[0167] 한편, 소스 패킷 번호의 경우에서도, 엔트리 SPNEPCoarse로서 MSB측의 소정 비트 수의 값만 이용하도록 해도 괜찮다. 예를 들면, 엔트리 SPNEPCoarse로서 엔트리 SPNEPStart의 제 31비트로부터 제 16비트까지의 17비트를 이용하고, 엔트리 SPNEPFine는, 엔트리 SPNEPStart의 제 16비트로부터 제 0비트까지의 17비트를 이용한다.

[0168] 상술한 기재에 근거하여, 엔트리 PTSEPStart 및 엔트리 SPNEPStart는, 다음과 같이 정의된다.

[0169] 엔트리 PTSEPStart는, 도 25에 도시된 바와 같이, 데이터 길이가 33비트의 부호 없는 정수이며, AV스트림중에서, 랜덤 억세스가 가능한 픽쳐(예를 들면 IDR(Instantaneous Decoding Refresh) 픽쳐나 I(Intra) 픽쳐)로부터 개시하는 비디오 액세스 단위의 33 bit length의 PTS를 나타낸다.

[0170] 엔트리 SPNEPStart는, 도 26에 도시된 바와 같이, 32비트의 부호 없는 정수이며, 엔트리 PTSEPStart에 관련되는 비디오 액세스 단위의 제 1바이트째를 포함하는 소스 패킷의, AV스트림중에서의 주소를 나타낸다. 엔트리 SPNEPStart는, 소스 패킷 번호의 단위로 표시되며, AV스트림 파일중의 최초의 소스 패킷부터, 값 "0"을 초기치로서, 소스 패킷마다 1씩 증가하는 값으로 카운트된다.

[0171] 도 24를 참조하여, 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)에는, 제 1의 for 루프문에 의해 큰 단위로의 검색을 행하기 위한 서브 테이블(EP coarse table)이 기술되며, 제 2의 for 루프문에 의해 서브 테이블(EP coarse table)의 검색 결과에 근거하여 상세한 검색을 행하기 위한 서브 테이블(EP fine table)이 기술된다.

[0172] 제 1의 for 루프문의 직전에, 필드 EPFineTableStartAddress가 배치된다. 필드 EPFineTableStartAddress는, 32비트의 데이터 길이를 가지며, 최초의 제 2의 for 루프에 있어서의 필드 ReservedEPFine[EP_fine_id]의 제 1바이트째의 개시 주소를, 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)의 제 1바이트째부터의 상대 바이트 수로 표시한다. 상대 바이트 수는, 값 "0"부터 개시한다.

[0173] 제 1의 for 루프문은, 루프 변수 [i]에 의해, 서브 테이블(EP coarse table)의 엔트리수 Nc까지 반복되며, 엔트리수 Nc의 그룹 수만큼 필드 RefToEPFineID[i], 엔트리 PTSEPCoarse[i] 및 엔트리 SPNEPCoarse[i]가 저장된다.

제 1의 for 루프문에 있어서, 필드 RefToEPFineID[i]는, 18비트의 데이터 길이를 가지며, 필드 RefToEPFineID[i]에 계속 되는 필드 PTSEPCoarse[i]가 가리키는 엔트리 PTSEPCoarse에 관련되는 엔트리 PTSEPFine를 가지는, 서브 테이블(EP fine table) 내의 엔트리 번호를 나타낸다. 엔트리 PTSEPFine와 이 엔트리 PTSEPFine에 관련되는 엔트리 PTSEPCoarse란, 동일한 엔트리 PTSEPStart로부터 도출된다. 필드 RefToEPFineID[i]는, 제 2의 for 루프 문중에서 기술되는 순번에 의해 정의되는 루프 변수 [EP_fine_id]의 값에 의해 주어진다.

[0174] 제 1의 for 루프문의 뒤에, 패딩 워드를 사이에 두고 제 2의 for 루프문에 의한 기술이 이루어진다. 제 2의 for 루프문은, 루프 변수 [EP_fine_id]에 의해 서브 테이블(EP fine table)의 엔트리수 Nf까지 반복되며, 엔트리수 Nf의 그룹 수만큼, 1비트의 데이터 길이를 가지는 필드 ReservedEPFine[EP_fine_id]와 3비트의 데이터 길이를 가지는 필드 IEndPositionOffset[EP_fine_id]와 11비트의 데이터 길이를 가지는 필드 PTSEPFine[EP_fine_id]와 17비트의 데이터 길이를 가지는 필드 SPNEPFine[EP_fine_id]가 저장된다. 이들 중에서, 필드 PTSEPFine[EP_fine_id]및 필드 SPNEPFine[EP_fine_id]는, 루프 변수 [EP_fine_id]에 근거하여 서브 테이블(EP fine table)에서 참조되는 엔트리 PTSEPFine 및 엔트리 SPNEPFine의 각각이 저장된다.

[0175] 엔트리 PTSEPCoarse 및 엔트리 PTSEPFine와 엔트리 SPNEPCoarse 및 엔트리 SPNEPFine는, 다음과 같이 도출된다. 서브 테이블(EP fine table)에, 관련하는 데이터 SPNEPStart의 값의 승순에 따라 배치되어 있는 Nf개의 엔트리가 있다고 가정한다. 각각의 엔트리 PTSEPFine는, 대응하는 엔트리 PTSEPStart로부터, 다음 식(1)과 같이 도출된다.

[0176] PTSEPFine[EP_fine_id]=(PTSEPStart[EP_fine_id)>>9) / 211.....(1)

[0177] 엔트리 PTSEPCoarse와 대응하는 엔트리 PTSEPFine와의 관계는, 다음 식(2), (3)이 된다.

[0178] PTSEPCoarse[i]=(PTSEPStart[RefToEPFineID[i]]>>19) / 214.....(2)

[0179] PTSEPFine[RefToEPFineID[i]]=(PTSEPStart[RefToEPFineID[i]]>>9) / 211...(3)

[0180] 각각의 엔트리 SPNEPFine는, 대응하는 엔트리 SPNEPStart로부터, 다음 식(4)와 같이 도출된다.

[0181] SPNEPFine[EP_fine_id]=SPNEPStart[EP_fine_id] / 217.....(4)

[0182] 엔트리 SPNEPCoarse와 대응하는 엔트리 SPNEPFine와의 관계는, 다음 식(5), (6)이 된다.

[0183] SPNEPCoarse[i]=SPNEPStart[RefToEPFineID[i]].....(5)

[0184] SPNEPFine[RefToEPFineID[i]]=SPNEPStart[RefToEPFineID[i]] / 217.....(6)

[0185] 한편 상술의 식(1)~(6)에 있어서, 기호 「>>x」는, 데이터의 LSB측으로부터 x비트를 초과하는 자리수로부터 비트를 이용하는 것을 의미한다.

[0186] 다음에, 확장 데이터를 저장하기 위한 블록 blkExtensionData()에 대해 설명한다. 이 블록 blkExtensionData()는, 소정의 확장 데이터를 저장 가능하다고 정의되며, 인덱스 테이블이 저장되는 파일"index.bdmv", 플레이 리스트가 저장되는 파일"xxxxx.mpls"및 클립 정보 파일"zzzzz.clpi"의 각 파일에 기술할 수 있다.

[0187] 도 27은, 블록 blkExtensionData()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkExtensionData()의 마지막까지의 데이터 길이를 바이트 수로 표시한다. 이 필드 Length가 가리키는 데이터 길이가 "0"이 아니면, if문 이하의 기재가 이루어진다.

[0188] 필드 DataBlockStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥중의, 확장 데이터의 본체가 저장되는 블록 DataBlock()의 개시 주소를, 이 블록 blkExtensionData()의 선두 바이트로부터의 상대 바이트 수로 표시한다. 즉, 상대 바이트 수는,"0"부터 개시된다. 한편 필드 DataBlockStartAddress는, 다음에 나타내는 32비트 얼라이먼트의 조건을 채우지 않으면 안 된다. DataBlockStartAddress%4=0

[0189] 24비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 NumberOfExtDataEntries가 배치된다. 필드 NumberOfExtDataEntries는, 8비트의 데이터 길이를 가지며, 이 블록 blkExtensionData()의 블록 DataBlock()에 저장되는 확장 데이터의 엔트리수를 나타낸다. 확장 데이터의 엔트리는, 확장 데이터의 본체를 취득하기 위한 정보가 저장된다. 이 예에서는, 확장 데이터의 엔트리는, 필드 ExtDataType, 필드 ExtDataVersion, 필드 ExtDataStartAddress 및 필드 ExtDataLength로 구성되는 블록 ext_data_entry()이며, 블록 blkExtensionData()에 있어서, 제 1의 for 루프문에 따라 이 필드 NumberOfExtDataEntries에 나타나는 갯수만큼, 이 블록 ext_data_entry()가 존재한다.

[0190] 필드 ExtDataType는, 16비트의 데이터 길이를 가지며, 이 블록 blkExtensionData()에 기술되는 확장 데이터가 기록 장치용의 확장 데이터인 것을 나타낸다. 이 필드 ExtDataType의 값은, 확장 데이터를 식별하는 제 1의 값이며, 이 블록 blkExtensionData()를 포함한 규격서의 허가자(사용 인가자)가 할당한다고 정의할 수 있다. 필드 ExtDataVersion는, 확장 데이터를 식별하는 제 2의 값이며, 이 확장 데이터의 버젼 번호를 나타내는 것이라고 정의할 수 있다. 한편, 이 블록 blkExtensionData()에 있어서, 필드 ExtDataType 및 필드 ExtDataVersion의 값에는, 동일한 블록 ext_data_entry()가 2이상, 존재해서는 안 된다.

[0191] 필드 ExtDataStartAddress는, 32비트의 데이터 길이를 가지며, 이 필드 ExtDataStartAddress가 포함되는 확장 데이터의 엔트리(블록 ext_data_entry())에 대응하는 확장 데이터의 개시 주소를 나타낸다. 필드 ExtDataStartAddress는, 블록 blkExtensionData()의 선두 바이트로부터의 상대 바이트 수이며, 확장 데이터 ext_data의 개시 주소를 나타낸다. 한편, 필드 ExtDataStartAddress는, 다음에 나타내는 32비트 얼라이먼트의 조건을 만족해야 한다.

ExtDataStartAddress%4=0

[0192] 필드 ExtDataLength는, 32비트의 데이터 길이를 가지며, 이 필드 ExtDataStartAddress가 포함되는 확장 데이터의 엔트리(블록 ext_data_entries())에 대응하는 확장 데이터의 데이터 길이를 나타낸다. 데이터 길이는, 바이트 수로 나타난다.

[0193] 필드 NumberOfExtDataEntries로 나타난 갯수만큼, 확장 데이터의 엔트리(블록 ext_data_entry())가 기술되면,

각각 16비트의 데이터 길이를 가지며 임의의 데이터열로 구성되는 필드 padding_word가, 2 필드를 그룹으로 하여 임의의 회수 L1만큼 반복된다. 그 후, 확장 데이터의 본체가 저장되는 블록 DataBlock()이 기술된다. 블록 DataBlock()은, 1이상의 확장 데이터가 저장된다. 각각의 확장 데이터 ext_data는, 상술한 필드 ExtDataStartAddress 및 필드 ExtDataLength에 근거하여, 블록 DataBlock()로부터 추출된다.

[0194] 도 28은, 블록 blkExtensionData()에 있어서의 각 데이터의 참조 관계를 모식적으로 나타낸다. 필드 Length에 의해, 필드 Length 직후의 위치로부터 블록 blkExtensionData()의 끝까지의 데이터 길이가 나타난다. 필드 DataBlockStartAddress에 의해, 블록 DataBlock()의 개시 위치가 나타난다. 필드 NumberOfExtDataEntries로 나타나는 갯수만큼, 블록 ext_data_entry가 기술된다. 마지막 블록 ext_data_entry로부터 블록 DataBlock()의 사이에는, 임의의 길이로 필드 padding_word가 놓여진다.

[0195] 블록 DataBlock() 내에는, 블록 ext_data_entry()로 나타나는 확장 데이터 ext_data가 놓여진다. 각각의 확장 데이터 ext_data의 위치 및 데이터 길이는, 대응하는 블록 ext_data_entry()내의 필드 ExtDataStartAddress 및 필드 ExtDataLength에 의해 표시된다. 따라서, 블록 DataBlock()내에서의 확장 데이터 ext_data의 배치 순서는, 대응하는 블록 ext_data_entry()의 배치 순서와 일치하지 않아도 좋다.

[0196] 이와 같이, 확장 데이터를, 확장 데이터의 본체가 저장되는 블록 DataBlock()과 블록 DataBlock() 내의 확장 데이터에 대한 액세스 정보 등이 저장되는 블록 ext_data_entry()에 의한 2층 구조로 하여, 복수의 확장 데이터를 저장하는 것이 가능해진다.

[0197] 다음에, 상술의 확장 데이터의 일례의 작성 방법 및 독출 방법에 대해 설명한다. 도 29는, 블록 blkExtensionData()에 데이터를 기입할 때의 일례의 처리를 나타내는 플로차트이다. 이 도 29는, 블록 blkExtensionData()중의 (n+1)번째의 엔트리로서, 확장 데이터를 추가하고, 블록 blkExtensionData()를 고쳐 쓰는 경우의 예이다.

[0198] 먼저, 스텝(S10)에서, 기입하고자 하는 확장 데이터의 데이터 길이를 취득하고, 필드 ExtDataLength[n+1]의 값으로 설정한다. 한편, [n+1]의 기술은, (n+1) 번째의 엔트리의 번호에 대응한다. 다음에, 스텝(S11)에서, 현재의 블록 blkExtensionData()에 열거되어 있는 블록 ext_data_entry()의 필드 ExtDataLength 및 필드 ExtDataStartAddress의 값을 조사하고, 블록 DataBlock()의 사용 상황을 취득한다.

[0199] 그리고, 다음의 스텝(S12)에서, 블록 DataBlock()중에, 기입하고자 하는 확장 데이터의 데이터 길이인 필드 ExtDataLength[n+1]에 나타나는 데이터 길이 이상의, 연속한 빈 영역이 있는지 없는지가 판단된다. 만약, 있다고 판단되면, 처리는 스텝(S14)으로 이행된다.

[0200] 한편, 필드 ExtDataLength[n+1]에 나타나는 데이터 길이 이상의 연속한 빈 영역이 없다고 판단되면, 처리는 스텝(S13)으로 이행되며, 블록 blkExtensionData()에 있어서의 필드 Length의 값을 크게 하고, 필드 ExtDataLength[n+1]에 나타나는 데이터 길이 이상의 연속한 빈 영역을 블록 DataBlock()내에 만든다. 빈 영역이 생기면, 처리가 스텝(S14)으로 이행된다.

[0201] 스텝(S14)에서는, 확장 데이터를 기입하는 영역의 선두 주소를 결정하고, 그 선두 주소의 값을 필드 ExtDataStartAddress[n+1]로 한다. 다음의 스텝(S15)에서, 필드 ExtDataStartAddress[n+1]로부터, 상술의 스텝(S10)에서 설정된 필드 ExtDataLength[n+1]의 길이의 확장 데이터 ext_data[n+1]를 기입한다.

[0202] 데이터의 기입이 종료하면, 스텝(S16)에서, 블록 ext_data_entry()에 대해서, 필드 ExtDataLength[n+1]와 필드 ExtDataStartAddress[n+1]를 추가한다.

[0203] 한편, 상술한 기재에 있어서, 고쳐 쓰기를 실시하는 블록 blkExtensionData()는, 벌써 디스크 등의 기록 매체로부터 독출되어 기록 장치의 메모리에 기억되어 있는 것으로 가정한다. 그 때문에, 스텝(S13)에 있어서의, 필드 Length의 값의 변경에 의한 블록 blkExtensionData()의 확대는, 시스템에 의해 실행되며, 시스템이 메모리 할당을 적절히 실시하는 것으로 된다.

[0204] 도 30은, 블록 blkExtensionData()로부터 확장 데이터를 돌축할 때의 일례의 처리를 나타내는 플로차트이다. 한편, 이 도 30의 플로차트에 의한 처리는, 재생 전용의 기록 매체와 기록 가능한 기록 매체와의 양쪽 모두에 적용 가능한 것이다. 먼저, 최초의 스텝(S20)에서, 읽어들이려고 하는 확장 데이터가 준거하는 규격으로부터, 필드 ExtDataType의 값을 취득하고, 스텝(S21)에서, 읽어들이려고 하는 확장 데이터의 종별로부터, 필드 ExtDataVersion의 값을 취득한다.

[0205] 다음의 스텝(S22)에서, 블록 blkExtensionData()에 열거되어 있는 블록 ext_data_entry()를 1개씩 차례차례,

읽어들인다. 그리고, 스텝(S23)에서, 읽어들인 블록 ext_data_entry()에 포함되는 필드 ExtDataType 및 필드 ExtDataVersion의 값이, 상술의 스텝(S20) 및 스텝(S21)에서 취득한 필드 ExtDataType 및 필드 ExtDataVersion의 값과 일치하는지 아닌지가 판단된다.

[0206] 일치하지 않다고 판단되면, 처리는 스텝(S26)으로 이행되며, 블록 blkExtensionData()내에 열거되는 블록 ext_data_entry()를 모두 독출했는지 아닌지가 판단된다. 모두 독출했다고 판단되면, 처리는 스텝(S27)으로 이행되며, 이 블록 blkExtensionData()에는, 독출하려고 한 확장 데이터가 존재하지 않는다고 하고, 일련의 처리가 종료된다. 모두 독출하였다고 판단되면, 처리는 스텝(S22)으로 돌아가고, 다음의 블록 ext_data_entry()가 독출된다.

[0207] 상술의 스텝(S23)에 있어서, 블록 ext_data_entry()에 포함되는 필드 ExtDataType 및 필드 ExtDataVersion의 값이, 취득한 필드 ExtDataType 및 필드 ExtDataVersion의 값과 일치한다고 판단되면, 처리는 스텝(S24)으로 이행된다. 여기에서는, 블록 blkExtensionData()중의 [i]번째의 엔트리에서 일치하는 것으로 가정한다.

[0208] 스텝(S24)에서는, [i]번째의 엔트리의 블록 ext_data_entry()로부터 필드 ExtDataLength[i]의 값과 필드 ExtDataStartAddress[i]의 값을 읽어들인다. 그리고, 스텝(S25)에서, 스텝(S24)에서 읽어들인 필드 ExtDataStartAddress[i]로 나타나는 주소로부터, 필드 ExtDataLength[i]로 나타나는 데이터 길이만큼, 데이터를 독출한다.

[0209] 다음에, 상술한, 인덱스 파일 "index.bdmv", 무비 오브젝트 파일"MovieObject.bdmv", 플레이 리스트 파일 "xxxxx.mpls" 및 클립 정보 파일"zzzzz.clpi"에 각각 정의 가능한, 확장 데이터를 저장하는 확장 데이터 블록 blkExtensionData()에 대해 설명한다.

[0210] 먼저, 인덱스 파일 "index.bdmv"에 대해서 정의되는 일례의 확장 데이터 블록에 대해 설명한다. 여기에서는, 플레이 리스트마다 기록 가능한 기록 매체에 특유의 속성 정보를 부가하도록 한, 일례의 확장 데이터 블록에 대해 설명한다. 도 31은, 이 플레이 리스트 속성을 기술하기 위한, 파일 "index.bdmv"내의 필드 blkExtensionData()에 있어서의 블록 DataBlock()(도 27 참조)의 일례의 구조를 나타내는 문맥을 나타낸다. 이 도 31의 예에서는, 블록 DataBlock()이 블록 blkIndexExtensionData()로서 기술되어 있다.

[0211] 먼저, 상술의 도 27을 참조하여, 블록 blkExtensionData()에 대해 필드 ExtDataType를 값 "0x1000", 필드 ExtDataVersion를 값 "0x0100"으로 한다. 이러한 필드 ExtDataType 및 필드 ExtDataVersion에 기술된 값은, 예를 들면 재생장치 측에 있어서, 미리 ROM(Read Only Memory) 등에 기억된 테이블이 참조되어 식별된다.
블록 DataBlock()내의 필드 ExtDataStartAddress 및 필드 ExtDataLength로 표시되는 영역에, 블록 blkIndexExtensionData()가 저장된다.

[0212] 블록 blkIndexExtensionData()에 있어서, 필드 TypeIndicator는, 다음에 계속되는 데이터의 종류를 나타내는, ISO646에 규정된 부호화 방식으로 부호화된 4 문자로 구성되는 문자열이 기술된다. 이 도 31의 예에서는, 필드 TypeIndicator에 ISO646에 규정의 방식으로 부호화된 4문자의 문자열 "IDEX"가 기술되고, 다음에 계속되는 데이터 종류가 인덱스 파일에 있어서의 확장 데이터인 것이 표시된다.

[0213] 필드 TypeIndicator에 이어 32비트의 데이터 길이를 가지는 영역 reserved가 배치되며, 그 다음에, 32비트의 데이터 길이를 가지는 필드 TableOfPlayListStartAddress가 배치된다. 필드 TableOfPlayListStartAddress에는, 블록 blkTableOfPlayList()의, 이 블록 blkIndexExtensionData() 선두를 기준으로 한 개시 주소가 표시된다.

[0214] 필드 TableOfPlayListStartAddress의 다음에, 32비트의 데이터 길이를 가지는 필드 MakersPrivateDataStartAddress가 배치된 블록 blkMakersPrivateData()의 이 블록 blkIndexExtensionData() 선두를 기준으로 하는 개시 주소가 표시되며, 192비트의 데이터 길이를 가지는 영역 reserved를 통해 블록 blkUIAppInfoAVCHD()가 배치된다. 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N1로 나타나는 회수만큼 반복되며, 다음에, 블록 blkTableOfPlayLists()가 배치된다.이어서, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N2로 나타나는 회수만큼 반복되며, 다음에 블록 blkMakersPrivateData()가 배치된다. 이 블록 blkMakersPrivateData()의 뒤에, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N3로 나타나는 회수만큼 반복된다.

[0215] 한편, 블록 blkUIAppInfoAVCHD() 및 블록 blkMakersPrivateData()는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다.

[0216] 도 32는, 상술한 블록 blkTableOfPlayLists()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkTableOfPlayLists()의 마지막 바이트까지의 데이터 길이를 바이트 수로 표시한다. 필드 Length에 이어서, 플레이백 타이틀을 재생하기 위한 플레이 리스트에 관한 정보가 기술되는 블록 blkFirstPlaybackTitlePlayLists()와 메뉴 타이틀에 관한 정보가 기술되는 블록 blkMenuTitlePlayLists()가 배치된다. 이러한 블록 blkFirstPlaybackTitlePlayLists() 및 블록 blkMenuTitlePlayLists()는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다.

[0217] 다음에, 16비트의 데이터 길이를 가지는 필드 NumberOfTitlePlayListPair가 배치된다. 필드 NumberOfTitlePlayListPair는, 플레이백 타이틀 및 메뉴 타이틀 이외의 타이틀을 재생하기 위한 플레이 리스트의 수가 기술된다. 다음의 for 루프문에 따라, 필드 NumberOfTitlePlayListPair로 나타나는 수만큼, 블록 blkMovieTitlePlayListPair()가 기술된다. 블록 blkMovieTitlePlayListPair()는, 필드 PlayListFileName, 필드 PlayListAttribute 및 필드 RefToTitleId를 포함한다. 즉, 블록 blkMovieTitlePlayListPair()는, 이 for 루프문으로 나타나는[i]번째의 플레이 리스트에 대해서, 해당 플레이 리스트의 파일명, 해당 플레이 리스트에 부여된 속성, 및, 해당 플레이 리스트의 참조 타이틀 ID로 구성되는 플레이 리스트의 정보를 구조화한 것이다.

[0218] 이 for 루프문에 의한 배치 순서는, 기록순서로 여겨진다. 즉, 한 개의 플레이 리스트가 추가되면, 필드 NumberOfTitlePlayListPair의 값이 "1"만큼 증가되어, 기존의 플레이 리스트의 정보의 뒤에, 추가된 플레이 리스트의 정보가 추기된다.

[0219] 필드 PlayListFileName는, 40비트(5바이트)의 데이터 길이를 가지며, 플레이 리스트의 파일명이 ISO646에 규정된 부호화 방식으로 부호화되어 기술된다. 필드 PlayListFileName의 다음에, 6비트의 데이터 길이를 가지는 영역 reserved를 통해 필드 PlayListAttribute가 배치된다. 필드 PlayListAttribute는, 2비트의 데이터 길이를 가지며, 해당 플레이 리스트에 부여된 속성을 나타낸다. 플레이 리스트는, 그 형성된 원인에 근거하여, 클립의 생성과 함께 생성되는 플레이 리스트에 대응하는 제 1의 종류와 기존의 타이틀 혹은 플레이 리스트의 일부 또는 전부를 이용해 작성되는 플레이 리스트에 대응하는 제 2의 종류와 메뉴를 재생하기 위해서 이용하는 제 3의 종류와 같이 3 종류로 나눌 수 있으며, 각 플레이 리스트에는, 플레이 리스트의 종류에 따라, 각각 대응하는 속성 [Real](제 1의 종류), 속성 [Virtual](제 2의 종류) 및 속성 [Menu](제 3의 종류)이 부여된다.

[0220] 한편, 이하에서는 적절하게, 속성 [Real]이 부여된 플레이 리스트를 리얼 플레이 리스트, 속성 [Virtual]이 부여된 플레이 리스트를 버추얼 플레이 리스트, 속성 [Menu]가 부여된 플레이 리스트를 메뉴 플레이 리스트라고 부른다.

[0221] 필드 RefToTitleId은, 동일 루프내의 필드 PlayListFileName에 나타나는 플레이 리스트가 작성되는 때에 속하는 타이틀의 ID(번호)가 기술된다. 보다 구체적인 예로서는, 인덱스 파일 "index.bdmv"내의 블록 blkIndexes()에 있어서의, 대응하는 값 title_id가 기술된다. 한편, 해당 플레이 리스트가 단지 퍼스트 플레이백 타이틀로부터 재생되는 경우, 필드 RefToTitleId의 값은, 제 1의 고정치, 예를 들면 "0xFFFF"로 여겨진다. 또, 해당 플레이 리스트가 단지 메뉴 타이틀만으로터 재생되는 경우에는, 필드 RefToTitleId의 값은, 제 2의 고정치, 예를 들면"0xFFFE"로 여겨진다.

[0222] 여기서, 리얼 플레이 리스트, 버추얼 플레이 리스트 및 메뉴 플레이 리스트에 대해서, 개략적으로 설명한다. 속성 [Real]이 첨부되는 리얼 플레이 리스트는, 예를 들면 클립의 생성과 함께 생성되어 디스크에 기록된다. 리얼 플레이 리스트는, 소재를 가리키는 최초의 플레이 리스트가 되므로, 오리지날의 플레이 리스트라고도 불린다. 일례로서 리얼 플레이 리스트는, 생성된 클립의 선두를 IN점, 최후미를 OUT점으로서 각각 지정한다.

[0223] 리얼 플레이 리스트는, 스트림 실체인 클립을 참조한 플레이 리스트이며, 신규로 클립이 작성되면, 작성된 클립을 참조하는 리얼 플레이 리스트가 반드시 작성된다. 환언하면, 디스크상에 있어서, 어느 리얼 플레이 리스트로부터도 참조되어 있지 않은 클립은 존재하지 않는다. 따라서, 디스크상에 있어서의 리얼 플레이 리스트의 총재생 시간은, 해당 디스크상에 기록된 클립의 총재생 시간에 일치하게 된다. 즉, 디스크상에 있어서의 기록 가능한 남은 시간은, 이용자로부터 보면, 단지 리얼 플레이 리스트 또는 리얼 플레이 리스트만으로 구성되는 타이틀의 기록 가능 시간과 일치한다.

[0224] 리얼 플레이 리스트는, 소재인 클립과 항상 1 대 1의 대응 관계를 가지며, 리얼 플레이 리스트를 편집 등에 의해 삭제하면, 대응하는 클립도, 디스크상으로부터 삭제된다. 또, 리얼 플레이 리스트에 있어서, 클립의 참조 구간의 일부를 삭제하면, 삭제된 부분에 따라, 해당 리얼 플레이 리스트에 대응하는 클립의 일부도 삭제된다.

이에 의해, 리얼 플레이 리스트에 대한 편집은, 디스크상에 기록되는 클립의 실체의 개변을 수반하는 편집이기 때문에, 실체 편집 혹은 리얼 편집 등으로 불린다.

[0225] 속성 [Virtual]이 첨부되는 버추얼 플레이 리스트는, 기존의 타이틀 혹은 플레이 리스트의 일부 또는 전부를 이용해 작성되는 플레이 리스트이다. 버추얼 플레이 리스트는, 기존의 클립에 대해서 IN점 및 OUT점을 설정하고, IN점 및 OUT점으로 정의되는 구간을 참조하여 작성한 플레이 리스트이다.

[0226] 일례로서 버추얼 플레이 리스트는, 상술한 리얼 플레이 리스트에 대해서 IN점 및 OUT점을 지정한다. 예를 들면, 복수의 리얼 플레이 리스트에 대해, 각각 IN점 및 OUT점을 지정함과 동시에, 각각 IN점 및 OUT점으로 지정되는 복수의 구간의 재생순서를 지정한다. 버추얼 플레이 리스트를 바탕으로 하여, 추가로 버추얼 플레이 리스트를 작성할 수도 있다. 즉, 한 개 또는 복수의 버추얼 플레이 리스트에 대해서, IN점 및 OUT점을 지정하는 버추얼 플레이 리스트를 작성할 수 있다.

[0227] 버추얼 플레이 리스트는, 편집에 있어서, 예를 들면 사이즈가 큰 클립(스트림 실체)을 변경하지 않고, 고속으로 작성 가능하다. 또, 버추얼 플레이 리스트는, 삭제할 때도, 클립에 대한 참조 관계만을 삭제하여도 괜찮고, 클립 실체에 대해서 변경을 할 필요가 없다. 이와 같이, 버추얼 플레이 리스트의 편집은, 클립의 실체의 개변을 수반하지 않는 편집이기 때문에, 가상 편집 혹은 버추얼 편집 등으로 불린다.

[0228] 속성 [Menu]가 첨부되는 메뉴 플레이 리스트는, 메뉴를 재생하기 위해서 이용하는 플레이 리스트이며, 메뉴의 작성 및 갱신시에 생성된다. 메뉴 플레이 리스트는, 즉, 메뉴 타이틀을 재생하기 위한 무비 오브젝트로부터 호출되는 플레이 리스트이다.

[0229] 다음에, 플레이 리스트 파일 "xxxxx.mpls"에 대해서 정의되는 일례의 확장 데이터 블록에 대해 설명한다. 도 33은, 플레이 리스트 파일 "xxxxx.mpls"내의 블록 blkExtensionData()에 있어서의 블록 DataBlock()(도 27 참조)의 일례의 구조를 나타내는 문맥을 나타낸다. 이 도 33의 예에서는, 블록 DataBlock()이 블록 blkPlayListExtensionData()로서 기술되어 있다.

[0230] 먼저, 상술한 도 27을 참조하여, 플레이 리스트 파일 "xxxxx.mpls"내의 블록 blkExtensionData()에 있어서, 필드 ExtDataType 및 필드 ExtDataVersion가 각각 소정의 값, 예를 들면 상술한 바와 같이 동일하게 각각 값 "0x1000", 값 "0x0100"으로 되는 것으로 가정한다.

[0231] 블록 blkPlayListExtensionData()에 있어서, 필드 TypeIndicator는, 다음에 계속되는 데이터의 종류를 나타내는, ISO646에 규정된 부호화 방식으로 부호화된 4 문자로 구성되는 소정의 문자열이 기술된다. 이 필드 TypeIndicator에 기술되는 문자열에 의해서, 다음에 계속되는 데이터 종류가 플레이 리스트 파일에 있어서의 확장 데이터하는 것이 표시된다.

[0232] 필드 TypeIndicator의 다음에, 32비트(4바이트)의 데이터 길이를 가지는 영역 reserved를 통해 32비트의 데이터 길이를 가지는 필드 PlayListMarkExtStartAddress가 배치되며, 그 다음에, 32비트의 데이터 길이를 가지는 필드 MakersPrivateDataStartAddress가 배치된다. 필드 PlayListMarkExtStartAddress 및 필드 MakersPrivateDataStartAddress는, 각각, 블록 blkPlayListMarkExt() 및 블록 blkMakersPrivateData()의, 이 블록 blkPlayListExtensionData() 선두를 기준으로 하는 개시 주소가 나타난다.

[0233] 필드 MakersPrivateDataStartAddress의 다음에, 192비트의 데이터 길이를 가지는 영역 reserved를 통해 블록 blkPlayListMeta()가 배치된다. 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N1로 나타나는 회수만큼 반복되며, 다음에, 블록 blkPlayListMarkExt()가 배치된다. 게다가, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N2로 나타나는 회수만큼 반복되며, 다음에, 블록 blkMakersPrivateData()가 배치된다. 블록 blkMakersPrivateData()의 뒤에는, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N3로 나타나는 회수만큼 반복된다.

[0234] 도 34는, 블록 blkPlayListMeta()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length 직후부터 이 블록 blkPlayListMeta()의 마지막까지의 데이터 길이를 나타낸다. 필드 Length의 다음에, 각각 16비트의 데이터 길이를 가지는 필드 MakerID 및 필드 MakerModelCode가 배치된다. 필드 MakerID 및 필드 MakerModelCode는, 이 플레이 리스트 파일을 갱신한 메이커 및 해당 메이커의 기종을 식별하는 정보가 각각 기술된다.

[0235] 필드 MakerModelCode의 다음에, 32비트의 데이터 길이를 가지는 영역 reserved를 통해 16비트의 데이터 길이를 가지는 필드 RefToMenuThumbnailIndex가 배치된다. 이 필드 RefToMenuThumbnailIndex는, 이 플레이 리스트

파일에 의해 재생되는 일련의 클립을 대표하는 섬네일화상이 존재하는 경우, 그 섬네일화상을 특정하는 섬네일 번호가 기술된다. 다음에 8비트의 데이터 길이를 가지는 블록 blkTimeZone()가 배치되며, 이 플레이 리스트 파일을 갱신했을 때에, 장치로 설정되어 있는 타임 존(time zone)을 나타내는 정보가 기술된다. 게다가, 56비트의 데이터 길이를 가지는 필드 RecordTimeAndDate가 배치되며, 이 플레이 리스트 파일을 갱신한 시각 및 일자가 기술된다.

[0236] 필드 RecordTimeAndDate의 다음에, 8비트의 데이터 길이를 가지는 영역 reserved를 통해, 각각 8비트의 데이터 길이를 가지는 필드 PlayListCharacterSet 및 필드 PlayListNameLength가 배치되며, 계속 이어서, 255바이트의 데이터 길이를 가지는 필드 PlayListName가 배치된다. 이러한 필드 PlayListCharacterSet, 필드 PlayListNameLength 및 필드 PlayListName에 의해, 이 플레이 리스트 파일에 의해 나타나는 플레이 리스트에 첨부된 이름에 관한 정보가 기술된다.

[0237] 즉, 필드 PlayListCharacterSet에는, 필드 PlayListName에 기술되어 있는 문자열의 캐릭터 세트가 표시된다. 필드 PlayListNameLength는, 필드 PlayListName에 기술되는 플레이 리스트명의 바이트 길이를 나타낸다. 필드 PlayListName에는, 이 플레이 리스트에 첨부된 이름이 기술된다. 이 필드 PlayListName에 있어서, 왼쪽에서, 필드 PlayListNameLength에 표시된 만큼의 바이트수가 유효한 문자열이며, 그것이 이 플레이 리스트의 이름으로 여겨진다. 필드 PlayListName 중에서, 필드 PlayListNameLength로 표시된 유효한 문자열의 뒤의 바이트열에는, 어떠한 값이 들어가 있어도 괜찮다.

[0238] 필드 PlayListName의 다음에, 블록 Additional_data()가 배치된다. 블록 Additional_data()는, 추가적인 정보를 저장하기 위해서 예약된 영역에서 만나며, 32비트의 데이터 길이를 가지는 필드 Length2에 나타나는 값의 바이트 분의 데이터 길이를 가지는 영역이 예약된다.

[0239] 도 35는, 플레이 리스트 파일에 있어서의 블록 blkMakersPrivateData()의 일례의 구조를 나타내는 문맥을 나타낸다. 블록 blkMakersPrivateData()는, 이 플레이 리스트 파일에 관해서, 메이커 고유적인 정보가 기술되는 블록이다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length 직후부터 이 블록 blkMakersPrivateData()의 마지막까지의 데이터 길이를 나타낸다. 이 필드 Length는, 값 "0"으로 이 MakersPrivateData()에 정보가 기술되어 있지 않은 것을 가리킨다. 필드 Length가 값 "0"이외로 표시되면, if문 이하의 기술이 이루어진다.

[0240] 필드 DataBlockStartAddress는, 32비트의 데이터 길이를 가지며, 이 문맥중의, 메이커 고유 정보 본체가 저장되는 블록 DataBlock()의 개시 주소를, 이 블록 blkMakersPrivateData()의 선두 바이트로부터의 상대 바이트 수로 표시한다. 24비트의 데이터 길이를 가지는 영역 reserved를 통해, 8비트의 데이터 길이를 가지는 필드 NumberOfMakerEntries가 배치된다.

[0241] 다음의 for문에 따라, 필드 NumberOfMakerEntries로 표시되는 갯수만큼 확장 데이터의 엔트리, 즉, 필드 MakerID, 필드 MakerModelCode, 필드 MpdStartAddress 및 필드 MpdLength가 기술된다. 필드 MakerID 및 필드 MakerModelCode는, 각각 16비트의 데이터 길이를 가지며, 메이커의 식별 정보 및 해당 메이커에 의한 기종의 식별 정보가 기술된다. 또, 필드 MpdStartAddress 및 필드 MpdLength는, 각각 32비트의 데이터 길이를 가지며, 확장 데이터의 본체가 저장되는 블록 DataBlock()의, 이 블록 blkExtensionData()의 선두 바이트로부터의 상대 바이트 수에 의한 개시 주소와 데이터 길이를 가리킨다.

[0242] 필드 NumberOfMakerEntries로 나타나는 갯수만큼 확장 데이터의 엔트리가 기술되면, 각각 16비트의 데이터 길이를 가지며 임의의 데이터열로 구성되는 필드 padding_word가, 2 필드를 그룹으로서 임의의 회수 L1만큼 반복된다. 그 후, 확장 데이터의 본체가 저장되는 블록 DataBlock()이 기술된다. 블록 DataBlock()에는, 한 개 이상의 확장 데이터 ext_data가 저장된다. 즉, 필드 MakerID 및 필드 MakerModelCode로 표시되는 메이커 및 기종마다, 메이커 고유적인 확장 데이터가 블록 DataBlock()에 저장된다. 각각의 확장 데이터는, 상술한 필드 MpdStartAddress 및 필드 MpdLength에 근거하여, 블록 DataBlock()으로부터 추출된다.

[0243] 한편 플레이 리스트 파일에 있어서의 확장 데이터 블록 blkPlayListExtensionData()내의 블록 blkPlayListMarkExt()는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다.

[0244] 다음에, 클립 정보 파일 "zzzzz.clpi"에 대해서 정의되는 일례의 확장 데이터 블록에 대해 설명한다. 도 36은, 클립 정보 파일내의 블록 blkExtensionData()에 있어서의 블록 DataBlock()(도 27 참조)의 일례의 구조를 나타내는 문맥을 나타낸다. 이 도 36의 예에서는, 블록 DataBlock()이 블록 blkClipExtensionData()로서 기술되어 있다.

[0245] 블록 blkClipExtensionData()에 있어서, 필드 TypeIndicator는, 다음에 계속 되는 데이터의 종류를 나타내는, ISO646에 규정된 부호화 방식으로 부호화된 4문자로 구성되는 소정의 문자열이 기술된다. 이 필드 TypeIndicator에 기술되는 문자열에 의해서, 다음에 계속되는 데이터 종류가 클립 정보 파일에 있어서의 확장 데이터라는 것이 표시된다.

[0246] 필드 TypeIndicator의 다음에, 32비트(4바이트)의 데이터 길이를 가지는 영역 reserved를 통해, 각각 32비트의 데이터 길이를 가지는 필드 ProgramInfoExtStartAddress 및 필드 MakersPrivateDataStartAddress가 배치된다. 필드 ProgramInfoExtStartAddress 및 필드 MakersPrivateDataStartAddress는, 각각, 이 블록 blkClipExtensionData()내의 블록 blkProgramInfoExt() 및 블록 blkMakersPrivateData()의, 이 블록 blkClipExtensionData() 선두를 기준으로 한 개시 주소를 나타낸다.

[0247] 필드 MakersPrivateDataStartAddress의 다음에, 192비트의 데이터 길이를 가지는 영역 reserved를 통해 블록 blkClipInfoExt()가 배치된다. 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N1로 나타나는 회수만큼 반복되며, 다음에, 블록 blkProgramInfoExt()가 배치된다. 이어서, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N2로 나타나는 회수만큼 반복되며, 다음에, 블록 blkMakersPrivateData()가 배치된다. 블록 blkMakersPrivateData()의 뒤에는, 16비트의 데이터 길이를 가지는 패딩 워드 padding_word가 값 N3으로 나타나는 회수만큼 반복된다.

[0248] 도 37은, 블록 blkProgramInfoExt()의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 32비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkProgramInfoExt()의 끝까지의 데이터 길이를 나타낸다. 다음에, 8비트의 데이터 길이를 가지는 영역 reserved를 통해, 필드 NumberOfProgramSequence가 배치된다.

[0249] 필드 NumberOfProgramSequence는, 8비트의 데이터 길이를 가지며, 이 블록 blkProgramInfoExt()에 기술되는 프로그램 순서의 정보의 수가 나타난다. 이 예에서는, 필드 NumberOfProgramSequence의 값이 "1"로 고정적으로 되어 있다. 다음의 제 1의 for 루프문에 따라, 필드 NumberOfProgramSequence로 나타나는 수만큼, 필드 NumberOfStreamCodingInfoExtInPs[i] 및 다음의 추가로 제 2의 for 루프문이 반복된다. 제 2의 for 루프문에 의해, 8비트의 데이터 길이를 가지는 필드 NumberOfStreamCodingInfoExtInPs[i]로 나타나는 수만큼, 필드 StreamPID[i][j] 및 블록 blkStreamCodingInfoExt(i,j)의 그룹이 저장된다.

[0250] 16비트의 데이터 길이를 가지는 필드 StreamPID[i][j]는, 값 [i] 및 값 [j]로 지시되는 테이블을 구성하며, [i]번째의 프로그램 순서에 의해서 참조된 PMT(Program Map Table)에 기술된 기초 스트림에 대응하는 PID의 값을 나타낸다. 다음의 블록 blkStreamCodingInfoExt(i,j)는, 값 [i] 및 값 [j]로 표시되는 기초 스트림의 부호화 방식에 관한 정보가 기술된다.

[0251] 도 38은, 블록 blkStreamCodingInfoExt(i,j)의 일례의 구조를 나타내는 문맥을 나타낸다. 필드 Length는, 8비트의 데이터 길이를 가지며, 이 필드 Length의 직후부터 블록 blkStreamCodingInfoExt(i,j)의 끝까지의 데이터 길이를 나타낸다.다음의 필드 StreamCodingType는, 8비트의 데이터 길이를 가지며, 대응하는 스트림의 부호화의 종류를 나타낸다. 이 필드 StreamCodingType가 값 "0x1B"이면, if문 이하의 기술이 이루어지며, 최초의 8비트의 데이터 길이를 가지는 필드 HorizontalSize에 의해, 화면의 수직 방향의 사이즈 즉 라인수를 나타낸다. 한편, 블록 blkStreamCodingInfoExt(i,j)에 있어서, 필드 HorizontalSize 이하의 정보는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다.

[0252] 또, 클립 정보 파일내의 확장 데이터 블록 blkClipExtensionData()에 있어서의 블록 ClipEnfoExt() 및 블록 blkMakersPrivateData()는, 이 발명과 관련성이 별로 없기 때문에, 설명을 생략한다.

[0253] 다음에, 가상 재생기에 대해서, 개략적으로 설명한다. 상술한 바와 같은 데이터 구조를 가지는 디스크가 재생기에 장전되면, 재생기는, 디스크로부터 독출된 무비 오브젝트 등에 기술된 명령을, 재생기 내부의 하드웨어를 제어하기 위한 고유의 명령으로 변환할 필요가 있다. 재생기는, 이러한 변환을 행하기 위한 소프트웨어를, 재생기에 내장되는 ROM(Read Only Memory)에 미리 기억하고 있다. 이 소프트웨어는, 디스크와 재생기를 통해 재생기에 AVCHD 포맷의 규정에 따르는 동작을 실행시키므로, 가상 재생기라고 칭해진다.

[0254] 도 39a 및 도 39b는, 이 가상 재생기의 동작을 개략적으로 도시한다. 도 39a는, 디스크의 로딩시의 동작의 예를 나타낸다. 디스크가 재생기에 장전되어 디스크에 대한 이니셜 액세스가 되면(스텝 : S30), 한 개의 디스크에 대해 공유적으로 이용되는 공유 파라미터가 기억되는 레지스터가 초기화된다(스텝 : S31). 그리고, 다음의 스텝(S32)에서, 무비 오브젝트 등에 기술된 프로그램이 디스크로부터 독출되어 실행된다. 한편, 이니셜

액세스는, 디스크 장전시와 같이, 디스크의 재생이 처음으로 행해지는 것을 말한다.

[0255] 도 39b는, 재생기가 정지상태로부터 이용자에 의해 예를 들면 플레이 키가 가압되어 재생이 지시받았을 경우의 동작의 예를 나타낸다. 최초의 정지상태(스텝 : S40)에 대해서, 이용자에 의해, 예를 들면 원격 제어 명령 등을 이용하여 재생이 지시받는다(UO : User Operation). 재생이 지시받으면, 먼저, 레지스터 즉 공통 파라미터가 초기화되어(스텝 : S41), 다음의 스텝(S42)에서, 무비 오브젝트 실행 단계로 이행한다.

[0256] 무비 오브젝트의 실행 단계에 있어서의 플레이 리스트의 재생에 대해서, 도 40을 이용하여 설명한다. UO 등에 의해, 타이틀 번호 #1의 콘텐츠를 재생 개시하는 지시가 있었을 경우에 대해 생각한다. 재생기는, 콘텐츠의 재생 개시 지시에 따르고, 상술한 도 2에 나타나는 인덱스 테이블(Index Table)을 참조하여, 타이틀 #1의 콘텐츠 재생에 대응하는 오브젝트의 번호를 취득한다. 예를 들면 타이틀 #1의 콘텐츠 재생을 실현하는 오브젝트의 번호가 #1이었다고 하면, 재생기는, 무비 오브젝트 #1의 실행을 개시한다.

[0257] 이 도 40의 예에서는, 무비 오브젝트 #1에 기술된 프로그램은 2행으로 구성되며, 1행째의 명령이 "Play PlayList(1)"라고 하면, 재생기는, 플레이 리스트 #1의 재생을 개시한다. 플레이 리스트 #1은, 한 개 이상의 플레이 아이템으로구성되며, 플레이 아이템이 차례차례 재생된다. 플레이 리스트 #1중의 플레이 아이템의 재생이 종료하면, 무비 오브젝트 #1의 실행으로 돌아와서, 2행째의 명령이 실행된다. 도 40의 예에서는, 2행째의 명령이 "jump MenuTitle"이며, 이 명령이 실행되어 인덱스 테이블에 기술된 메뉴 타이틀(MenuTitle)을 실현하는 무비 오브젝트의 실행이 개시된다.

[0258] 다음에, 이 발명의 실시의 한 형태에 대해 설명한다. 이 발명에서는, 기록 매체에 클립이 기록될 때에, 기록되는 클립에 근거하는 챕터를 기존의 플레이 리스트에 대해서 추기하도록 된다. 그리고, 챕터를 기존의 플레이 리스트에 대해서 추기할 때에, 플레이 리스트에 소정으로 설치된 제약에 근거하여 해당 클립에 근거하는 챕터가 해당 플레이 리스트에 추기 가능한지 아닌지를 판단한다. 판단의 결과, 추기가 가능하다고 판단되면, 해당 클립에 근거하는 챕터를 해당 플레이 리스트에 대해서 추기한다. 한편, 추기가 불가하다고 판단되면, 신규로 플레이 리스트를 작성하고, 이 신규로 작성된 플레이 리스트에 대해서, 기록된 클립에 근거하는 챕터를 등록한다.

[0259] 이와 같이 기록 제어를 실시하여, 이용자는, 플레이 리스트에 설치된 제약을 의식하지 않고 클립의 기록을 실시할 수 있으므로 기록 시에 조작성이 향상되어 편리성이 뛰어나게 된다.

[0260] 도 41은, 이 발명의 실시의 한 형태에 적용 가능한 기록 장치의 일례의 구성을 개략적으로 도시하고 있다. 이 기록 장치는, 입력된 디지털 비디오 데이터 및 디지털 오디오 데이터를, 소정의 방식으로 압축 부호화 및 다중화한 AV스트림을 기록 매체에 기록하도록 하고 있다.

[0261] 이 도 41에 예시되는 기록 장치는, 외부로부터 입력되는 비디오 데이터 및 오디오 데이터를 기록 매체에 기록하는, 단독의 기록 장치로서 이용할 수도 있고, 광학계나 촬상 소자 등을 갖춘 카메라 블록과 조합하여, 촬상한 촬상 신호에 근거하는 비디오 데이터를 기록 매체에 기록하는, 비디오 카메라 장치의 기록 블록으로서 이용할 수도 있다.

[0262] 적용 가능한 압축 부호화나 다중화의 방식으로서는, 여러가지로 생각된다. 예를 들면, H.264 | AVC에 규정되는 방식을, 이 발명의 실시의 한 형태의 압축 부호화로서 적용할 수 있다. 이것에 한정하지 않고, MPEG2 방식에 근거하여 압축 부호화를 실시하도록 해도 괜찮다. 또, 다중화 방식은, 예를 들면 MPEG2 시스템을 적용할 수 있다. 이하에서는, 비디오 데이터의 압축 부호화를 H.264 | AVC에 규정되는 방식에 근거하여 실시하며, 비디오 데이터 및 오디오 데이터의 다중화를, MPEG2 시스템에 규정되는 방식에 근거하여 실시하는 것으로 설명한다.

[0263] 제어부(30)는, 예를 들면 CPU(Central Processing Unit), RAM(Random Access Memory) 및 ROM(Read Only Memory)등으로 구성되며(도시하지 않음), ROM에 미리 기억된 프로그램이나 데이터에 근거하여, RAM를 작업 메모리로서 이용하고 이 기록 장치의 기록부(10)의 각 부를 제어한다. 한편, 제어부(30)와 기록부(10)의 각부를 접속하는 경로는, 번잡함을 피하기 위해서, 도 41에서는 생략하고 있다.

[0264] UI(User Interface)부(31)는, 이 기록 장치의 동작을 이용자가 조작하기 위한 조작자가 소정으로 설치되며, 조작자에 대한 조작에 대응하는 제어 신호를 출력한다. 이 제어 신호는, 제어부(30)에 공급된다. 제어부(30)는, 이용자 조작에 따라 UI부(31)로부터 공급된 제어 신호에 근거하여 실행되는 프로그램의 처리에 의해, 기록부(10)의 각부의 동작을 제어한다. 예를 들면, UI부(31)에 대해 실행되는 조작에 따라, 기록 장치에 의

한 기록 동작의 개시 및 정지의 동작이 제어부(30)에 의해 제어된다.

[0265] 일례로서, 제어부(30)는, 소정의 프로그램으로 구성되는 소프트웨어의 기본적인 기능을 제공하는 OS(Operating System)상에서, 이용자 인터페이스의 제공이나 기록부(10)의 각부의 제어를 제공하는 어플리케이션 소프트웨어를 실행시킴과 동시에, 소프트웨어와 기록부(10) 각 부의 실제의 하드웨어와의 사이의 중개를 실시하는 드라이버 소프트웨어를 실행시킨다. OS는, 게다가 후술하는 기록 매체(20)상에 기록된 파일이나 데이터를 관리하기 위한 파일 시스템을 제공한다. 어플리케이션 소프트 웨어는, OS에 의해 제공되는 파일 시스템을 통해, 기록 매체(20)상에 기록된 파일에 액세스한다.

[0266] 베이스 밴드의 디지털 비디오 데이터가 단자(40)로부터 입력된다. 또, 해당 디지털 비디오 데이터에 따라, 베이스 밴드의 디지털 오디오 데이터가 단자(41)로부터 입력된다.

[0267] 디지털 비디오 데이터는 단자(40)로부터 기록부(10)에 입력되어, 비디오 엔코더(11)에 공급된다. 비디오 엔코더(11)는, 공급된 디지털 비디오 데이터를, 소정의 방식으로 압축 부호화 한다. H.264 | AVC에 규정되는 방식에 근거하여 압축 부호화가 되는 이 예에서는, 예를 들면, DCT(Discrete Cosine Transform)와 화면내 예측에 의해 프레임내 압축을 실시함과 동시에, 움직임 벡터를 이용한 프레임간 압축을 실시하고, 추가로 엔트리피 부호화를 실시하여 압축 효율을 높인다. 비디오 엔코더(11)에서 압축 부호화된 디지털 비디오 데이터는, H.264 | AVC의 기초 스트림(ES)으로서 멀티플렉서(MUX)(13)에 공급된다.

[0268] 디지털 오디오 데이터는 단자(41)로부터 기록부(10)에 입력되어, 오디오 엔코더(12)에 공급된다. 오디오 엔코더(12)는, 소정의 압축 부호화 방식, 예를 들면 AC3(Audio Code number 3)에 의해 압축 부호화 된다. 오디오 데이터의 압축 부호화 방식은, AC3에 한정되는 것은 아니다. 오디오 데이터를 압축 부호화하지 않고, 베이스 밴드의 데이터를 그대로 이용하는 일도 생각할 수 있다. 압축 부호화된 디지털 오디오 데이터는, 멀티플렉서(13)에 공급된다.

[0269] 멀티플렉서(13)는, 각각 압축 부호화되어 공급된 디지털 비디오 데이터 및 디지털 오디오 데이터를 소정의 방식으로 다중화하고, 한 개의 데이터 스트림으로 출력한다. MPEG2 시스템에 근거하는 다중화를 하는 이 예에서는, MPEG2의 트랜스포트 스트림을 이용하고, 공급된 압축 비디오 데이터 및 압축 오디오 데이터를 시분할로 다중화한다. 예를 들면, 멀티플렉서(13)는, 버퍼 메모리를 가지며, 공급된 압축 비디오 데이터 및 압축 오디오 데이터를 버퍼 메모리에 저장한다.

[0270] 버퍼 메모리에 저장된 압축 비디오 데이터는, 소정 사이즈마다 분할되어 헤더가 부가되고, PES(Packetized Elementary Stream) 패킷화 된다. 압축 오디오 데이터도 같이 소정 사이즈마다 분할되어 헤더가 부가되어 PES 패킷화 된다. 헤더에는, 패킷에 저장되는 데이터의 재생 시각을 나타내는 PTS나 복호 시각을 나타내는 DTS(Decoding Time Stanp)라는, MPEG2 시스템에 규정되는 소정의 정보가 저장된다. PES 패킷은, 더욱 분할되어 트랜스포트 패킷(TS패킷)의 페이 로드(pay load)에 내장된다. TS패킷의 헤더에는, 페이 로드에 담긴 데이터를 식별하기 위한 PID(Packet Identification)가 저장된다. 멀티플렉서(13)로부터 출력된 TS패킷은, 스트림 버퍼(14)에 일단 저장된다.

[0271] 한편, TS패킷은, 실제로는, MUX(13)에 있어서 추가로 소정 사이즈의 헤더가 부가되어 출력된다. 이, TS패킷에 대해서 소정의 헤더를 부가한 패킷을, 소스 패킷이라고 부른다.

[0272] 기록 제어부(15)는, 기록 매체(20)에 대한 데이터의 기록을 제어한다. 기록 매체(20)로서는, 예를 들면 기록 가능한 타입의 DVD(Digital Versatile Disc)를 이용할 수 있다. 이것에 한정하지 않고, 기록 매체(20)로서 하드 디스크 드라이브를 이용해도 괜찮고, 반도체 메모리를 기록 매체(20)에 적용하는 것도 가능하다.또, 기록 매체(20)로서 대용량을 실현한 Blu-rayDisc(블루 레이 디스크 : 등록상표)를 적용하는 일도 생각할 수 있다.

[0273] 기록 제어부(15)는, 스트림 버퍼(14)에 모은 데이터량을 감시하고, 스트림 버퍼(14)에 소정량 이상의 데이터가 모어지면, 스트림 버퍼(14)로부터 기록 매체(20)의 기록 단위 분의 데이터를 독출하여 기록 매체(20)에 기입한다.

[0274] 관리 정보처리부(16)는, 예를 들면 CPU, 작업 메모리로서의 RAM 및 프로그램의 소정의 데이터가 미리 기억되는 ROM으로 구성된다(도시하지 않음). 이것에 한정하지 않고, 관리 정보처리부(16)는, 예를 들면 제어부(30)에 있어서의 프로그램 처리에 의해 관리 정보처리부(16)의 기능을 실현하는 것도 가능하다. 이 경우, 예를 들면 제어부(30)가 가지는 RAM이 휘발성 메모리(17)로서 이용됨과 동시에, 불휘발성 메모리(18)가 제어부(30)에 접속된다.

[0275] 관리 정보처리부(16)는, 기록 데이터에 근거하여, 휘발성 메모리(17)를 작업메모리로서 이용하고, 상술한 인덱스 파일 "index.bdmv", 무비 오브젝트 파일"MovieObject.bdmv", 플레이 리스트 파일 "xxxxx.mpls"및 클립 정보 파일"zzzzz.clpi"에 저장하기 위한 정보를 생성한다. 생성된 정보는, 소정의 타이밍에서 기록 매체(20)에 기입된다.

[0276] 일례로서, 관리 정보처리부(16)는, 멀티플렉서(13)로부터 기록 데이터의 시간 정보를 취득함과 동시에, 기록 제어부(15)로부터 기록 데이터의 기록 매체(20)에 대한 주소 정보를 취득하고, 취득된 이러한 시간 정보 및 주소 정보에 근거하여 EP_map 정보가 생성된다. 또, UI부(31)에 대한 기록 개시, 기록 종료의 조작에 따라 제어부(30)로부터 출력되는 제어 신호와 멀티플렉서(13) 및 기록 제어부(15)로부터의 기록 데이터에 관한 정보에 근거하여, 클립 정보 파일 "zzzzz.clpi"의 생성함과 동시에, 생성된 클립 정보 파일의 정보가 플레이 리스트 파일에 추기되어 플레이 리스트 파일 "xxxxx.mpls"의 갱신을 한다.

[0277] 플레이 리스트 파일의 갱신 시에, 클립 정보 파일의 정보가 플레이 리스트 파일에 추기되는 것에 의해 플레이 리스트 파일에 대해서 소정에 설치된 제약을 만족하는지 아닌지를 판단한다. 판단의 결과, 제약을 만족하지 않는다고 판단되면, 신규로 플레이 리스트 파일이 작성되어 생성된 클립 정보 파일의 정보가 신규로 작성된 플레이 리스트 파일에 기록된다.

[0278] 게다가, 기록 매체(20)에 대해서 신규로 기록을 할 때, 인덱스 파일"index.bdmv"이나 무비 오브젝트 파일 "MovieObject.bdmv"의 생성 또는 갱신을 한다.

[0279] 도 42는, 이 발명의 실시의 한 형태에 적용 가능한 일례의 데이터 구조를 나타낸다. 인덱스 파일 "index.bdmv"는, 한 개 내지 복수의 타이틀을 가진다. 무비 오브젝트 파일 "MovieObject.bdmv"는, 인덱스 파일 "index.bdmv"가 가지는 타이틀에 대응하고, 한 개 내지 복수의 무비 오브젝트를 포함한다. 무비 오브젝트의 각각은, 1개의 플레이 리스트 파일 "xxxxx.mpls"를 호출한다. 플레이 리스트 파일 "xxxxx.mpls"는, 한 개 내지 복수의 플레이 아이템을 포함하며, 플레이 아이템의 각각은, 클립 정보 파일 "zzzzz.clpi"를 참조한다. 클립 정보 파일"zzzzz.clpi"는, 클립의 실체인 클립 AV스트림 파일 "zzzzz.m2ts"와 1대 1의 관계에 있다.

[0280] 이러한 구조에 대해서, 이용자에게는, 인덱스 파일 "index.bdmv"가 가지는 타이틀 단위로, 기록 매체에 기록된 클립이 보이게 된다. 이용자가 소망하는 타이틀을 선택하면, 무비 오브젝트 파일 "MovieObject.bdmv"로부터 해당 타이틀에 대응하는 무비 오브젝트가 참조된다. 그리고, 참조된 무비 오브젝트에 기술된 플레이 리스트 파일 "xxxxx.mpls"가 호출되며, 플레이 리스트 파일에 포함되는 플레이 아이템에 따라 클립 정보 파일 "zzzzz.clpi"가 참조되어 대응하는 클립 AV스트림 파일 "zzzzz.m2ts"가 재생된다.

[0281] 플레이 리스트 파일 "xxxxx.mpls"에 대해서 시각 정보를 나타내는 마크(플레이 리스트 마크)를 설치하여, 점프 위치를 설정할 수 있다. 마크에 의해서 챕터가 정의된다. 챕터는, 이용자로부터 보이는 타이틀내의 재생 단위이다. 기록 개시 위치에는, 반드시 마크가 설치된다. 기록 개시 위치 이외의 위치에 마크를 마련할 수도 있다.

[0282] 즉, 플레이 리스트 파일"xxxxx.mpls"에 대해서, 예를 들면 기록 개시에 따라 플레이 리스트 마크를 설정함과 동시에, 클립을 참조하는 플레이 아이템을 등록하는 것으로, 해당 플레이 리스트 파일에 대해서 챕터가 형성된다. 환언하면, 플레이 리스트 파일에 대해서 플레이 리스트 마크를 기록함과 동시에, 플레이 아이템을 기록하는 것으로, 해당 플레이 리스트 파일에 대해서 챕터가 기록된다고 말할 수 있다.

[0283] 상술한 것처럼, 리얼 플레이 리스트는, 클립과 함께 생성된다. 도 42의 예에서는, 플레이 리스트 파일 "00000.mpls", "00200.mpls" 및 "00018.mpls"가 리얼 플레이 리스트의 속성을 가진다.

[0284] 이중에서, 플레이 리스트 파일 "00000.mpls"는, 신규로 생성된 클립의 정보를 플레이 리스트에 추기 기록해 나가는 예이다. 예를 들면, 클립 AV스트림 파일 "00001.m2ts"에 대응하는 클립 정보 파일 "00001.clpi"를 참조하는 플레이 아이템 #0이 이미 저장되어 있는 플레이 리스트 파일 "00000.mpls"에 대해서, 신규로 기록된 클립 AV스트림 파일 "00125.m2ts"에 대응하는 클립 정보 파일 "00125.m2ts"를 참조하는 플레이 아이템 #1이 추기 기록된다. 플레이 아이템이 가리키는 선두의 시각에는, 각각 마크가 설치된다. 이 플레이 리스트 파일 "00000.mpls"를 재생하면, 플레이 아이템 #0에 근거하여 클립 AV스트림 파일 "00001.m2ts"가 재생되며, 이어서 플레이 아이템 #1에 근거하여 클립 AV스트림 파일 "00125.m2ts"가 재생된다.

[0285] 플레이 리스트 파일 "00200.mpls"는, 한 개의 클립에 대해서 한 개의 플레이 리스트 파일이 생성되며, 플레이 리스트 파일이 유일하게 한 개의 플레이 아이템만을 포함한 예이다.

[0286] 게다가, 플레이 리스트 파일 "00018.mpls"는, 복수의 플레이 아이템이 한 개의 클립을 참조하는 예이다. 예를 들면, 기록 개시 및 정지에 의해 플레이 아이템이 생성됨과 동시에, 한 개의 클립에 대해서 데이터가 추기되는 제어가 생각된다. 플레이 아이템 #0의 선두에 마크가 설치되며, 플레이 아이템 #0 및 플레이 아이템 #1을 연속적으로 재생하는 것으로, 클립 AV스트림 파일"00002.m2ts"의 전체가 재생된다.

[0287] 한편, 버추얼 플레이 리스트는, 도 42에 플레이 리스트 파일 "00005.mpls"로서 표시된 바와 같이, 이미 존재하는 클립에 대해서 재생 구간을 지정한다. 이 예에서는, 플레이 리스트 파일 "00005.mpls"에 포함되는 플레이 아이템 #0이 클립 정보 파일 "00028.clpi"를 참조하여 구간을 지정하고, 플레이 아이템 #1이 클립 정보 파일 "00002.clpi"를 참조하여 구간을 지정하고 있다. 또, 플레이 아이템#0 및 플레이 아이템 #1의 선두에, 마크가 설치되어 있다. 플레이 리스트 파일 "00005.mpls"를 재생하면, 먼저 플레이 아이템 #0에 근거하여 클립 AV스트림 파일 "00028.m2ts"의 지정 구간이 재생되며, 이어서, 플레이 아이템 #1에 근거하여 클립 AV스트림 파일 "00002.m2ts"가 재생된다.

[0288] 이 발명에서는, 상술한 것처럼, 신규로 기록되는 클립에 근거하는 챕터를, 기존의 플레이 리스트 파일에 추기하는지, 신규 플레이 리스트 파일을 작성하여 기록하는지를, 플레이 리스트 파일에 설치된 소정의 제약에 근거하여 판단하고 있다.플레이 리스트 파일에 설치되는 제약의 일례를 이하에 나타낸다. 제약에는, 포맷상의 제약, 실장 사양상의 제약, 상품 사양상의 제약 등을 생각할 수 있다. 포맷상의 제약으로서는, 이하의 제약(1)~제약(7)의 각 항목이 생각된다.

[0289] 제약(1) : 한 개의 플레이 리스트 파일에 존재 가능한 플레이 아이템 수의 상한.

[0290] 제약(2) : 한 개의 플레이 리스트 파일에 존재 가능한 플레이 리스트 마크 수의 상한.

[0291] 제약(3) : 한 개의 플레이 리스트 파일이 복수의 클립을 참조하는 경우에, 참조되는 복수의 클립 각각의 소정의 비디오 속성이 일치하고 있지 않으면 안 된다.

[0292] 제약(4) : 한 개의 플레이 리스트 파일의 파일 사이즈의 상한.

[0293] 제약(5) : 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈의 상한.

[0294] 제약(6) : 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일내의 큰 단위로 검색을 실시하는 엔트리 포인트의 합계의 상한.

[0295] 제약(7) : 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일내의 정밀한 단위로 검색을 실시하는 엔트리 포인트의 합계의 상한.

[0296] 이중에서, 제약(1)의 플레이 아이템 수 및 제약(2)의 플레이 리스트 마크 수에 관해서, 포맷상, 한 개의 플레이 리스트 파일에 존재 가능한 플레이 아이템 수의 상한이 예를 들면 999개로 되며, 한 개의 플레이 리스트 파일에 존재 가능한 플레이 리스트 마크 수의 상한이 예를 들면 999개로 된다. 따라서, 신규로 클립을 기록할 때에, 해당 클립에 근거하는 챕터를 추기하려고 하는 플레이 리스트 파일에 있어서, 챕터를 추기했을 때에 이러한 제약이 만족되는지 아닌지를 판단할 필요가 있다.

[0297] 제약(3)의 비디오 속성에 관해서, 한 개의 플레이 리스트 파일이 참조하는 복수의 클립 AV스트림 파일에 있어서, 예를 들면, 화상 프레임 사이즈, 주사 방식, 애스펙트비, 프레임 속도, 코덱의 종별이라고 하는, 비디오 데이터의 부호화에 관련되는 속성이 불변이도록, 포맷에서 규정되어 있다. 따라서, 신규로 클립을 기록할 때에, 해당 클립에 근거하는 챕터를 추기하려고 하는 플레이 리스트 파일에 이미 기록되어 있는 플레이 아이템에 의해 참조되는 클립에 있어서의 이러한 속성과 신규로 기록하려고 하는 클립에 있어서의 이러한 속성을 비교할 필요가 있다.

[0298] 제약(4)의 한 개의 플레이 리스트 파일의 파일 사이즈와 제약(5)의 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈에 대해서, 포맷상, 각각 상한이 규정되어 있다. 이 파일 사이즈의 규정은, 기록 장치 및 재생장치의 메모리 용량에 관련한다.

[0299] 즉, 예를 들면 기록시에, 기록되어 있는 클립에 대응하는 플레이 리스트나, 기록되어 있는 클립의 클립 정보 파일은, 일단, 기록 장치의 메모리에 저장되며, 메모리상에서 기록에 수반하는 플레이 리스트 파일이나 클립 정보 파일의 갱신 처리가 이루어진다. 그리고, 메모리에 저장된 이러한 플레이 리스트 파일이나 클립 정보 파일은, 소정의 타이밍에서 기록 매체에 기입된다. 플레이 리스트 파일이나, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈에 제한을 마련함으로써, 예를 들면 기록중에 메모리에 빈 용량이

없어져 기록이 강제적으로 정지되는 등의 상태가 되는 것이 방지된다.

[0300] 플레이 리스트 파일의 파일 사이즈의 상한은, 예를 들면 600kB(킬로바이트)가 상한으로 된다. 또, 한 개의 클립 정보 파일의 파일 사이즈의 상한이 예를 들면 1MB(메가바이트)로 되어 있는 경우, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계의 파일 사이즈의 상한은, 예를 들면 2MB로 된다.

[0301] 제약(6) 및 제약(7)의, 엔트리 포인트의 상한은, 상술의 클립 정보 파일의 파일 사이즈의 상한과 관련된다. 즉, 이미 설명한 것처럼, 큰 단위로 검색을 실시하는 엔트리 포인트 및 정밀한 단위로 검색을 실시하는 엔트리 포인트는, 클립 정보 파일내의 블록 blkCPI()에 있어서의 블록 blkEPMap()에 저장되는 정보이다. 즉, 큰 단위로 검색을 실시하는 엔트리 포인트는, 블록 blkEPMap()에 있어서의 필드 PTSEPCoarse 및 필드 SPNEPCoarse이며, 정밀한 검색을 실시하는 엔트리 포인트는, 블록 blkEPMap()에 있어서의 필드 PTSEPFine 및 필드 SPNEPFine이다.

[0302] 포맷상, 큰 단위로 검색을 실시하는 엔트리 포인트와 정밀한 단위로 검색을 실시하는 엔트리 포인트와의 각각에 대해서, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일에 대해 합계한 수에 대해서 상한이 규정되어 있다. 따라서, 신규로 클립을 기록하려고 할 때에, 해당 클립을 참조하는 플레이 아이템을 추기 하려고 하는 플레이 리스트 파일에 대해서, 플레이 아이템을 추기했을 때에 이러한 제약이 만족되는지 아닌지를 판단할 필요가 있다.

[0303] 실장 사양상의 제약으로서는, 이하의 항목이 생각된다.

[0304] 제약(8) : 플레이 리스트 파일에 있어서의 확장 데이터 블록내의 블록 blkMakersPrivateData()에 대해서, 다른 기기로 기록한 플레이 리스트 파일의 블록 blkMakersPrivateData()의 정보를 계승할 수 없는 경우는, 해당 플레이 리스트 파일을 갱신해서는 안 된다.

[0305] 예를 들면, 플레이 리스트 파일에 있어서의 확장 데이터 블록내의 블록 blkMakersPrivateData()는, 도 35에 있어서 블록 DataBlock()에 도시한 바와 같이, 특히 데이터 사이즈가 제한되어 있지 않기 때문에, 기기에 따라서는, 수 100kB라고 하는, 비교적 큰 데이터 사이즈의 데이터를 쓰는 사양도 생각할 수 있다. 기기의 실장에 따라서는, 예를 들면 메모리 용량의 제한 등의 요인에 의해, 다른 기기로 기록된 이러한 큰 사이즈의 블록 blkMakersPrivateData()의 정보를 계승하여 플레이 리스트 파일의 갱신이나 편집을 실시하는 것이 곤란하다는 것도 생각할 수 있다.

[0306] 상품 사양상의 제약으로서는, 이하의 항목이 생각된다.

[0307] 제약(9) : 타이틀이나 챕터에 관해서 다른 기기와 개념이 다른 경우에도, 이용자의 혼란을 부르지 않게 할 필요가 있다.

[0308] 제약(10) : 규정 시간 이상의 연속 촬영 및 기록 가능한 필요가 있다.

[0309] 제약(9)의 타이틀 및 챕터에 관해서, 기기에 의해서, 플레이 리스트 및 챕터의 구성 방법이 다른 것을 생각할 수 있다. 이러한 기기 사이에서, 클립의 기록된 기록 매체를 교환했을 경우에, 예를 들면 재생시에 있어서의 타이틀의 표시 등에 불편함이 생길 가능성이 있다.

[0310] 일례로서 텔레비젼 방송을 녹화하도록 된 거치형의 기록 재생장치(비디오 데크등)에서는, 한 개의 프로그램이 한 개의 플레이 리스트로서, 5분 간격으로 자동적으로 플레이 리스트 마크를 마련하여 챕터를 구성하는 사양으로 되며, 비디오 카메라 장치에서는 한 번의 촬영마다, 즉 기록 개시시에 플레이 리스트 마크를 마하여 챕터를 구성하여, 복수의 챕터를 한 개의 플레이 리스트에 등록하는 사양이 되는 것으로 한다. 또, 거치형의 기록 재생 기기에서는 플레이 리스트의 선두만 섬네일화상을 표시하며, 챕터 마다의 섬네일화상을 표시하지 않는 사양이 되는 것으로 한다.

[0311] 이러한 조건에 있어서, 거치형의 기록 재생장치로 작성된 플레이 리스트 파일이 기록된 기록 매체를, 비디오 카메라 장치에 장전하고, 비디오 카메라 장치로 기록된 클립을 참조하는 플레이 아이템을, 거치형의 기록 재생장치로 작성된 플레이 리스트 파일에 추기해 기록하는 것을 생각한다. 이 경우, 거치형의 기록 재생장치로 기록된 어느 프로그램의 마지막 분 장면으로서 비디오 카메라 장치로 촬영된 영상이 챕터로서 등록되게 된다. 이 기록 매체를, 비디오 카메라 장치로부터 다시 거치형의 기록 재생장치에 장전해 재생시키면, 비디오 카메라 장치로 촬영된 영상에 의한 챕터는, 섬네일화상으로서 표시되지 않고, 이전에 거치형의 기록 재생장치로 기록된 프로그램의 마지막 장면으로서 재생되므로, 부자연스럽다.

[0312] 제약(10)의 규정 시간 이상의 연속 촬영 및 기록에 관해서, 기록기의 사양으로서 소정 시간 이상의 연속 기록이 보증되는 것이 일반적이다. 한편, 위에서 설명한 바와 같이, 포맷상, 한 개의 플레이 리스트에 관련하는 클립 정보 파일의 엔트리 포인트의 합계수에 대해서 상한이 설치되어 있다. 예를 들면, 기록 매체의 빈 용량이 충분해도, 신규로 기록하는 클립을 참조하는 플레이 아이템을 추기 하려고 하는 플레이 리스트 파일에 있어서, 이미 참조되고 있는 클립 정보 파일에 있어서의 엔트리 포인트의 합계수와 포맷상 규정되고 있는 상한으로부터, 보증해야 할 소정 시간 분의 엔트리 포인트수를 확보할 수 없는 경우가 있다. 따라서, 신규로 클립을 기록하려고 할 때에, 해당 클립을 참조하는 플레이 아이템을 추기하려고 하는 플레이 리스트 파일에 관련하는 엔트리 포인트수에 근거하여 소정 시간 이상의 연속 기록이 가능한지 아닌지를 판단할 필요가 있다.

[0313] 다음에, 이 발명의 실시의 한 형태에 의한, 신규 기록하는 클립에 근거하는 챕터의, 플레이 리스트 파일에의 추기 여부를 판단하는 처리에 대해 설명한다. 상술한 제약(1)~제약(10)의 각 항목에 대해 각각 판단을 실시하고, 1항목에서도 추기 불가의 판단이 이루어졌을 경우에는, 신규로 플레이 리스트를 작성하며, 신규 기록하는 클립에 근거하는 챕터를, 신규로 작성된 플레이 리스트에 대해서 기록한다.

[0314] 도 43은, 신규 기록하는 클립에 근거하는 챕터의 플레이 리스트 파일에의 추기 여부를 판정하는 일련의 처리를 나타내는 플로차트이다. 먼저, 이 도 43의 플로차트를 이용해 전체적인 처리의 흐름을 개략적으로 설명한다. 여기에서는, 기록 매체(20)를 기록 가능한 타입의 DVD라고 하는, 디스크 형태 기록 매체(이하, 디스크)라고 한다. 또, 이하에 설명하는 각 판단 처리 등은, 도 41을 이용해 설명한 기록 장치에 있어서는, 제어부(30)에 있어서 행해진다.

[0315] 기록 매체(20)가 기록 장치에 장전되면, 제어부(30)에 의해 기록 제어부(15)가 제어되어 기록 매체(20)로부터 인덱스 파일 "index.bdmv"가 독출된다(스텝 : S100). 독출된 인덱스 파일 "index.bdmv"는, 예를 들면 관리 정보처리부(16)를 통해 휘발성 메모리(17)에 기억된다.

[0316] 다음의 스텝(S101)에서, 제어부(30)는, 휘발성 메모리(17)에 기억된 인덱스 파일 "index.bdmv"내에 기술된 정보에 근거하여, 다음에 기록되는 클립에 근거하는 챕터를 추기하기 위한 플레이 리스트(플레이 리스트 파일)의 후보를 특정한다.

[0317] 예를 들면, 인덱스 파일 "index.bdmv"에 있어서의 확장 데이터 블록인 블록 blkIndexExtensionData()내의 블록 blkTableOfPlayLists()를 참조하고, 가장 새롭게 기록된 리얼 플레이 리스트를 검색하여 그 리얼 플레이 리스트의 파일명을 취득한다.

[0318] 보다 구체적으로는, 도 29를 참조하면, 블록 blkIndexExtensionData()에 열거된 블록 blkMovieTitlePlayListPair()중에서, 필드 PlayListAttribute가 속성 [Real]을 나타내고 있는 블록 blkMovieTitlePlayListPair() 가운데, 가장 뒤에 기술되어 있는 블록 blkMovieTitlePlayListPair()를 검색하고, 검색된 블록 blkMovieTitlePlayListPair()로부터 필드 PlayListFileName의 데이터를 취득한다.

[0319] 다음의 스텝(S102)에서, 스텝(S101)에서 특정된 추기 후보의 플레이 리스트 파일이 기록 매체(20)로부터 독출되며 휘발성 메모리(17)에 기억된다. 그리고, 독출된 추기 후보의 플레이 리스트 파일에 관련하는 모든 클립 정보 파일이 기록 매체(20)로부터 독출된다. 독출된 클립 정보 파일은, 휘발성 메모리(17)에 기억된다.

[0320] 보다 구체적으로는, 도 10 ~ 도 12를 참조하면, 추기 후보의 플레이 리스트 파일에 있어서, 블록 blkPlayList()내의 모든 블록 blkPlayItem()이 참조되어 블록 blkPlayItem()에 기술되는 필드 ClipInformationFileName의 데이터가 모두 취득된다. 그리고, 취득된 필드 ClipInformationFileName의 데이터에 나타나는 클립 정보 파일을, 기록 매체(20)로부터 모두 독출한다.

[0321] 이하의 스텝(S104)~스텝(S111)에서, 휘발성 메모리(17)에 기억된 추기 후보의 플레이 리스트 파일 및 클립 정보 파일에 근거하여, 상술한 각 제약에 근거하는 챕터의 추기 여부의 판정이 이루어진다. 즉, 스텝(S104)에서, 상술의 제약(1)에 대응하여, 스텝(S101)에서 취득된 추기 후보의 플레이 리스트 파일에 포함되는 플레이 아이템 수에 근거하여 추기 여부의 판정을 실시한다. 다음의 스텝(S105)에서, 상술한 제약(2)에 대응하여, 추기 후보의 플레이 리스트 파일에 포함되는 플레이 리스트 마크 수에 근거하여 추기 여부의 판정을 실시한다. 다음의 스텝 (S106)에서, 상술한 제약(3)에 대응하여, 추기 후보의 플레이 리스트 파일에 기술되는 비디오 속성과 신규로 기록하려고 하는 클립의 비디오 속성과에 근거하여 추기 여부의 판정을 실시한다.

[0322] 다음의 스텝(S107)에서, 상술한 제약(4)에 대응하여, 추기 후보의 플레이 리스트 파일의 파일 사이즈에 근거하여 추기 여부의 판정을 실시한다. 다음의 스텝(S108)에서, 상술한 제약(5)에 대응하여, 추기 후보로 여겨진

플레이 리스트 파일로부터 참조되는 클립 정보 파일의 합계 파일 사이즈에 근거하여 추기 여부의 판정을 실시한다. 다음의 스텝(S109)에서, 상술한 제약(6) 및 제약(7)에 대응하여, 추기 후보로 여겨진 플레이 리스트 파일로부터 참조되는 클립 정보 파일에 저장되는 엔트리 포인트의 합계 수에 근거하는 추기 여부의 판정을 실시한다.

[0323] 다음의 스텝(S110)에서, 상술한 제약(8)에 대응하여, 추기 후보로 여겨진 플레이 리스트 파일내에 있어서의 다른 기기에 의한 고유적인 확장 데이터의 유무에 근거하는 추기 여부의 판정을 실시한다. 게다가, 다음의 스텝(S111)에서, 상술한 제약(9)에 대응하여, 추기 후보의 플레이 리스트 파일의 최종 갱신자에 근거하는 추기 여부의 판정을 실시한다.

[0324] 다음의 스텝(S112)에서는, 상술한 스텝(S104)~스텝(S111)까지의 각 판단 처리에 의한 판정 결과에 근거하여, 신규 기록되는 클립에 근거하는 챕터를, 추기 후보로 여겨진 플레이 리스트 파일에 대해서 추기 하는지 아닌지의 최종적인 판단이 이루어진다. 예를 들면, 스텝(S104)~스텝(S111)의 각 처리에 의한 판정 결과를, 제어부(30)가 가지는 레지스터 등에 각각 보관 유지하고, 스텝(S112)에 있어서, 레지스터에 보관 유지된 각 처리에 의한 판정 결과에 근거하여 판단을 실시한다.

[0325] 즉, 스텝(S112)에서는, 스텝(S104)~스텝(S111)까지의 각 처리의 모두에 대해 추기 가능하다고 판정되면, 신규 기록하는 클립에 근거하는 챕터를, 스텝 (S101)에서 취득된 추기 후보의 플레이 리스트 파일에 대해서 추기하도록 판단한다.

[0326] 한편, 상술한 스텝(S104)~스텝(S111)까지의 각 처리에 있어서, 추기 불가로 판정된 처리가 하나라도 존재한다면, 상술의 스텝(S101)에서 취득된 추기 후보의 플레이 리스트 파일에 근거하는 챕터의 추기가 불가하다고 판단한다. 이 경우, 신규의 리얼 플레이 리스트의 플레이 리스트 파일을 작성하고, 이 신규의 플레이 리스트 파일에 대해서, 신규로 기록되는 클립에 근거하는 챕터를 기록한다.

[0327] 한편, 상술한 스텝(S104)~스텝(S111)의 각 처리의 순서는, 이 순서에 한정되지 않는다. 즉, 스텝(S104)~스텝(S111)의 각 처리는, 임의의 순서로 실시할 수 있다. 또, 스텝(104)~스텝(S111)의 각 처리를 병렬적으로 실시하는 것도 가능하다. 게다가 스텝(104)~스텝(S111)의 각 처리의 모두를 실시하지 않고, 각 처리중 한 개 또는 복수의 처리를 선택적으로 실시해도 괜찮다.

[0328] 이하에, 상술한 스텝(104)~스텝(S111)의 각 처리의 각각을, 보다 상세하게 설명한다. 도 44는, 스텝(S104)에 의한, 제약(1)에 대응하는, 추기 후보의 플레이 리스트 파일에 포함되는 플레이 아이템 수에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다. 이 처리에서는, 스텝(S120)에 일례가 도시된 바와 같이, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayList()(도 11참조)를 참조하여 필드 NumberOfPlayItems의 값을 취득하고, 취득된 필드 NumberOfPlayItems의 값과 한 개 플레이 리스트 파일에 존재 가능한 플레이 아이템 수에 대해서 설정된 처리의 상한치, 예를 들면 값 "999"를 비교한다.

[0329] 비교의 결과, 필드 NumberOfPlayItems의 값이 소정의 상한치 미만이면, 신규 기록하는 클립에 근거하는 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다. 한편, 필드 NumberOfPlayItems의 값이 소정의 상한치 이상이면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0330] 도 45는, 스텝(S105)에 의한, 제약(2)에 대응하는, 추기 후보의 플레이 리스트 파일에 포함되는 플레이 리스트 마크 수에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다. 이 처리에서는, 스텝(S130)에 일례가 도시된 바와 같이, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayListMark()(도 14 참조)를 참조하여 필드 NumberOfPlayListMarks의 값을 취득하고, 취득된 필드 NumberOfPlayListMarks의 값과 한 개의 플레이 리스트 파일에 존재 가능한 플레이 리스트 마크 수에 대해서 설정된 소정의 상한치, 예를 들면 값 "999"를 비교한다.

[0331] 비교의 결과, 필드 NumberOfPlayListMarks의 값이 소정의 상한치 미만이면, 신규 기록하는 클립에 근거하는 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다. 한편, 필드 NumberOfPlayListMarks의 값이 소정의 상한치 이상이면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0332] 한편, 이미 설명한 것처럼, 플레이 리스트 마크에는, 엔트리 마크 및 링크 포인트의 2개의 타입이 존재한다. 이 스텝(S130)에 있어서는, 이러한 엔트리 마크 및 링크 포인트의 합계 수에 대해서 상한치 미만인지 아닌지가 판정된다.

[0333] 도 46은, 스텝(S106)에 의한, 제약(3)에 대응하는, 추기 후보의 플레이 리스트 파일에 기술되는 비디오 속성과

신규로 기록하려고 하는 클립의 비디오 속성과에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다. 먼저, 최초의 스텝 (S140)에 있어서, 기록하는 비디오 데이터의 속성 정보를 취득한다. 이 실시의 한 형태에서는, 기록 장치에 대해서 현재 설정되어 있는 촬영 모드나 녹화 모드에 근거하여, 기록하는 비디오 데이터의 속성 정보로서 화상 프레임 사이즈, 애스펙트비, 프레임 속도 및 주사 종별(인터레이스 / 프로그래시브)을 취득한다.

[0334] 이하의 스텝(S141)~스텝(S144)에서, 추기 후보의 플레이 리스트 파일에 관련하는 클립의 비디오 속성이 취득된다. 이 실시의 한 형태에서는, 추기 후보의 플레이 리스트 파일에 대해서 복수의 플레이 아이템이 저장되어 있는 경우, 가장 최초로 기록된 플레이 아이템이 참조하는 클립 정보 파일로부터, 비디오 속성을 취득한다. 개략적으로는, 비디오 속성으로서 스텝(S141)에서 화상폭을 취득하고, 스텝(S142)에서 화상 높이(라인수) 및 주사 종별을 취득하고, 스텝(S143)에서 애스펙트비를 취득하고, 스텝(S144)에서 프레임 속도를 취득한다.

[0335] 보다 구체적으로는, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayList()(도 11 참조)에 있어서, 가장 선두측에 저장되는 블록 blkPlayItem()(도 12 참조)를 참조하여, 필드 ClipInformationFileName의 데이터를 취득하고 해당 플레이 아이템이 참조하는 클립 정보 파일의 파일명을 취득한다. 상술한 도 43의 플로차트에 있어서의 스텝(S103)의 처리에 의해 독출된, 추기 후보의 플레이 리스트에 관련하는 모든 클립 정보 파일 가운데, 취득된 파일명에 대응하는 클립 정보 파일을 참조한다.

[0336] 스텝(S141)에서는, 해당 클립 정보 파일에 있어서의 확장 데이터인 블록 blkClipExtensionData()내의 블록 blkProgramInfoExt()(도 37 참조)가 참조되어 해당 블록 blkProgramInfoExt()에 있어서, 추가로 블록 blkStreamCodingInfoExt(i,j)(도 38 참조)가 참조된다. 그리고, 해당 블록 blkStreamCodingInfoExt(i,j)안의 필드 HorizontalSize의 값이 취득된다.

[0337] 스텝(S142)에서는, 해당 클립 정보 파일에 있어서의 블록 blkProgramInfo()(도 17 참조)안의 블록 blkStreamCodingInfo()(도 18 참조)가 참조되어 필드 VideoFormat의 값이 취득된다. 필드 VideoFormat는, 도 19를 이용해 이미 설명한 것처럼, 4비트로 표현 가능한 값을 이용하고, 비디오 데이터의 라인 수와 주사 방식(인터레이스 / 프로그래시브)과의 조합이 도시되어 있다.

[0338] 스텝(S143)에서는, 스텝(S142)과 같이 블록 blkStreamCodingInfo()가 참조되어 필드 AspectRatio의 값이 취득된다. 필드 AspectRatio에는, 도 21을 이용해 이미 설명한 것처럼, 4비트로 표현 가능한 값을 이용하고, 비디오 데이터의 화상 프레임의 애스펙트비가 표시되어 있다.

[0339] 스텝(S144)에서는, 스텝(S143) 및 스텝(S143)과 같이, 블록 blkStreamCodingInfo()가 참조되어 필드 FrameRate의 값이 취득된다. 필드 FrameRate에는, 도 20을 이용해 이미 설명한 것처럼, 4비트로 표현 가능한 값을 이용하고, 비디오 데이터의 프레임 속도가 표시되어 있다.

[0340] 스텝(S145)에서는, 스텝(S140)에서 기록기로부터 취득된, 기록하는 비디오 데이터의 속성 정보의 각각과 스텝(S141)~스텝(S144)에서 취득된, 추기 후보의 플레이 리스트 파일에 관련하는 클립의 비디오 속성의 각각이 비교된다.

[0341] 비교의 결과, 기록하는 비디오 데이터의 속성 정보의 각각과 추기 후보의 플레이 리스트 파일에 관련하는 클립의 비디오 속성의 각각이 모두 일치하고 있으면, 신규 기록하는 클립에 근거하는 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다. 한편, 기록하는 비디오 데이터의 속성 정보의 각각과 추기 후보의 플레이 리스트 파일에 관련하는 클립의 비디오 속성의 각각에 있어서, 한 개라도 불일치의 속성이 있는 경우, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0342] 도 47은, 스텝(S107)에 의한, 제약(4)에 대응하는, 추기 후보의 플레이 리스트 파일의 파일 사이즈에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다. 이 처리에 대해서는, 추기 후보의 플레이 리스트에 대해서 적어도 소정 챕터 수를 추기해도, 한 개의 플레이 리스트 파일의 파일 사이즈의 상한을 넘지 않는 다는 것을 이용하여, 추기의 여부를 판단한다.

[0343] 한편, 소정 챕터 수는, 하나의 챕터, 복수 챕터, 한 개의 플레이 리스트 파일에 존재 가능한 최대 챕터 수에 대한 나머지의 챕터 수 등, 기록기의 설계 사상등에 의해 여러가지로 생각된다. 여기에서는, 소정의 챕터 수를, 한 개의 플레이 리스트 파일에 존재 가능한 최대 챕터 수에 대한 나머지의 챕터 수인 것으로 한다.

[0344] 먼저, 스텝(S150)에 있어서, 미리 계산해 둔, 하나의 챕터 마다의 플레이 리스트 파일의 사이즈 증가분 SIZE_1CHAP를 취득한다. 즉, 이 스텝(S150)에서는, 예를 들면 하나의 챕터를 형성하기 위한 플레이 아이템 및 플레

이 리스트 마크의 데이터량을 계산한다.

[0345] 도 12를 참조하면, 플레이 아이템에 있어서, 필드 StillMode 및 필드 StillTime, 및, 블록 blkUOMaskTable() 및 블록 blkSTNTable()는, 데이터 길이가 변동할 가능성이 있는 데이터이지만, 실제로는, 이러한 데이터는, 기록 모드나 기록기에 의해서 고정 길이로로 여겨진다. 또, 플레이 리스트 마크는, 도 14를 참조하면, 각 필드의 값이 고정 길이로 되어 있다. 이와 같이, 한 개의 클립을 기록할 때에 생성되는 플레이 아이템 및 플레이 리스트 마크의 데이터 사이즈는, 미리 계산할 수 있다. 또, 계산된 값은, 기록 시간에 의존하지 않는 성질의 것이다. 이 한 개의 챕터 마다의 플레이 리스트 파일의 사이즈 증가분 SIZE_1 CHAP의 값은, 미리 계산되어 예를 들면 기록기가 가지는 ROM 등에 기억된다. 이 도 47의 플로차트에 의한 처리를 실행할 때마다 계산해도 괜찮다.

[0346] 플레이 리스트에 대해서 챕터를 추기하는 경우에는, 플레이 아이템과 플레이 리스트 마크가 증가하게 된다. 따라서, 플레이 리스트에 대해서 챕터를 추기할 때, 제약(1) 및 제약(2)의, 한 개의 플레이 리스트 파일에 존재 가능한 플레이 아이템 및 플레이 리스트 마크 수도 고려에 넣을 필요가 있다.

[0347] 스텝(S151)에서는, 추기 후보의 플레이 리스트 파일에 추기 가능한 플레이 아이템수PI_REMAIN를 계산한다. 즉, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayList()(도 11 참조)를 참조하여 필드 NumberOfPlayItems의 값을 취득하고, 추기 후보의 플레이 리스트에 포함되는 플레이 아이템 수를 구한다. 그리고, 추기 후보의 플레이 리스트에 포함되는 플레이 아이템수를, 한 개의 플레이 리스트 파일에 존재 가능한 플레이 아이템 수에 대해서 설정된 소정의 상한치 PI_MAX, 예를 들면 "999"로부터 줄이고, 추기 후보의 플레이 리스트에 추기 가능한 플레이 아이템수 PI_REMAIN를 구한다.

[0348] 즉, 추기 후보의 플레이 리스트 파일 추기 가능한 플레이 아이템 수 PI_REMAIN는, 아래와 같은 식(1)에 의해 구해진다.

[0349] PI_REMAIN=PI_MAX－NumberOfPlayItems.....(1)

[0350] 스텝(S152)에서는, 추기 후보의 플레이 리스트 파일에 추기 가능한 플레이 리스트 마크 수 MARK_REMAIN를 계산한다. 즉, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayListMark()(도 14 참조)를 참조하여 필드 NumberOfPlayListMarks의 값을 취득하고, 추기 후보의 플레이 리스트에 포함되는 플레이 리스트 마크의 수를 구한다. 그리고, 추기 후보의 플레이 리스트에 포함되는 플레이 리스트 마크 수를, 한 개의 플레이 리스트에 존재 가능한 플레이 아이템 수에 대해서 설정된 소정의 상한치 MARK_MAX, 예를 들면 "999"로부터 줄이고, 추기 후보의 플레이 리스트에 추기 가능한 플레이 리스트 마크수 MARK_REMAIN를 구한다.

[0351] 즉, 추기 후보의 플레이 리스트 파일에 추기 가능한 플레이 리스트 마크 수MARK_REMAIN는, 아래와 같은 식(2)에 의해 구해진다.

[0352] MARK_REMAIN=MARK_MAX－NumberOfPlayListMarks.....(2)

[0353] 다음의 스텝(S153)에서는, 스텝(S151)에서 구해진 추기 후보의 플레이 리스트에 추기 가능한 플레이 아이템 수 PI_REMAIN와 스텝(S152)에서 구해진 추기 후보의 플레이 리스트 파일에 추기 가능한 플레이 리스트 마크 수 MARK_REMAIN중 작은 분을, 추기 후보에 추기 가능한 챕터 수 CHAP_REMAIN로 한다.

[0354] 즉, 추기 후보의 플레이 리스트 파일에 추기 가능한 챕터 수 CHAP_REMAIN는, [MIN]을 괄호내의 값중 작은 것을 선택하는 것을 나타내는 것으로서, 상술의 식(1) 및 식(2)의 결과를 이용하고, 아래와 같은 식(3)에 의해 구해진다. 한편 [MIN]은, 괄호내의 값중 최소의 값을 선택하는 것을 나타낸다.

[0355] CHAP_REMAIN=MIN (PI_REMAIN, MARK_REMAIN).....(3)

[0356] 다음의 스텝(S154)에서는, 추기 후보의 플레이 리스트 파일의 파일 사이즈를 취득하고, 제약(4)의 한 개의 플레이 리스트 파일의 파일 사이즈의 상한 PL_SIZE_MAX에 대해서 남겨진 데이터 사이즈 SIZE_REMAIN를 구한다. 플레이 리스트 파일의 파일 사이즈는, 예를 들면 OS의 파일 시스템에 의해 제공되는 기능를 이용해 취득할 수 있다. 데이터 사이즈 SIZE_REMAIN(kB)는, 아래와 같은 식(4)에 의해 구해진다. 한편 추기 후보의 플레이 리스트 파일의 파일 사이즈를 FILE_SIZE(kB)로 한다.

[0357] SIZE_REMAIN(kB)=PL_SIZE_MAX(kB)－FILE_SIZE(kB).....(4)

[0358] 다음의 스텝(S155)에서, 추기 후보의 플레이 리스트 파일에 대해서 추기 가능한 챕터 분의 플레이 아이템 및 플레이 리스트 마크를 추기 했을 경우에, 추기 후보의 플레이 리스트 파일의 파일 사이즈가 한 개의 플레이 리스

트 파일의 파일 사이즈의 상한을 넘는지 아닌지가 판단된다.

[0359] 즉, 스텝(S150)에서 구한 한 챕터 마다의 플레이 리스트 파일의 사이즈 증가분 SIZE_1 CHAP과, 식(3) 및 식(4)의 결과를 이용하여, 아래와 같은 식(5)을 만족하는지 아닌지가 판단된다.

[0360] $SIZE\_1CHAP \times CHAP\_REMAIN \geq SIZE\_REMAIN$.....(5)

[0361] 만약, 식(5)가 만족되면, 신규 기록의 클립에 근거하는 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단된다. 한편, 식(5)가 만족되지 않으면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다거 판단한다.

[0362] 도 48은, 상술의 스텝(S108)의, 제약(5)에 대응하는, 추기 후보로 여겨진 플레이 리스트 파일로부터 참조되는 클립 정보 파일의 합계 파일 사이즈에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다.

[0363] 상술의 제약(5)에 의하면, 추기 후보로 여겨진 플레이 리스트 파일로부터 참조되는 클립 정보 파일의 합계 파일 사이즈에는, 예를 들면 2MB와 같이, 상한치 CLIP_SUM_MAX가 설정된다. 챕터의 추기에 의해 추기 후보의 플레이 리스트에 대해서 클립 정보 파일이 추가되었을 때에, 해당 추기 후보의 플레이 리스트 파일에 관련하는 클립 정보 파일의 합계 사이즈가, 이 상한치 CLIP_SUM_MAX를 넘지 않게 할 필요가 있다.

[0364] 스텝(S160)에서, 미리 계산해 둔, 한 개의 클립 정보 파일의 최대 사이즈 SIZE_1 CLIP를 취득한다. 클립 정보 파일은, 블록 blkEPMap()에, PTS치와 클립 AV스트림 파일의 바이트 주소를 관련짓는 엔트리 포인트의 정보가 저장된다(도 23~ 도 26 참조). 이 엔트리 포인트의 정보는, 기록 시간에 의해 변화하는 값이다.클립 정보 파일에 저장되는 그 외의 정보는, 부호화 방식 등에 의해 기록 시간에 의존하지 않는 고정적인 값이다. 한편, 상술한 것처럼, 기록기의 사양으로서 연속 기록을 보증하는 최저 시간이 설정된다. 엔코더에 있어서의 비디오나 오디오의 부호화 방식에 의해, 연속 기록을 보증하는 최저 시간 분의 엔트리 포인트 수가 정해진다. 연속 기록을 보증하는 최저 시간 분의 엔트리 포인트의 합계 데이터 사이즈를 계산하고, 계산 결과에 근거하여 최대 사이즈 SIZE_1 CLIP를 구한다.

[0365] 다음의 스텝(S161)에서, 추기 후보의 플레이 리스트 파일로부터 참조되고 있는 모든 클립 정보 파일에 대해 파일 사이즈를 취득하고, 합계 사이즈 SIZE_TOTAL_CLIP를 계산한다.

[0366] 즉, 추기 후보의 플레이 리스트 파일내의 모든 블록 blkPlayItem()(도 12 참조)를 참조하고, 각 블록 blkPlayItem()에 대해 필드 ClipInformationFileName의 데이터를 취득한다. 그리고, 취득된 필드 ClipInformationFileName의 데이터에 나타나는 클립 정보 파일의 파일 사이즈를, 추기 후보의 플레이 리스트 파일내의 모든 블록 blkPlayItem()에 대해 각각 구하여 합계한다. 클립 정보 파일의 파일 사이즈는, 예를 들면 OS의 파일 시스템에 의해 제공되는 기능을 이용해 취득할 수 있다.

[0367] 다음의 스텝(S162)에서, 아래와 같은 식(6)에 따라, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈의 상한치로부터, 스텝(S161)에서 계산된, 추기 후보의 플레이 리스트 파일로부터 참조되고 있는 모든 클립 정보 파일의 합계 사이즈 SIZE_TOTAL_CLIP를 줄인 값과, 스텝(S160)에서 계산된, 한 개의 클립 정보 파일의 최대 사이즈 SIZE_1 CLIP가 비교되며, 추기 후보의 플레이 리스트 파일에 대해서 최대 사이즈 SIZE_1 CLIP의 클립 정보 파일을 추가 가능한가 아닌가가 판정된다.

[0368] $CLIP\_SUM\_MAX - SIZE\_TOTAL\_CLIP \geq SIZE\_1CLIP$.....(6)

[0369] 비교의 결과, 만약, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈의 상한치로부터, 추기 후보의 플레이 리스트 파일로부터 참조되고 있는 모든 클립 정보 파일의 합계 사이즈 SIZE_TOTAL_CLIP를 줄인 값이, 한 개의 클립 정보 파일의 최대 사이즈 SIZE_1 CLIP보다 크다고 판정되면, 신규 기록의 클립에 근거하는 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단된다.

[0370] 한편, 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일의 합계 파일 사이즈의 상한치로부터, 추기 후보의 플레이 리스트 파일로부터 참조되고 있는 모든 클립 정보 파일의 합계 사이즈 SIZE_TOTAL_CLIP를 줄인 값이, 한 개의 클립 정보 파일의 최대 사이즈 SIZE_1 CLIP보다 작으면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0371] 도 49는, 상술의 스텝(S109)의, 제약(6) 및 제약(7)에 대응하는, 추기 후보로 여겨진 플레이 리스트 파일로부터 참조되는 클립 정보 파일에 저장되는 엔트리 포인트의 합계 수에 근거하는 추기 여부 판단의 일례의 처리를 나타낸다.

[0372] 이미 설명한 것처럼, 한 개의 플레이 리스트 파일로부터 참조되는 클립 정보 파일에 관해, 블록 blkEPMap()에 저장되는 엔트리 포인트의 합계 수에 대해서 상한이 설치되어 있다. 챕터의 추기에 의해 추기 후보의 플레이 리스트에 대해서 클립 정보 파일이 추가되었을 때에, 해당 추기 후보의 플레이 리스트 파일에 관련하는 클립 정보 파일에 저장되는 엔트리 포인트의 합계 수가, 이 상한치를 넘지 않게 할 필요가 있다.

[0373] 한편, 제약(6) 및 제약(7)에 의하면, 큰 단위로 검색을 실시하는 엔트리 포인트와 정밀한 단위로 검색을 실시하는 엔트리 포인트와의 각각에 대해 상한이 설치되어 있다. 따라서, 판정도, 클립 정보 파일에 저장되는 큰 단위로 검색을 실시하는 엔트리 포인트 및 정밀한 단위로 검색을 실시하는 엔트리 포인트의 각각 에 대해서, 실시할 필요가 있다.

[0374] 한 개의 플레이 리스트 파일에 관련되는 클립 정보 파일내의, 큰 단위로 검색을 실시하는 엔트리 포인트의 합계치의 상한치 MAX_EP_COARSE는, 예를 들면 24576개로 여겨진다. 또, 정밀한 단위로 검색을 실시하는 엔트리 포인트의 합계치의 상한치 MAX_EP_FINE는, 예를 들면 180000개로 여겨진다.

[0375] 먼저, 스텝(S170)에서, 미리 계산해 둔 한 개 챕터 마다의 최대 엔트리 포인트수를 취득한다. 상술한 것처럼, 기록기의 사양으로서 일반적으로, 연속 기록이 보증되는 최저 시간이 설정된다. 이 연속 기록이 보증되는 최저 시간 분의 기록을 실시했을 때의, 최대의 엔트리 포인트 수를 미리 계산해 두고, 이 스텝 (S170)에서, 이 값을 취득한다.

[0376] 도 25 및 도 26을 이용해 설명한 것처럼, 큰 단위로 검색을 실시하는 엔트리 포인트는, 11.5초 마다(PTS : 엔트리 PTSEPCoarse) 또는 25MB 마다(소스 패킷 번호 : 엔트리 SPNEPCoarse)에 설치된다. 여기서, 연속 기록이 보증되는 최저 시간을 시간 MIN_TIME(hr), 해당 최저 시간 분의 기록을 실시했을 경우에 생성되는 클립 AV스트림 파일의 데이터량을 데이터량 MIN_SIZE(MB)로 했을 경우, 큰 단위로 검색을 실시하는 엔트리 포인트에 대해서, 한 개의 챕터 마다의 최대 엔트리 포인트수 NEEDED_EP_COARSE는, 다음 식(7)에 의해 구해진다. 한편, 식(7) 및 후술하는 식(8)에 있어서, [CEIL]는, 괄호내의 값에 대해서 소수점을 잘라내는 것을 나타낸다.

[0377] NEEDED_EP_COARSE=CEIL (3600[sec]×MIN_TIME[hr]÷11.5[sec])+CEIL (MIN_SIZE[MB]÷25[MB])..... (7)

[0378] 또, 정밀한 단위로 검색을 실시하는 엔트리 포인트는, GOP마다, PTS의 정밀도(엔트리 PTSEPFine) 또는 소스 패킷의 정밀도(엔트리 SPNEPFine)로 설치된다. 이 실시의 한 형태에서는, 정밀한 단위로 검색을 실시하는 엔트리 포인트로서 엔트리 PTSEPFine만을 이용하는 것으로 하며, 프레임 주파수를 29.97Hz, 1GOP가 15프레임으로 구성된다고 했을 경우, 정밀한 단위로 검색을 실시하는 엔트리 포인트에 대해서, 한 개 챕터 마다의 최대 엔트리 포인트 수 NEEDED_EP_FINE는, 다음 식(8)에 의해 구해진다. 한편 식(8) 중, [90[kHz]÷3003]은, PTS 정밀도에 근거하는 프레임 주파수(29.97 Hz)를 나타낸다.

[0379] NEEDED_EP_FINE=CEIL(3600[sec]×MIN_TIME[hr]×90[kHz]÷3003÷15[Frame/GOP])..... (8)

[0380] 다음의 스텝(S171)에서, 추기 후보의 플레이 리스트 파일로부터 참조되고 있는 모든 클립 정보 파일에 대해서, 큰 단위로 검색을 실시하는 엔트리 포인트와 정밀한 단위로 검색을 실시하는 엔트리 포인트를 각각 취득하고, 큰 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_COARSE와 정밀한 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_FINE를 각각 구한다.

[0381] 보다 구체적으로는, 클립 정보 파일내의 필드 NumberOfStreamPIDEntries와 필드 NumberOfEPCoarseEntries[k] 및 필드 NumberOfEPFineEntries[k]와에 근거하여, 큰 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_COARSE와 정밀한 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_FINE를 구할 수 있다.

[0382] 다음의 스텝(S172)에서는, 추기 후보의 플레이 리스트 파일에 대해서 챕터를 추기 했을 때에, 큰 단위로 검색을 실시하는 엔트리 포인트에 대해서, 상한치 MAX_EP_COARSE를 초과하는 지의 여부가 판정된다. 즉, 상술의 스텝(S170)에서 취득된, 큰 단위로 검색을 실시하는 엔트리 포인트의 한 개 챕터 마다의 최대 엔트리 포인트 수 NEEDED_EP_COARSE와 스텝(S171)에서 구해진 큰 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_COARSE를 이용하여, 다음 식(9)에 근거하여 판정이 이루어진다.

[0383] MAX_EP_COARSE－TOTAL_EP_COARSE≥NEEDED_EP_COARSE.....(9)

[0384] 만약, 각 값의 관계가 이 식(9)을 채우지 않은, 즉, 한 개 챕터 마다의 큰 단위로 검색을 실시하는 엔트리 포인트의 최대 엔트리 포인트 수 NEEDED_EP_COARSE가, 한 개의 플레이 리스트 파일에 관련하는 클립 정보 파일의 큰

단위로 검색을 실시하는 엔트리 포인트의 합계치의 상한치 MAX_EP_COARSE와 추기 후보의 플레이 리스트 파일에 관련하는 클립 정보 파일의 큰 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_COARSE와의 차분보다 크면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0385] 한편, 각 값의 관계가 식(9)를 만족한다고 판정되면, 처리는 다음의 스텝 (S173)으로 이행된다. 스텝 (S173)에서는, 정밀한 단위로 검색을 실시하는 엔트리 포인트에 대해서, 같은 판정이 이루어진다. 즉, 상술의 스텝(S170)에서 취득된, 정밀한 단위로 검색을 실시하는 엔트리 포인트의 한 개 챕터 마다의 최대 엔트리 포인트 수 NEEDED_EP_FINE와 스텝(S171)에서 구해진 정밀한 단위로 검색을 실시하는 엔트리 포인트의 합계 TOTAL_EP_FINE를 이용하여, 큰 단위로 검색을 실시하는 엔트리 포인트의 상한치 MAX_EP_FINE에 대해서, 다음 식(10)에 근거하여 판정이 이루어진다.

[0386] MAX_EP_FINE－TOTAL_EP_FINEE≥NEEDED_EP_FINE.....(10)

[0387] 만약, 각 값의 관계가 이 식(10)을 채우지 않았다고 판정되면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다. 한편, 각 값의 관계가 이 식(10)을 만족한다고 판정되면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다.

[0388] 도 50은, 상술의 스텝(S110)의, 제약(8)에 대응하는, 추기 후보로 여겨진 플레이 리스트 파일내에 있어서의 다른 기기에 의한 고유적인 확장 데이터의 유무에 근거하는 추기 여부 판정의 일례의 처리를 나타낸다.

[0389] 먼저, 최초의 스텝(S180)에서, 추기 후보의 플레이 리스트 파일의 확장 데이터로부터, AVCDH 포맷에 근거하는 확장 데이터를 검색한다.

[0390] 즉, 도 10을 참조하면, 추기 후보의 플레이 리스트 파일의 필드 ExtensionDataStartAddress의 값이 취득되어 취득된 값이 "0"지 아닌지가 판단된다(도시하지 않음). 값이 "0"이 아니면, 해당 플레이 리스트 파일내에 확장 데이터 블록 blkExtensionData()가 존재하므로, 필드 ExtensionDataStartAddress의 값에 근거하여 블록 blkExtensionData()가 참조된다. 그리고, 도 33을 참조하면, 플레이 리스트 파일내의 확장 데이터 블록 blkPlayListExtensionData()에 있어서, 필드 ExtDataType 및 필드 ExtDataVersion가 AVCHD 포맷에 규정되어 있는 값으로 되어 있는지 아닌지를 조사한다.

[0391] 다음의 스텝(S181)에서는, 스텝(S180)의 검색 결과에 근거하여, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재하는지 아닌지가 판단된다.

[0392] 즉, 스텝(S180)의 검색 결과에 근거하여, 추기 후보의 플레이 리스트 파일내의 필드 ExtensionDataStartAddress의 값이 "0"인지, 혹은, 확장 데이터 블록 blkPlayListExtensionData()내의 필드 ExtDataType 및 필드 ExtDataVersion가 AVCHD 포맷에 규정되어 있는 값이 아닌 경우는, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재하지 않는다고 판단된다. 이 경우, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0393] 한편, 스텝(S181)에서, 스텝(S180)의 검색 결과에 근거하여, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재한다고 판단되면, 처리는 스텝(S182)으로 이행된다. 스텝(S182)에서는, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터 이외에, 추가로 확장 데이터가 존재하는지 아닌지가 판단된다. 만약, 존재한다고 판단되면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0394] 스텝(S182)에서, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터 이외의 확장 데이터가 존재하지 않는다고 판단되면, 처리는 스텝(S183)으로 이행된다. 스텝(S183)에서는, 검색된, 추기 후보의 플레이 리스트 파일의 확장 데이터 블록 blkPlayListExtensionData()의 블록 blkMakersPrivateData()가 참조되어 자신의 기기의 데이터가 검색된다. 즉, 도 35를 참조하면, 블록 blkMakersPrivateData()내의 필드 MakerID 및 필드 MakerModelCode에 근거하여, 자신의 기기를 나타내는 데이터가 검색된다.

[0395] 다음의 스텝(S184)에서, 스텝(S183)의 검색 결과에 근거하는 판단이 이루어진다. 스텝(S184)에서는, 스텝(S183)의 검색 결과에 근거하여, 만약, 자신의 기기의 확장 데이터가 존재하지 않는다고 판단되면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다. 한편, 스텝(S184)에서, 스텝(S183)의 검색 결과에 근거하여 자신의 기기의 확장 데이터가 존재한다고 판단되면, 처리는 스텝(S185)으로 이행된다.

[0396] 스텝(S185)에서는, 스텝(S183)의 검색 결과에 근거하여, 블록 blkMakersPrivateData()내에 자신의 기기 이외의

기기에 의한 확장 데이터가 존재하는지 아닌지가 판단된다. 즉, 블록 blkMakersPrivateData()내의 필드 MakerID 및 필드 MakerModelCode에, 자신의 기기를 나타내는 데이터 이외의 데이터가 존재하면, 블록 blkMakersPrivateData()내에 자신의 기기 이외의 기기에 의한 확장 데이터가 존재한다고 판단할 수 있다.

[0397] 스텝(S185)에서, 블록 blkMakersPrivateData()내에 자신의 기기 이외의 기기에 의한 확장 데이터가 존재한다고 판단되면, 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다. 한편, 블록 blkMakersPrivateData() 내에 자신의 기기에 의한 확장 데이터만이 존재한다고 판단되면, 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다.

[0398] 한편, 상술의 스텝(S181)에서는, 추기 후보의 플레이 리스트 파일에 AVCHD 포맷에 근거하는 확장 데이터가 존재하지 않는 경우에, 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가한 것으로 되었지만, 이것은 이 예로 한정되지 않는다. 즉, 기록기의 사양에 따라서는, 추기 후보의 플레이 리스트 파일에, 확장 데이터 블록 blkMakersPrivateData가 전혀 존재하지 않는 경우에, 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하게 하는 일도 생각할 수 있다.

[0399] 도 51은, 상술한 스텝(S111)의, 제약(9)에 대응하는, 추기 후보의 플레이 리스트 파일의 최종 갱신자에 근거하는 추기 여부 판정의 일례의 처리를 나타낸다. 추기 후보의 플레이 리스트 파일의 최종 갱신자가 자신의 기기가 아닌 경우에는, 추기 후보의 플레이 리스트 파일에 있어서의 타이틀이나 챕터의 개념이 자신의 기기와는 다르며, 재생시에 불편함이 생길 가능성이 있다. 추기 후보의 플레이 리스트 파일의 최종 갱신자 정보에 근거하여, 최종 갱신자가 자신의 기기인 경우에게만, 추기 후보의 플레이 리스트 파일에 대해서 챕터의 추기를 하도록 제어하는 것으로서, 한 개의 플레이 리스트 파일내에서의 타이틀이나 챕터의 개념의 차이에 의한 불편함을 회피하는 것이 가능하다.

[0400] 먼저, 최초의 스텝(S190)에서, 추기 후보의 플레이 리스트 파일의 확장 데이터로부터, AVCDH 포맷에 근거하는 확장 데이터를 검색한다. 그리고, 다음의 스텝 (S191)에서, 스텝(S190)의 검색 결과에 근거하여, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재하는지 아닌지가 판단된다. 한편, 이러한 스텝(S190) 및 스텝(S191)의 처리는, 도 50을 이용하여 설명한 스텝(S180) 및 스텝(S181)의 처리와 동일하며, 번잡함을 피하기 위해서 상세한 설명을 생략한다.

[0401] 스텝(S191)에서, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재하지 않는다고 여겨졌을 경우, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0402] 한편, 스텝(S191)에서, 스텝(S190)의 검색 결과에 근거하여, 추기 후보의 플레이 리스트 파일의 확장 데이터에, AVCHD 포맷에 근거하는 확장 데이터가 존재한다고 판단되면, 처리는 스텝(S192)으로 이행된다. 스텝(S192)에서는, 추기 후보의 플레이 리스트 파일의 최종 갱신자가 확인된다. 즉, 확장 데이터 블록 blkPlayListExtensionData()의 블록 blkPlayListMeta()가 참조되어 필드 MakerID 및 필드 MakerModelCode의 데이터가 취득된다.

[0403] 스텝(S192)에서 추기 후보의 플레이 리스트 파일의 최종 갱신자가 확인되면, 처리는 스텝(S193)으로 이행되어, 확인된 최종 갱신자가 자기인지 아닌지가 판단된다. 즉, 확장 데이터 블록 blkPlayListExtensionData()의 블록 blkPlayListMeta()에 있어서의 필드 MakerID 및 필드 MakerModelCode의 데이터가 자신의 기기를 나타내고 있으면, 최종 갱신자가 자신의 기기라고 판단할 수 있다.

[0404] 스텝(S193)에서, 추기 후보의 플레이 리스트 파일의 최종 갱신자가 자신의 기기라고 판단되면, 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 가능하다고 판단한다. 한편, 추기 후보의 플레이 리스트 파일의 최종 갱신자가 자신의 기기가 아니라고 판단하면, 해당 챕터의 추기 후보의 플레이 리스트 파일에 대한 추기가 불가하다고 판단한다.

[0405] 한편, 최종 갱신자의 정보는, 인덱스 파일에도 저장할 수 있다. 예를 들면, 인덱스 파일에 있어서의 확장 데이터 블록 blkIndexExtensionData()내의 블록 blkMakersPrivateData()에 최종 갱신자의 정보를 저장하는 것을 생각한다. 이 경우, 이 인덱스 파일에 저장된 최종 갱신자 정보를 이용하고, 이 도 51의 플로차트에 의한 판단을 실시해도 괜찮다.

[0406] 상술한 제약(10)의, 규정 시간 이상의 연속 촬영 및 기록에 관해서는, 도 43의 스텝(S108) 및 도 48을 이용하여 설명한 클립 정보 파일의 파일 사이즈에 관한 처리, 및, 도 43의 스텝(S109) 및 도 49를 이용하여 설명한 클립

정보 파일에 있어서의 엔트리 포인트에 관한 처리에 포함되므로, 상세한 설명을 생략한다.

[0407] 상술에서는, 도 43을 이용하여 설명한, 추기 후보의 플레이 리스트 파일에 대해서 챕터를 추기 하는지, 신규에 플레이 리스트 파일을 작성하는지의 일련의 판단을, 기록 매체(20)가 기록기에 장전되었을 때에 행해지도록 설명했지만, 이것은 이 예로 한정되지 않는다. 예를 들면, 챕터의 추기마다, 예를 들면 기록 개시 조작마다 판단을 실시하도록 해도 괜찮다.

[0408] 다음에, 상술한 도 43~ 도 51을 이용하여 설명한 처리의 결과, 추기 후보의 플레이 리스트 파일에 대해서 챕터를 추기할 때의 처리에 대해 설명한다. 도 52는, 플레이 리스트 파일에 대해서 챕터를 추기하는 일례의 처리를 나타내는 플로차트이다.

[0409] 스텝(S200)에서 기록 개시 조작을 하면, 다음의 스텝(S201)에서, 클립 AV스트림의 기록 매체(20)에 대한 기록이 개시된다. 예를 들면, UI부(31)에 설치된 기록 개시를 지시하는 기록 개시 스윗치가 조작되는 것으로, 기록 개시를 지시하는 제어 신호가 UI부(31)로부터 제어부(30)에 공급되며, 제어부(30)에 의해, 이 기록 개시를 지시하는 제어 신호에 근거하여, 기록부(10)의 각부에 대해, 단자(40)로부터 입력되는 베이스 밴드의 비디오 데이터와 단자(41)로부터 입력되는 베이스 밴드의 오디오 데이터를 기록 매체(20)에 기록하도록 제어한다.

[0410] 기록 개시의 제어에 대응하여, 클립 AV스트림이 기록 매체(20)에 기록된다(스텝 : S201). 즉, 입력된 비디오 데이터 및 오디오 데이터가 비디오 엔코더(11)및 오디오 엔코더(12)에서 각각 압축 부호화 되며, 멀티플렉서(13)에서 패킷화 된 TS패킷(실제로는 추가로 소정의 헤더가 부가된 소스 패킷)으로 여겨져 스트림 버퍼(14)에 공급된다. 스트림 버퍼(14)에 소정량 이상의 TS패킷이 모아지면, 기록 제어부(15)에 의해 스트림 버퍼(14)로부터 TS패킷이 독출된다. 독출된 TS패킷은, 소정의 파일명이 첨부된 클립 AV스트림 파일에 저장되어 기록 매체(20)에 기록된다.

[0411] 예를 들면, 기록 매체(20)에 이미 파일명 "00001.m2ts"인 클립 AV스트림 파일이 기록되어 있는 경우에는, 새롭게 기록되는 클립 AV스트림 파일의 파일명으로서 이미 기록되어 있는 파일과 중복되지 않는 파일명이 선택되어, 예를 들면 파일명 "00002.m2ts"로 여겨진다.

[0412] 한편, 클립 AV스트림의 기록 매체(20)에의 기록에 수반하여, 관리 정보처리부(16)에 의해, 기록되는 데이터의 재생 시간과 주소와의 대응 관계를 나타내는 정보가 리얼 타임으로 생성된다. 이 데이터는, 상술한 클립 정보 파일"zzzzz.clpi"내의 블록 blkEPMap()에 저장되는 데이터로서 휘발성 메모리(17)에 기억된다.

[0413] 다음의 스텝(S202)에서, 기록 정지 조작을 실행하였는지가 판단된다. 예를 들면, 이용자에 의해 UI부(31)에 설치된 기록 정지 스윗치가 조작되어, 기록이 정지되었다고 판단되면, 처리는 스텝(S203)으로 이행된다. 한편, 기록이 정지되어 있지 않으면, 처리는 스텝(S201)으로 돌아와, 클립 AV스트림의 기록 매체(20)로의 기록이 계속된다.

[0414] 스텝(S203)에서는, 기록의 정지에 수반하여, 스트림 버퍼(14)에 모아진 스트림이 모두 기록 매체(20)에 기입된다. 예를 들면, 기록 제어부(15)는, 제어부(30)로부터의 기록 정지의 명령에 대응하여, 스트림 버퍼(14)에 모아진 스트림(TS패킷)을 모두 독출하고, 기록 매체(20)에 기입한다.

[0415] 또, 기록 정지의 명령에 대응하여, 예를 들면 비디오 엔코더(11) 및 오디오 엔코더(12)의 동작이 정지된다. 이 때, 도 13a를 이용하여 설명한 제 1의 심리스 접속을 실시하기 위해서, 예를 들면, 오디오 엔코더(12)의 동작이 비디오 엔코더(11)의 동작이 정지하고 나서 소정 시간 후에 정지되도록 제어된다.

[0416] 다음의 스텝(S204)~스텝(S208)에서, 관리 정보처리부(16)에 의해, 기록 매체(20)에 기입된 클립 AV스트림 파일에 관한 클립 정보 파일이 생성됨과 동시에, 추기 후보의 플레이 리스트 파일의 갱신이 이루어진다.

[0417] 먼저, 스텝(S204)에서, 관리 정보처리부(16)에 의해, 클립 정보 파일"zzzzz.clpi"가 생성된다. 파일명은, 예를 들면 이 클립 정보 파일이 가리키는 클립 AV스트림 파일의 파일명과 대응하는 파일명으로 되며, 해당 클립 AV스트림 파일의 파일명이 "00002.m2ts"이면, 이 클립 정보 파일의 파일명은, 확장자(extension)보다 이전의 부분이 동일한 파일명 "00002.clpi"로 된다.

[0418] 클립 정보 파일 "00002.clpi"에, 도 15 ~ 도 21에 예시한 각 문맥에 따라, 각 필드나 플래그의 값이 소정의 값으로 설정되어 저장된다. 일례로서, TS패킷에 관한 정보나, 재생 시간(PTS)에 관한 정보는, 관리 정보처리부(16)에 의해, 클립의 기록중에 멀티플렉서(13)로부터 취득된 정보에 근거하여 생성된다. 또, 기록 매체(20)상의 기록 주소에 관한 정보는, 관리 정보처리부(16)에 의해, 클립의 기록중에 기록 제어부(15)로부터 취득된 정보에 근거하여 생성된다. 시스템에 의해 고유의 값은, 예를 들면 미리 ROM(도시하지 않음) 등에 기억되어

있는 정보에 근거한다. 게다가, 재생 시간과 주소와의 대응 관계를 나타내는 상술한 블록 blkEPMap()의 정보가, 클립 정보 파일 "00002.clpi"의 블록 blkCPI()에 저장된다.

[0419] 또, 블록 blkClipInfo()내의 플래그 IsCC5는, 이용자 조작에 의해 클립의 기록이 정지되었을 경우, 값이 1(바이너리치)로 여겨진다. 그에 따라, 블록 blkClipInfo()내의 if문(도 16참조)으로 나타나는 데이터가 소정의 형태로 설정된다.

[0420] 클립 정보 파일의 작성이 완료하면, 처리는 다음의 스텝(S205)으로 이행한다. 스텝(S205)~스텝(S208)의 처리는, 플레이 리스트 파일에 관한 처리이다. 이 스텝(S205)~스텝(S208)의 처리에 의해, 이미 기록 매체(20)상에 존재하는 플레이 리스트 파일에 대해서, 새롭게 기록된 클립 AV스트림 파일 "00002.m2ts"에 대응하는 플레이 아이템이 추가된다.

[0421] 먼저, 스텝(S205)에서, 플레이 리스트 파일내의 블록 blkPlayItem()에 있어서의 필드 ConnectionCondition의 값이 "5"로 설정되어, 이 클립이 다음의 클립과 제 1의 심리스 접속을 실시한다는 것이 표시된다(도 12참조).
다음에 스텝(S206)에서, 플레이 아이템 파일의 필드 NumberOfPlayItems의 값이 "1"만큼 증가되며, 해당 플레이 리스트에 대해서 플레이 아이템이 한 개 추가되는 것이 표시된다(도 11 참조).

[0422] 다음의 스텝(S207)에서, 블록 blkPlayItem()에 있어서의 필드 ClipInformationFileName, 필드 INTime 및 필드 OUTTime가 각각 설정되며, 클립의 기록에 수반하여 추가되는 블록 blkPlayItem()가 작성된다. 필드 ClipInformationFileName는, 상술의 스텝(S205)에서 작성된 클립 정보 파일의 파일명 "00002.clpi"가 저장된다. 실제로는, 클립 정보 파일의 확장자(extension)는 고정적으로 되어 있으므로, 피리어드(period)의 전의 부분 "00002"가 저장된다. 필드 INTime 및 필드 OUTTime는, 대응하는 클립 AV스트림 파일 "00002.m2ts"에 저장되는 비디오 스트림의 선두 및 종단의 시간을 나타내는 정보이며, 예를 들면 클립 정보 파일 "00002.clpi"내의 블록 blkCPI()에 있어서의 블록 blkEPMap()의 정보에 근거한다.

[0423] 다음의 스텝(S208)에서, 추기 후보의 플레이 리스트 파일내의 블록 blkPlayListMark()에 있어서의 필드 NumberOfPlayListMarks의 값이 "1"만큼 증가되며, 그에 따라 for 루프문내에 추가된 필드 MarkTimeStamp의 값이, 상술의 스텝 (S207)에서 블록 blkPlayItem()에 있어서의 필드 INTime의 값으로 설정된다. 즉, 새롭게 기록된 클립의 선두에, 플레이 리스트 마크가 설정된다. 플레이 리스트 마크가 설정되어, 챕터가 형성된다.
즉, 이에 의해, 추기 후보의 플레이 리스트 파일에 대해서 챕터가 추기된다.

[0424] 이와 같이 하여, 새롭게 기록된 클립 AV스트림 파일 "00002.m2ts"에 대해서, 클립 정보 파일 "00002.clpi"가 작성됨과 동시에, 추기 후보의 플레이 리스트 파일의 갱신이 이루어진다. 또 이 때, 플레이 리스트 파일에 있어서의 확장 데이터 블록 blkPlayListExtensionData()내의 블록 blkPlayListMeta()의 정보를 갱신하도록 해도 괜찮다.

[0425] 한편, 상술한 스텝(S203)에 의한 스트림 버퍼(14)에 모아진 데이터의 기록 매체(20)로의 기입은, 스텝(S208)의 처리의 뒤에 실시하도록 해도 괜찮다.

[0426] 플레이 리스트 파일을 신규로 작성하여 챕터를 형성하는 경우에는, 상술의 처리 가운데, 스텝(S205) 이하의 처리가 약간, 다르게 된다. 즉, 플레이 리스트 파일에 있어서의 각 필드의 데이터는, 각각 신규로 생성되게 된다. 이것에 한정하지 않고, 예를 들면 플레이 리스트 파일의 템플릿을 준비해 두고, 템플릿의 데이터를 변경하는 일도 생각할 수 있다.

[0427] 다음에, 이 발명의 실시의 한 형태의 다른 예에 대해 설명한다. 상술에서는, 이 발명이 단체의 기록 장치에 적용된 예에 대해 설명했다(도 41 참조). 이에 대해, 이 실시의 한 형태의 다른 예에서는, 이 발명을, 촬상 소자와 피사체로부터의 빛을 촬상 소자에 입사 키는 광학계를 가지며, 촬상 소자로 촬상된 촬상 신호에 근거하여 비디오 데이터를 기록 매체에 기록하도록 하는, 비디오 카메라 장치에 적용했다.

[0428] 도 53은, 이 발명의 실시의 한 형태의 다른 예에 의한 비디오 카메라 장치(100)의 일례의 구성을 나타낸다. 비디오 카메라 장치(100)에 있어서, 기록계의 구성은, 도 41을 이용하여 설명한 기록 장치의 구성을 거의 그대로 적용할 수 있으므로, 도 41과 공통되는 부분에는 동일한 부호를 교부하고, 상세한 설명을 생략한다.

[0429] 도 53의 구성에 있어서, 카메라부(50)는, 영상 신호에 관한 구성으로서 광학계(51), 촬상 소자(52), 촬상 신호 처리부(53), 카메라 제어부(54) 및 표시부(55)를 가지며, 음성 신호에 관한 구성으로서 마이크로폰(MIC)(56) 및 음성 신호 처리부(57)를 가진다. 제어부(30)는, 카메라부(50)의 각 부와의 사이에 각종 제어 신호나 정보의 교환을 실시하고, 카메라부(50)의 동작을 제어한다. 또, 제어부(30)는, 이용자 조작에 따라 UI부(31)로부

터 공급되는 제어 신호에 근거하여, 카메라부(50)의 동작을 제어한다.

[0430] 한편 비디오 카메라 장치(100)로서 구성되는 경우, 기록 개시 조작 및 기록 정지 조작은, 예를 들면, UI부(31)에 설치된 단일의 기록 스윗치를 이용하여 해당 기록 스윗치가 가압될 때마다 기록 개시 및 기록 정지가 교대로 지시받도록 되는 것이 일반적이다. 또, 이 비디오 카메라 장치(100)에서는, 기록 매체(20)로서 기록 가능한 타입의 DVD나 Blu-rayDisc라고 하는, 디스크 기록 매체를 적용하는 것으로 한다.

[0431] 카메라부(50)에 있어서, 광학계(51)는, 피사체로부터의 빛을 촬상 소자(52)로 유도하는 렌즈계, 조임 조정 기구, 포커스 조정 기구, 줌 기구, 셔터 기구 등을 갖춘다. 조임 조정 기구, 포커스 조정 기구, 줌 기구 및 셔터 기구의 동작은, 제어부(30)로부터 공급되는 제어 신호에 근거하여, 카메라 제어부(54)에 의해 제어된다.

[0432] 촬상 소자(52)는, 예를 들면 CCD(Charge Coupled Device)로 구성되어, 광학계(51)를 통해 조사된 빛을 광전 변환에 의해 전기신호로 변환하고, 소정의 신호 처리를 실시하고 촬상 신호로서 출력한다. 촬상 신호 처리부(53)는, 촬상 소자로부터 출력된 촬상 신호에 대해서 소정의 신호 처리를 실시하고, 베이스 밴드의 디지털 비디오 데이터로서 출력한다.

[0433] 예를 들면 촬상 신호 처리부(53)는, 촬상 소자(52)로부터 출력된 촬상 신호에 대해서, CDS(Correlated Double Sampling)회로에 의해 화상 정보를 가지는 신호만을 샘플링 함과 동시에, 노이즈를 제거하여, AGC(Auto Gain Control)회로에 의해 게인을 조정한다. 그리고, A/D변환에 의해 디지털 신호로 변환한다. 또, 촬상 신호 처리부(53)는, 이 디지털 신호에 대해서 검파계의 신호 처리를 실시하여, R(적색), G(녹색) 및 B(청색) 각 색의 성분을 추출하고, γ보정이나 화이트 밸런스 보정 등의 처리를 실시하여, 최종적으로 한 개의 베이스 밴드의 디지털 비디오 데이터로서 출력한다.

[0434] 또, 촬상 신호 처리부(53)는, 촬상 소자(52)로부터 출력된 촬상 신호의 정보를 제어부(30)에 보낸다. 제어부(30)는, 이 정보에 근거하여 광학계(51)를 제어하기 위한 제어 신호를 생성하고, 카메라 제어부(54)에 공급한다. 카메라 제어부(54)는, 이 제어 신호에 근거하여 포커스 조정 기구나 조임 조정 기구 등의 제어를 실시한다.

[0435] 게다가, 촬상 신호 처리부(53)는, 촬상 소자(52)로부터 출력된 촬상 신호에 근거하여, 예를 들면 LCD(Liquid Crystal Display)를 표시 소자로서 이용한 표시부 (55)로 송출하여 영상 신호를 생성한다.

[0436] 한편, 마이크로폰(56)은, 주위의 음성을 수음 해 전기신호로 변환해 출력한다. 마이크로폰(56)로부터 출력된 음성 신호는, 음성 신호 처리부(57)에 공급된다. 음성 신호 처리부(57)는, 공급된 음성 신호를, 리미터(limiter)를 통해 A/D변환을 실시하여 디지털 오디오 데이터로 변환하고, 노이즈 제거나 음질 보정 등 소정의 음성 신호 처리를 실시하여 베이스 밴드의 디지털 오디오 데이터로서 출력한다.

[0437] 카메라부(50)의 촬상 신호 처리부(53)로부터 출력된 베이스 밴드의 디지털 비디오 데이터는, 기록부(10)의 단자(40)에 공급된다. 또, 음성 신호 처리부(57)로부터 출력된 베이스 밴드의 디지털 오디오 데이터는, 기록부(10)의 단자(41)에 공급된다.

[0438] 촬영시에, 기록 매체(20)가 비디오 카메라 장치(100)에 소정의 형태로 장전되면, 도 43을 이용하여 설명한 처리에 따라, 촬영해 얻어진 비디오 데이터에 근거하는 챕터의 추기 후보의 플레이 리스트 파일이 특정됨과 동시에, 특정된 추기 후보의 플레이 리스트 파일에 대해서 챕터를 추기 가능한가 아닌가가 판단된다.

[0439] 예를 들면, 제어부(30)의 제어에 근거하여 기록 매체(20)에 기록되어 있는 인덱스 파일이 독출되며, 관리 정보 처리부(16)를 통해 휘발성 메모리(17)에 기억된다. 제어부(30)은, 휘발성 메모리(17)에 기억된 인덱스 파일의 정보에 근거하여 추기 후보의 플레이 리스트 파일을 특정하고, 기록 제어부(15)에 대해서, 해당 추기 후보의 플레이 리스트 파일을 기록 매체(20)로부터 독출하라는 명령을 내린다. 이 명령에 근거하여 기록 매체(20)로부터 독출된 추기 후보의 플레이 리스트 파일은, 관리 정보처리부(16)를 통해 휘발성 메모리(17)에 기억된다.

[0440] 제어부(30)는, 휘발성 메모리(17)에 기억된 추기 후보의 플레이 리스트 파일의 정보에 근거하여, 기록 제어부(15)에 대해서, 해당 추기 후보의 플레이 리스트 파일에 관련하는 모든 클립 정보 파일을 기록 매체(20)로부터 독출하라는 명령을 내린다. 이 명령에 근거하여 기록 매체(20)로부터 독출된 클립 정보 파일은, 휘발성 메모리(17)에 기억된다. 제어부(30)는, 휘발성 메모리(17)에 기억된 추기 후보의 플레이 리스트 파일과 해당 추기 후보의 플레이 리스트 파일에 관련하는 모든 클립 정보 파일과에 근거하여, 도 43의 플로차트에 있어서의 스텝(S104)~스텝 (S111)까지의 각 판단을 실시한다. 각 판단의 결과는, 각각, 예를 들면 제어부(30)의 레지스터에 보관 유지된다.

[0441] 제어부(30)는, 스텝(S104)~스텝 (S111)의 각 판단이 종료하면, 도 43의 스텝 (S112)의 처리에 의해, 스텝(S104)~스텝 (S111)의 각 판단의 판단 결과에 대해서 종합적인 판단을 실시하고, 촬영되어 얻어진 비디오 데이터에 근거하는 챕터를 추기 후보의 플레이 리스트 파일에 추기하는지 안하는지를 판단한다. 제어부(30)는, 이 판단 결과에 근거하여, 추기한다고 판단되었을 경우는, 챕터를 휘발성 메모리(17)에 기억되어 있는 추기 후보의 플레이 리스트 파일에 대해서 추기하고, 추기하지 않는다고 판단되었을 경우는, 신규에 플레이 리스트 파일을 생성하고 이 신규의 플레이 리스트 파일에 대해서 기록하도록, 관리 정보처리부(16)를 제어한다.

[0442] 기록 정지상태로부터 UI부(31)에 설치된 기록 스윗치가 가압되면, 기록 개시를 지시하는 제어 신호가 UI부(31)로부터 제어부(30)에 공급되며, 제어부(30)의 제어에 근거하여 카메라부(50)로부터 출력된 베이스 밴드의 디지털 비디오 신호 및 디지털 오디오 데이터가 기록 매체(20)에 기록되는 것이 개시된다.

[0443] 즉, 이미 설명한 것처럼, 제어부(30)의 제어에 근거하여 비디오 엔코더(11) 및 오디오 엔코더(12)의 동작이 개시되며, 비디오 데이터 및 오디오 데이터가 각각 비디오 엔코더(11) 및 오디오 엔코더(12)에서 부호화되고 멀티플렉서(13)에서 소정의 형태로 패킷화되고 다중화 되어 AV스트림 데이터로 된다. AV스트림 데이터는, 스트림 버퍼(14)를 통해, 기록 제어부(15)에 공급되며, 클립 AV스트림 파일로서 기록 매체(20)에 기록된다.

[0444] UI부(31)의 기록 스윗치가 다시 가압되면, 기록이 정지되어 클립 정보 파일의 작성이나, 플레이 리스트 파일의 갱신을 한다. 관리 정보처리부(16)는, 멀티플렉서(13) 및 기록 제어부(15)로부터의 정보에 근거하여, 기록 매체(20)에 기록되는 클립 AV스트림 파일에 대응하는 클립 정보 파일을 작성함과 동시에, 해당 클립 정보 파일을 참조하는 플레이 아이템을 생성한다.

[0445] 추기 후보의 플레이 리스트 파일에 챕터를 추기하도록 판단되고 있는 경우에는, 해당 추기 후보의 플레이 리스트 파일에 대해서, 생성된 플레이 아이템을 추가하는 동시에 플레이 리스트 마크를 설정하고, 챕터를 형성한다. 추기 후보의 플레이 리스트 파일에 대한 챕터의 추가가 불가하다고 판단되는 경우에는, 신규로 작성된 플레이 리스트 파일에 대해서, 생성된 플레이 아이템의 추가 및 플레이 리스트 마크의 설정을 실시한다.

[0446] 이 상태로부터 추가로 한번 더 기록 스윗치가 가압되면, 다시 기록 개시가 지시받아 새로운 클립 AV스트림 파일이 기록 매체(20)로 기록 개시된다. 이 재차의 기록 개시 시에, 도 43의 플로차트에 따라 다시 추기 후보의 플레이 리스트 파일에 대한 새로운 기록에 근거하는 챕터의 추기 여부 판단을 실시하면 좋다.

[0447] 한편, 상술에서는, 도 41에 도시한 기록 장치나 도 53에 도시한 비디오 카메라 장치(100)의 기록부(10)가 하드웨어적으로 구성되도록 설명했지만, 이것은 이 예로 한정되지 않는다. 즉, 기록부(10)는, 소프트웨어로서 구성하는 것도 가능하다. 이 경우, 소프트웨어는, 예를 들면 제어부(30)가 가지는 도시되지 않는 ROM에 미리 기억된다. 이것에 한정하지 않고, 기록부(10)를, 퍼스널 컴퓨터 등의 컴퓨터 장치상에 구성하는 것도 가능하다. 이 경우에는, 기록부(10)를 컴퓨터 장치에 실행시키는 소프트웨어는, CD-ROM나 DVD-ROM이라고 하는 기록 매체에 기록되어 제공된다. 컴퓨터 장치가 네트워크 접속 가능한 경우, 인터넷 등의 네트워크를 통해 해당 소프트웨어를 제공하는 것도 가능하다.

산업상 이용 가능성

[0448] 상술한 바와 같이, 제 1, 제 2 및 제 3의 발명은, 비디오 데이터 및 오디오 데이터가 다중화된 스트림으로 구성되는 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하도록 되며, 기존의 재생 관리 정보에 근거하여, 비디오 데이터의 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해서 추기 하는지 아닌지의 추기 여부 판단을 실시하도록 하기 위해, 기록 개시 조작 및 기록 정지 조작을 반복하여 진행한 경우에도, 이용자가 의식하지 않고, 제 2의 관리 정보의 생성이 적절히 제어된다.

[0449] 또, 제 4, 제 5 및 제 6의 발명은, 촬상되어 얻어진 비디오 데이터와 수음되어 얻어진 오디오 데이터가 다중화된 스트림으로 구성되는 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하도록 되며, 기존의 재생 관리 정보에 근거하여, 조작부에 대한 이용자 조작에 대응하는 기록 개시의 지시에 따라, 기록 매체에 기록되는 피사체를 촬영해 얻어진 비디오 데이터로 구성되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해 추기 하는지 아닌지의 추기 여부 판단을 실시하도록 하기 위해, 촬영에 임하여 기록 개시 조작 및 기록 정

지 조작을 반복하여 실시한 경우에도, 이용자가 의식하지 않고, 제 2의 관리 정보의 생성이 적절히 제어된다.

[0450] 제 1, 제 2 및 제 3의 발명은, 상술한 바와 같이, 비디오 데이터 및 오디오 데이터가 다중화된 스트림으로 구성되는 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와, 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하도록 되며, 기존의 재생 관리 정보에 근거하여, 비디오 데이터의 기록 개시의 지시에 따라 기록 매체에 기록되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해서 추기 하는지 아닌지의 추기 여부 판단을 실시하도록 하기 위해, 기록 개시 조작 및 기록 정지 조작을 반복하여 실시한 경우에도, 이용자가 의식하지 않고, 제 2의 관리 정보의 생성이 적절히 제어되는 효과가 있다.

[0451] 또, 제 4, 제 5 및 제 6의 발명은, 상술한 바와 같이, 촬상되어 얻어진 비디오 데이터와 수음되어 얻어진 오디오 데이터가 다중화된 스트림으로구성되는 스트림 파일의 속성 정보를 나타내는 제 1의 관리 정보와 스트림 파일의 재생 방법을 나타내는 정보를 포함하는 제 2의 관리 정보로 구성되며, 기록 매체에 기록되는 스트림 파일의 재생을 제어하기 위한 재생 관리 정보를 생성하도록 되며, 기존의 재생 관리 정보에 근거하여, 조작부에 대한 이용자 조작에 대응하는 기록 개시의 지시에 따라, 기록 매체에 기록되는 피사체를 촬영하여 얻어진 비디오 데이터로 구성되는 스트림 파일에 대한 재생 방법을 나타내는 정보를, 기존의 재생 관리 정보에 포함되는 소정의 제 2의 관리 정보에 대해 추기 하는지 아닌지의 추기 여부 판단을 실시하도록 하기 위해, 촬영에 임하여 기록 개시 조작 및 기록 정지 조작을 반복하여 실시한 경우에도, 이용자가 의식하지 않고, 제 2의 관리 정보의 생성이 적절히 제어되는 효과가 있다.

도면의 간단한 설명

[0018] 도 1은, 이 발명에 적용 가능한 AVCHD 포맷에 규정되는 데이터 모델을 개략적으로 가리키는 개략도이다.

[0019] 도 2는, 인덱스 테이블을 설명하기 위한 개략도이다

[0020] 도 3은, 클립 AV스트림, 클립 정보, 클립, 플레이 아이템 및 플레이 리스트의 관계를 나타내는 UML도이다.

[0021] 도 4는, 복수의 플레이 리스트로부터 동일한 클립을 참조하는 방법을 설명하기 위한 개략도이다.

[0022] 도 5는, 기록 매체에 기록되는 파일의 관리 구조를 설명하기 위한 개략도이다.

[0023] 도 6은, 파일 "index.bdmv"의 일례의 구조를 나타내는 문맥(syntax)을 나타내는 개략도이다.

[0024] 도 7은, 블록 blkIndexes()의 일례의 구조를 나타내는 문맥을 나타내는 개략 도이다.

[0025] 도 8은, 파일 "MovieObject.bdmv"의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0026] 도 9는, 블록 blkMovieObjects()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0027] 도 10은, 플레이 리스트 파일 "xxxxx.mpls"의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0028] 도 11은, 블록 blkPlayList()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0029] 도 12는, 블록 blkPlayItem()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0030] 도 13a 및 도 13b는, 제 1 및 제 2의 심리스(seamless) 접속을 설명하기 위한 개략도이다.

[0031] 도 14는, 블록 blkPlayListMark()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0032] 도 15는, 클립 정보 파일의 일례의 구조를 나타내는 문맥을 나타내는 개략 도이다.

[0033] 도 16은, 블록 blkClipInfo()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0034] 도 17은, 블록 blkProgramInfo()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0035] 도 18은, 블록 blkStreamCodingInfo(stream_index)의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0036] 도 19는, 필드 VideoFormat로 표시되는 비디오 데이터의 일례의 포맷을 일람으로 나타내는 개략도이다.

[0037] 도 20은, 필드 FrameRate로 표시되는 일례의 프레임 속도를 일람으로 나타내는 개략도이다.

[0038] 도 21은, 필드 AspectRatio로 표시되는 일례의 프레임 속도를 일람으로 나타내는 개략도이다.

[0039] 도 22는, 블록 blkCPI()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0040] 도 23은, 블록 blkEPMap()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0041] 도 24는, 블록 blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf)의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0042] 도 25는, 엔트리 PTSEPCoarse 및 엔트리 PTSEPFine의 일례의 포맷에 대해 표시한 개략도이다.

[0043] 도 26은, 엔트리 SPNEPCoarse 및 엔트리 SPNEPFine의 일례의 포맷에 대해 표시한 개략도이다.

[0044] 도 27은, 블록 blkExtensionData()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0045] 도 28은, 블록 blkExtensionData()에 있어서의 각 데이터의 참조 관계를 모식적으로 나타내는 개략도이다.

[0046] 도 29는, 블록 blkExtensionData()에 데이터를 기입할 때의 일례의 처리를 나타내는 플로차트이다.

[0047] 도 30은, 블록 blkExtensionData()로부터 확장 데이터를 읽어낼 때의 일례의 처리를 나타내는 플로차트이다.

[0048] 도 31은, 파일 "index.bdmv"내의 필드 blkExtensionData()에 있어서의 블록 DataBlock()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0049] 도 32는, 블록 blkTableOfPlayList()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0050] 도 33은, 플레이 리스트 파일 "xxxxx.mpls"내의 블록 blkExtensionData()에 있어서의 블록 DataBlock()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0051] 도 34는, 블록 blkPlayListMeta()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0052] 도 35는, 플레이 리스트 파일에 있어서의 블록 blkMakersPrivateData()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0053] 도 36은, 클립 정보 파일내의 블록 blkExtensionData()에 있어서의 블록 DataBlock()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0054] 도 37은, 블록 blkProgramInfoExt()의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0055] 도 38은, 블록 blkStreamCodingInfoExt(i,j)의 일례의 구조를 나타내는 문맥을 나타내는 개략도이다.

[0056] 도 39a 및 도 39b는, 가상 재생기의 동작을 개략적으로 가리키는 플로차트이다.

[0057] 도 40은, 가상 재생기의 동작을 개략적으로 가리키는 개략도이다.

[0058] 도 41은, 이 발명의 실시의 한 형태에 적용 가능한 기록 장치의 일례의 구성을 개략적으로 가리키는 블럭도이다.

[0059] 도 42는, 이 발명의 실시의 한 형태에 적용 가능한 일례의 데이터 구조를 나타내는 개략도이다.

[0060] 도 43은, 챕터(chapter)의 플레이 리스트 파일에 추기 여부를 판정하는 일례의 처리를 나타내는 플로차트이다.

[0061] 도 44는, 플레이 아이템 수에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0062] 도 45는, 플레이 리스트 마크 수에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0063] 도 46은, 비디오 속성에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0064] 도 47은, 파일 사이즈에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0065] 도 48은, 추기 후보 플레이 리스트 파일로부터 참조되는 클립 정보 파일의 합계 파일 사이즈에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0066] 도 49는, 추기 후보 플레이 리스트 파일로부터 참조되는 클립 정보 파일에 저장되는 엔트리 포인트의 합계수에 근거하는 추기 여부 판단의 일례의 처리를 나타내는 플로차트이다.

[0067] 도 50은, 다른 기기에 의한 고유적인 확장 데이터의 유무에 근거하는 추기 여부 판정의 일례의 처리를 나타내는 플로차트이다.

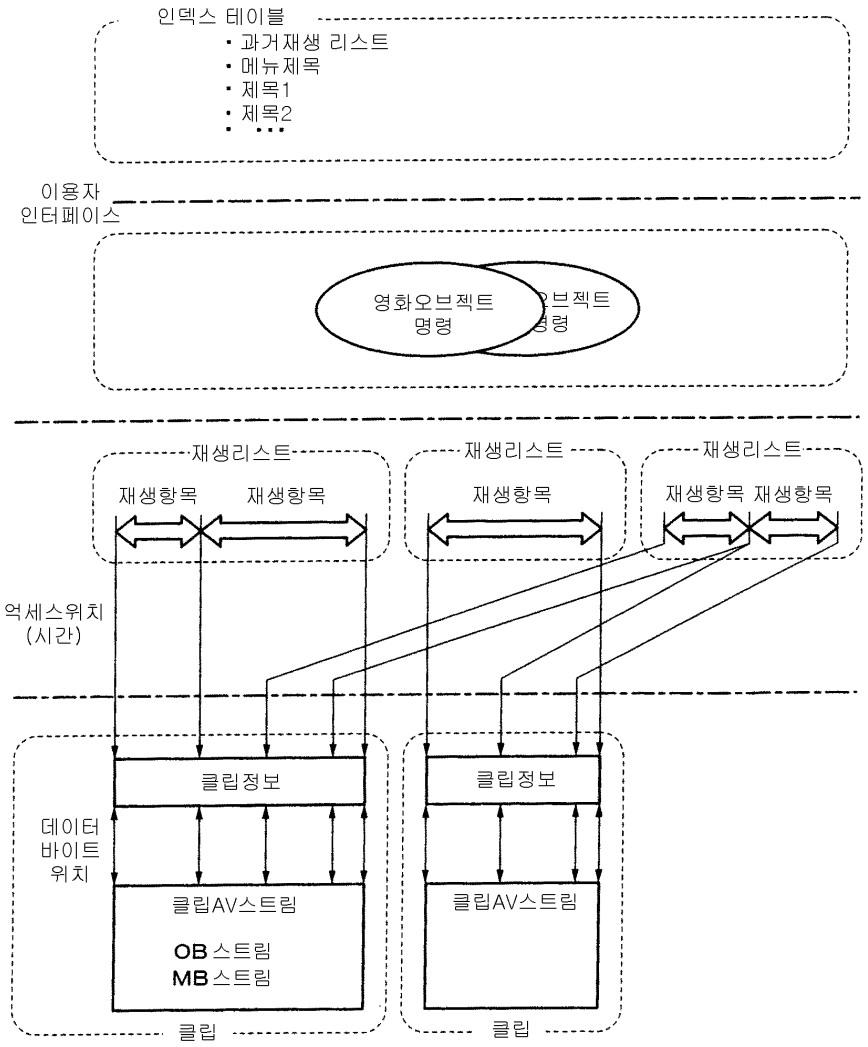
[0068] 도 51은, 추기 후보 플레이 리스트 파일의 최종 갱신자에 근거하는 추기 여부 판정의 일례의 처리를 나타내는 플로차트이다.

[0069] 도 52는, 이 발명의 실시의 한 형태에 의한 클립의 일례의 기록 방법을 나타내는 플로차트이다.

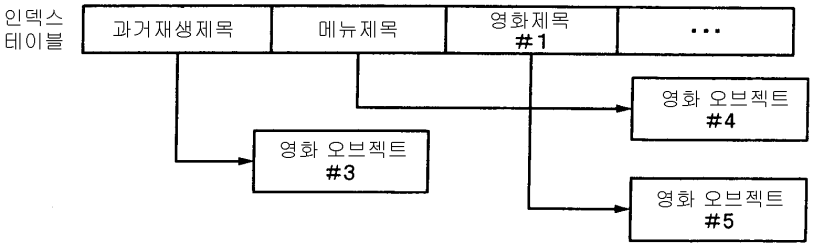
[0070] 도 53은, 이 발명의 실시의 한 형태의 다른 예에 의한 비디오 카메라 장치의 일례의 구성을 나타내는 블럭도이다.

도면

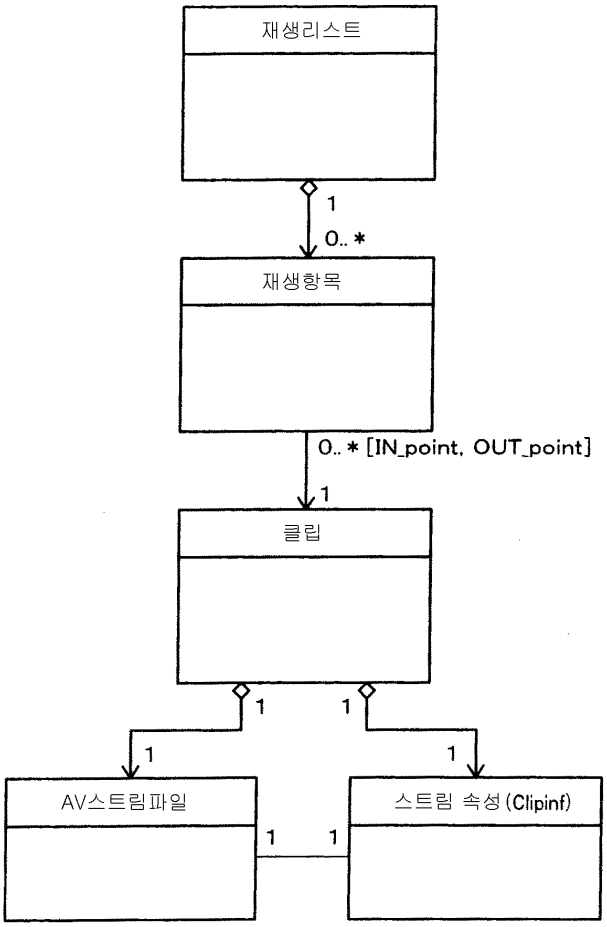
도면1



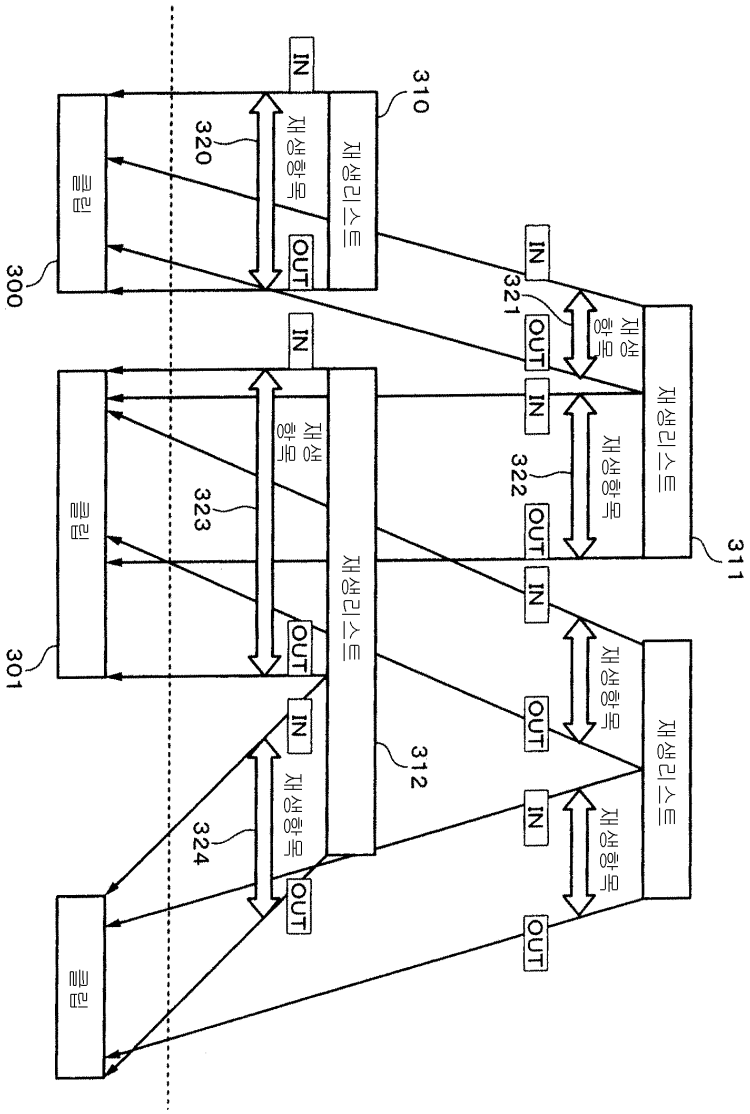
도면2


인덱스
테이블
과거재생제목
메뉴제목
영화제목
#1
...
영화 오브젝트
#4
영화 오브젝트
#3
영화 오브젝트
#5

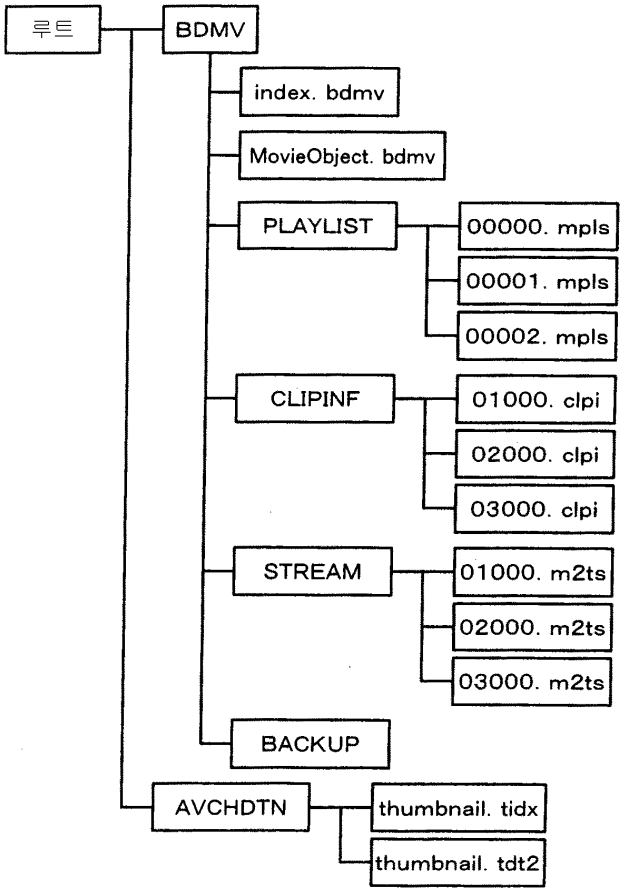
도면3


재생리스트
1
0..*
재생항목
0..* [IN_point, OUT_point]
1
클립
1
1
1
1
AV스트림파일
스트림 속성(Clipinf)
1
1

도면4


310
재생리스트
IN
재생항목
OUT
320
클립
300
311
재생리스트
321
322
323
301
312
324

도면5



도면6

신택스	데이터길이 (비트)	니모닉
Index table file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
IndexesStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	192	bslbf
blkAppInfoBDMV()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkIndexes()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
}		

도면7

신택스	데이터길이 (비트)	니모닉
blkIndexes() {		
Length	32	uimsbf
FirstPlaybackTitle() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	31	bslbf
'1'	1	bslbf
reserved	14	bslbf
FirstPlaybackTitleMobjIDRef	16	uimsbf
reserved	32	bslbf
}		
MenuTitle() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	31	bslbf
'1'	1	bslbf
reserved	14	bslbf
MenuTitleMobjIDRef	16	uimsbf
reserved	32	bslbf
}		
NumberOfTitles	16	uimsbf
for (*title_id*=0; *title_id* < NumberOfTitles; *title_id*++) {		
MovieTitle[*title_id*]() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	46	bslbf
MovieTitleMobjIDRef*[title_id]*	16	uimsbf
reserved	32	bslbf
}		
}		
}		

도면8

신택스	데이터길이 (비트)	니모닉
MovieObject file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
ExtensionDataStartAddress	32	uimsbf
reserved	224	bslbf
blkMovieObjects()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
}		

도면9

신택스	데이터길이 (비트)	니모닉
blkMovieObjects() {		
Length	32	uimsbf
reserved	32	bslbf
NumberOfMobjs	16	uimsbf
for (mobj_id=0; mobj_id<NumberOfMobjs; mobj_id++) {		
MovieObject*[mobj_id]*() {		
TerminalInfo() {		
'1'	1	bslbf
reserved	15	bslbf
}		
NumberOfNavigationCommands*[mobj_id]*	16	uimsbf
for (*command_id*=0; *command_id*<*NumberOfNavigationCommands*[*mobj_id*]; *command_id*++){		
NavigationCommand*[mobj_id][command_id]*	96	bslbf
}		
}		
}		
}		

도면10

신택스	데이터길이 (비트)	니모닉
Movie PlayList file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
PlayListStartAddress	32	uimsbf
PlayListMarkStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	160	bslbf
blkAppInfoPlayList()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkPlayList()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkPlayListMark()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
}		

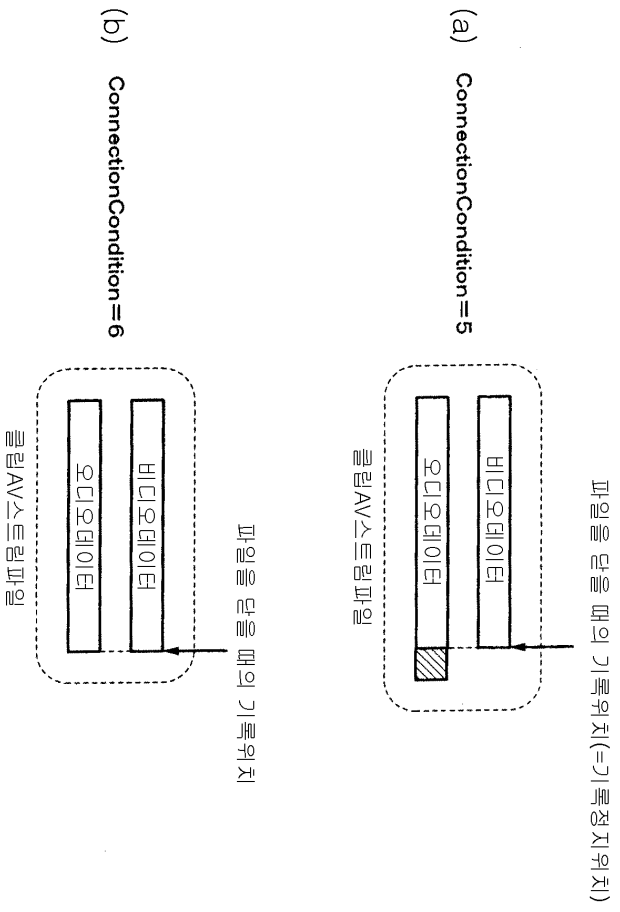
도면11

신택스	데이터길이 (비트)	니모닉
blkPlayList() {		
Length	32	uimsbf
reserved	16	bslbf
NumberOfPlayItems	16	uimsbf
NumberOfSubPaths	16	uimsbf
for(*PlayItem_id*=0; *PlayItem_id*<*NumberOfPlayItems*; *PlayItem_id*++) {		
blkPlayItem()		
}		
for(*SubPath_id*=0; *SubPath_id*<*NumberOfSubPaths*; *SubPath_id*++) {		
blkSubPath()		
}		
}		

도면12

신택스	데이터길이 (비트)	니모닉
blkPlayItem() {		
Length	16	uimsbf
ClipInformationFileName	8 * 5	bslbf
ClipCodecIdentifier	8 * 4	bslbf
reserved	12	bslbf
ConnectionCondition	4	bslbf
RefToSTCID	8	uimsbf
INTime	32	uimsbf
OUTTime	32	uimsbf
blkUOMaskTable()		
PlayItemRandomAccessFlag	1	bslbf
reserved	7	bslbf
StillMode	8	bslbf
if (StillMode== 0x01){		
StillTime	16	uimsbf
} else {		
reserved	16	bslbf
}		
blkSTNTable()		
}		

도면13



도면14

신택스	데이터길이(비트)	니모닉
blkPlayListMark() {		
Length	32	uimsbf
NumberOfPlayListMarks	16	uimsbf
for(*PL_mark_id*=0; *PL_mark_id*< *NumberOfPlayListMarks*; *PL_mark_id*++) {		
reserved	8	bslbf
MarkType	8	bslbf
RefToPlayItemID	16	uimsbf
MarkTimeStamp	32	uimsbf
EntryESPID	16	uimsbf
Duration	32	uimsbf
}		
}		

도면15

신택스	데이터길이 (비트)	니모닉
Clip information file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
SequenceInfoStartAddress	32	uimsbf
ProgramInfoStartAddress	32	uimsbf
CPIStartAddress	32	uimsbf
ClipMarkStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	96	bslbf
blkClipInfo()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkSequenceInfo()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkProgramInfo()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
blkCPI()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
blkClipMark()		
for (i=0; i<N5; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N6; i++) {		
padding_word	16	bslbf
}		
}		

도면16

신택스	데이터길이 (비트)	니모닉
blkClipInfo() {		
Length	32	uimsbf
reserved	16	bslbf
ClipStreamType	8	bslbf
ApplicationType	8	bslbf
reserved	31	bslbf
IsCC5	1	bslbf
TSRecordingRate	32	uimsbf
NumberOfSourcePackets	32	uimsbf
reserved	1024	bslbf
TSTypeInfoBlock()		
if (IsCC5 ==1_b) {		
reserved	8	bslbf
FollowingClipStreamType	8	bslbf
reserved	32	bslbf
FollowingClipInformationFileName	8 * 5	bslbf
ClipCodecIdentifier	8 * 4	bslbf
reserved	8	bslbf
}		
}		

도면17

신택스	데이터길이 (비트)	니모닉
blkProgramInfo() {		
Length	32	uimsbf
reserved	15	bslbf
'1'	1	bslbf
SPNProgramSequenceStart	32	uimsbf
ProgramMapPID	16	uimsbf
NumberOfStreamsInPS	8	uimsbf
reserved	8	bslbf
for (*stream_index*=0; *stream_index*<*NumberOfStreamsInPS*; *stream_index*++) {		
StreamPID[*stream_index*]	16	uimsbf
blkStreamCodingInfo(*stream_index*)		
}		
}		

도면18

신택스	데이터길이 (비트)	니모닉
blkStreamCodingInfo(*stream_index*) {		
Length	8	uimsbf
StreamCodingType	8	bslbf
if (*StreamCodingType*==0x1B) {		
VideoFormat	4	bslbf
FrameRate	4	bslbf
AspectRatio	4	bslbf
reserved	2	bslbf
CCFlag	1	bslbf
reserved	17	bslbf
reserved	128	bslbf
} else if (*StreamCodingType==0x80* || *StreamCodingType==0x81*) {		
AudioPresentationType	4	bslbf
SamplingFrequency	4	bslbf
AudioLanguageCode	8 * 3	bslbf
reserved	128	bslbf
} else if (*StreamCodingType==0x90*) { // Overlay Bitmap stream		
OBLanguageCode	8 * 3	bslbf
reserved	136	bslbf
} else if (*StreamCodingType==0x91*) { // Menu Bitmap stream		
MBLanguageCode	8 * 3	bslbf
reserved	136	bslbf
}		
}		

도면19

VideoFormat	의미	표준화
0	reserved	
1	**480i**	ITU-R BT.601-4[38]
2	**576i**	ITU-R BT.601-4[38]
3	**480p**	SMPTE 293M[39]
4	**1080i**	SMPTE 274M[40]
5	**720p**	SMPTE 296M[41]
6	**1080p**	SMPTE 274M[40]
7	**576p**	ITU-R BT.1358[42]
8-14	reserved	
15	reserved	

도면20

FrameRate	의미 **[Hz]**
0	reserved
1	**24000/1001 (23.976...)**
2	**24**
3	**25**
4	**30000/1001 (29.97...)**
5	reserved
6	**50**
7	**60000/1001 (59.94...)**
8	reserved
9-14	reserved
15	reserved

도면21

AspectRatio	의미
0	reserved
1	reserved
2	표시애스펙트비 **4:3**
3	표시애스펙트비 **16:9**
4	reserved
5-14	reserved
15	reserved

도면22

신택스	데이터길이 (비트)	니모닉
blkCPI() {		
Length	32	uimsbf
if(Length !=0) {		
reserved	12	bslbf
CPIType	4	bslbf
blkEPMap()		
}		
}		

도면23

신택스	데이터길이 (비트)	니모닉
blkEPMap(){		
reserved	8	bslbf
NumberOfStreamPIDEntries	8	uimsbf
for (*k=0*; *k*<*NumberOfStreamPIDEntries*; *k*++) {		
StreamPID*[k]*	16	bslbf
reserved	10	bslbf
EPStreamType*[k]*	4	bslbf
NumberOfEPCoarseEntries*[k]*	16	uimsbf
NumberOfEPFineEntries*[k]*	18	uimsbf
EPMapForOneStreamPIDStartAddress*[k]*	32	uimsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bslbf
}		
for (*k=0*; *k*<*NumberOfStreamPIDEntries*; *k*++) {		
blkEPMapForOneStreamPID(*EPStreamType[k], NumberOfEPCoarseEntries[k], NumberOfEPFineEntries[k]*)		
for (i=0; i<Y[k]; i++) {		
padding_word	16	bslbf
}		
}		
}		

도면24

신택스	데이터길이 (비트)	니모닉
blkEPMapForOneStreamPID(*EP_stream_type*, *Nc*, *Nf*) {		
EPFineTableStartAddress	32	uimsbf
for (i=0; i<*Nc*; i++) {		
// EP coarse table		
RefToEPFineID*[i]*	18	uimsbf
PTSEPCoarse*[i]*	14	uimsbf
SPNEPCoarse*[i]*	32	uimsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bslbf
}		
for (*EP_fine_id* =0; *EP_fine_id* < *Nf*; *EP_fine_id*++) {		
// EP fine table		
ReservedEPFine*[EP_fine_id]*	1	bslbf
IEndPositionOffset*[EP_fine_id]*	3	bslbf
PTSEPFine*[EP_fine_id]*	11	uimsbf
SPNEPFine*[EP_fine_id]*	17	uimsbf
}		
}		

도면25

- PTS :b0.b32 (33비트, 90kHz)
- PTSEPFine :b9.b19 (11비트, 5.7밀리초의 해상도로, 거의 11.5초 랩어라운드한다)
- PTSEPCoarse :b19.b32 (14비트, 5.8밀리초의 해상도로, 거의 26.5시간 랩어라운드한다)

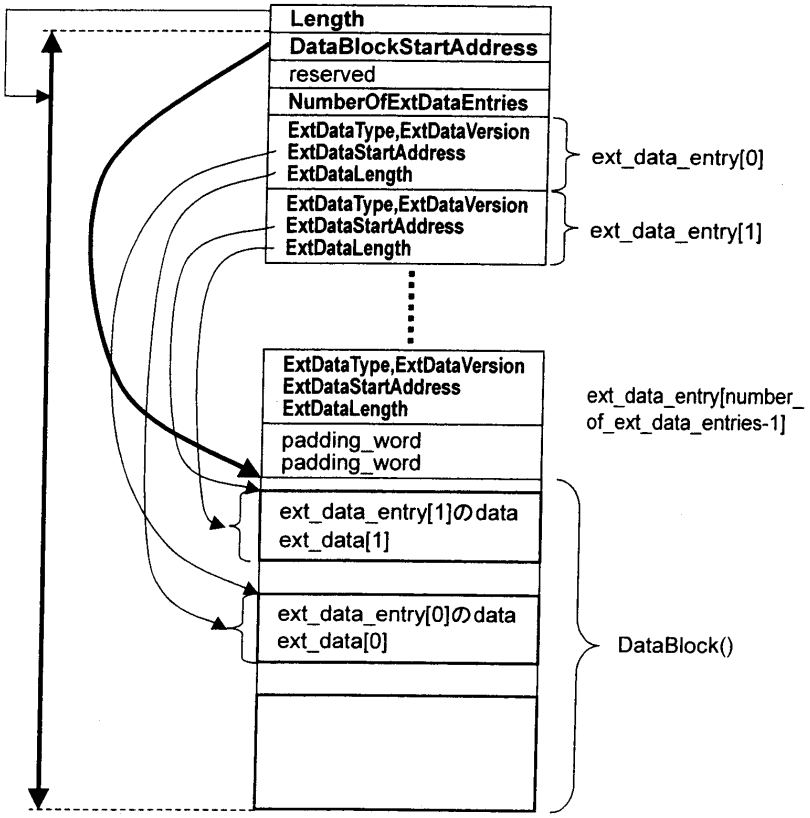
도면26

- **SPNEPStart** :**b0..b31** (32비트)
- **SPNEPFine** :**b0..b16** (17비트, AV스트림 파일중에서 거의 25메가파일로 랩어라운드한다)
- **SPNEPCoarse** :**b0..b31** (32비트)

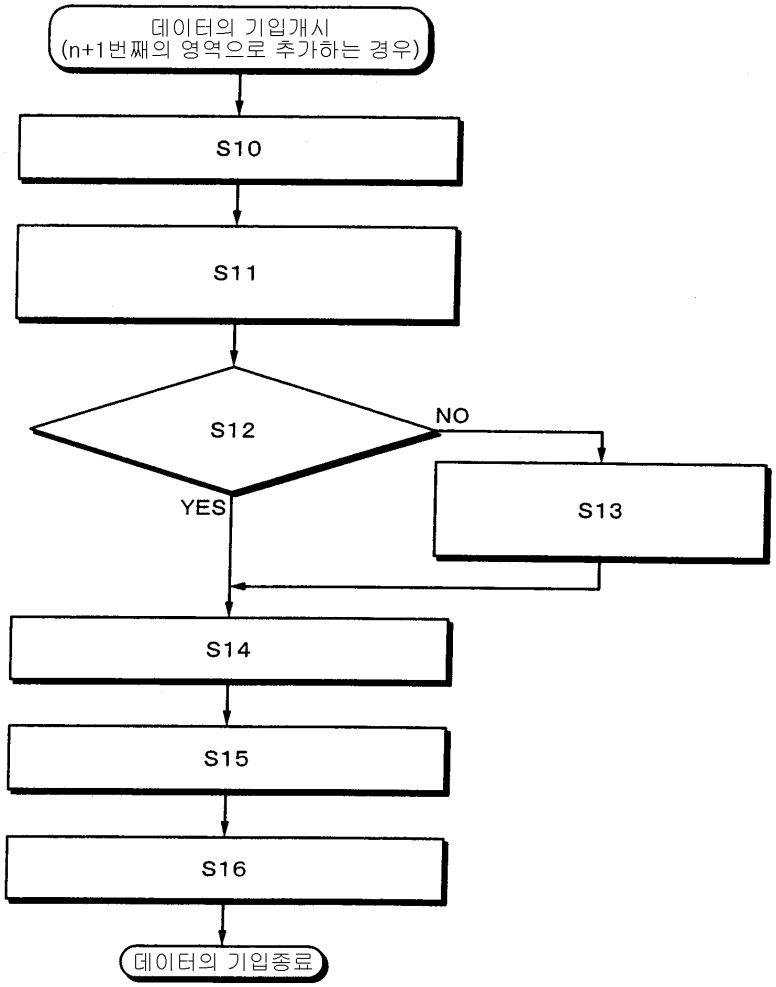
도면27

신택스	데이터길이 (비트)	니모닉
blkExtensionData() {		
Length	32	uimsbf
if(Length !=0){		
DataBlockStartAddress	32	uimsbf
reserved	24	
NumberOfExtDataEntries	8	uimsbf
for (i=0; i<*NumberOfExtDataEntries*; i++) {		
ext_data_entry() {		
ExtDataType	16	uimsbf
ExtDataVersion	16	uimsbf
ExtDataStartAddress	32	uimsbf
ExtDataLength	32	uimsbf
}		
}		
for (i=0; i<L1; i++) {		
padding_word	16	bslbf
padding_word	16	bslbf
}		
DataBlock()	32+ 8 * (Length - DataBlockStartAddress)	
}		
}		

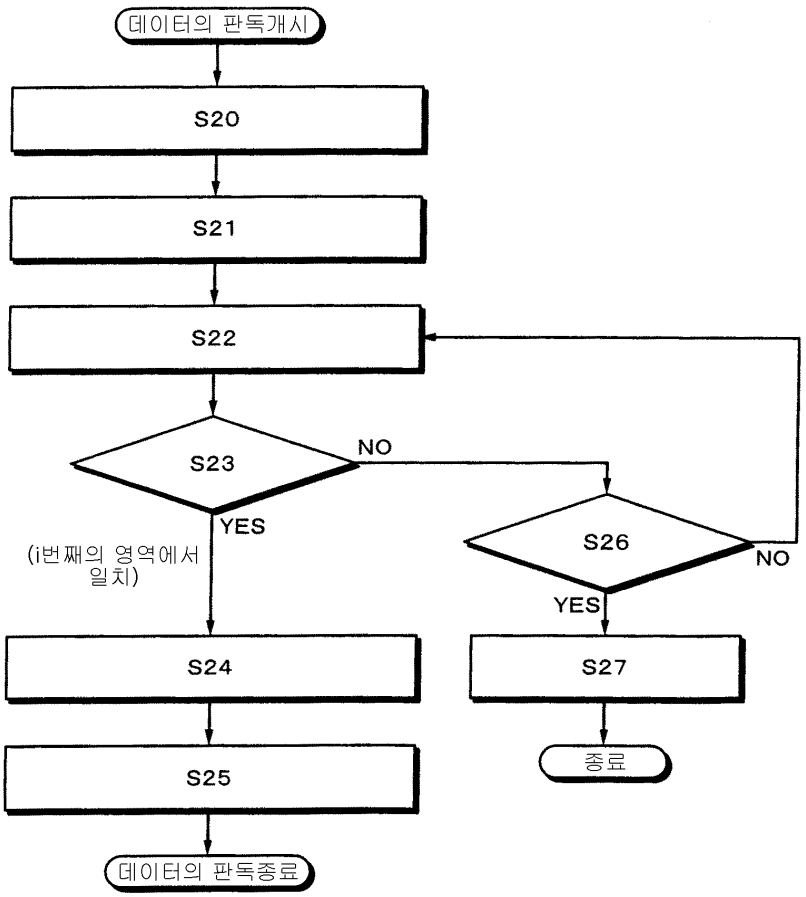
도면28


Length
DataBlockStartAddress
reserved
NumberOfExtDataEntries
ExtDataType,ExtDataVersion
ExtDataStartAddress
ExtDataLength
ext_data_entry[0]
ExtDataType,ExtDataVersion
ExtDataStartAddress
ExtDataLength
ext_data_entry[1]
ExtDataType,ExtDataVersion
ExtDataStartAddress
ExtDataLength
ext_data_entry[number_
of_ext_data_entries-1]
padding_word
padding_word
ext_data_entry[1]のdata
ext_data[1]
ext_data_entry[0]のdata
ext_data[0]
DataBlock()

도면29


데이터의 기입개시
(n+1번째의 영역으로 추가하는 경우)
S10
S11
S12
NO
YES
S13
S14
S15
S16
데이터의 기입종료

도면30


데이터의 판독개시
S20
S21
S22
S23
NO
YES
(i번째의 영역에서 일치)
S26
NO
YES
S24
S27
S25
종료
데이터의 판독종료

도면31

신택스	데이터길이 (비트)	니모닉
blkIndexExtensionData(){		
TypeIndicator	8 * 4	uimsbf
reserved	8 * 4	bslbf
TableOfPlayListsStartAddress	32	uimsbf
MakersPrivateDataStartAddress	32	uimsbf
reserved	192	bslbf
blkUIAppInfoAVCHD()		
for(i=0; i<N1;i++) {		
padding_word	16	bslbf
}		
blkTableOfPlayLists()		
for(i=0; i<N2;i++) {		
padding_word	16	bslbf
}		
blkMakersPrivateData()		
for (i=0; i<N3;i++){		
padding_word	16	bslbf
}		
}		

도면32

신택스	데이터길이 (비트)	니모닉
blkTableOfPlayLists() {		
Length	32	uimsbf
blkFirstPlaybackTitlePlayLists()		
blkMenuTitlePlayLists()		
NumberOfTitlePlayListPair	16	bslbf
for(i=0; i< NumberOfTitlePlayListPair;i++){		
blkMovieTitlePlayListPair() {		
PlayListFileName	8 * 5	bslbf
reserved	6	bslbf
PlayListAttribute	2	uimsbf
reserved	16	bslbf
RefToTitleId	16	uimsbf
}		
}		
}		

도면33

신택스	데이터길이 (비트)	니모닉
blkPlayListExtensionData() {		
TypeIndicator	8 * 4	bslbf
reserved	8 * 4	bslbf
PlayListMarkExtStartAddress	32	uimsbf
MakersPrivateDataStartAddress	32	uimsbf
reserved	192	bslbf
blkPlayListMeta()		
for (i=0; i<N1; i++){		
padding_word	16	bslbf
}		
blkPlayListMarkExt()		
for (i=0; i<N2; i++){		
padding_word	16	bslbf
}		
blkMakersPrivateData()		
for (i=0; i<N3; i++){		
padding_word	16	bslbf
}		
}		

도면34

신택스	데이터길이 (비트)	니모닉
blkPlayListMeta() {		
Length	32	uimsbf
MakerID	16	uimsbf
MakerModelCode	16	uimsbf
reserved	32	bslbf
RefToMenuThumbnailIndex	16	uimsbf
blkTimeZone()	8	uimsbf
RecordTimeAndDate	4 * 14	uimsbf
reserved	8	bslbf
PlayListCharacterSet	8	uimsbf
PlayListNameLength	8	bslbf
PlayListName	8 * 255	uimsbf
Additional_data(){		
Length2	32	uimsbf
reserved	8 * Length2	bslbf
}		
}		

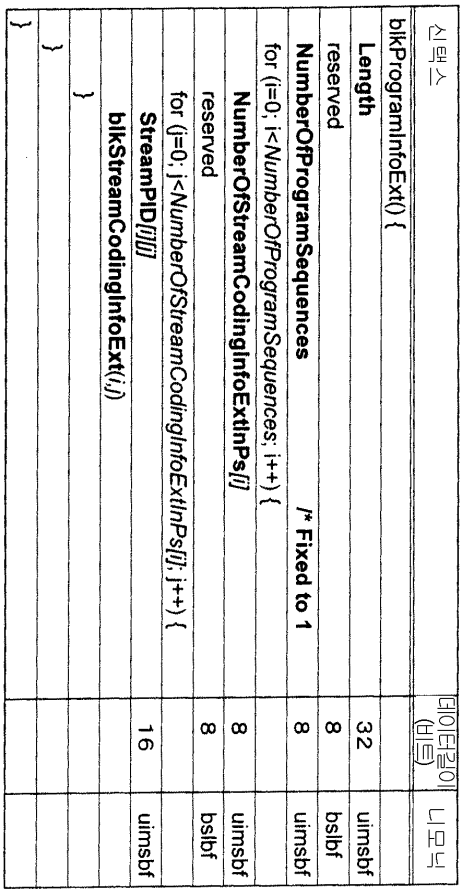
도면35

신택스	데이터길이 (비트)	니모닉
blkMakersPrivateData() {		
Length	32	uimsbf
if(Length !=0){		
DataBlockStartAddress	32	uimsbf
reserved	24	bslbf
NumberOfMakerEntries	8	uimsbf
for (i=0; i<*NumberOfMakerEntries*; i++){		
MakerID	16	uimsbf
MakerModelCode	16	uimsbf
MpdStartAddress	32	uimsbf
MpdLength	32	uimsbf
}		
for (i=0; i<L1; i++) {		
padding_word	16	bslbf
padding_word	16	bslbf
}		
DataBlock()	32+ 8 * (Length - DataBlockStartAddress)	
}		
}		

도면36

신택스	데이터길이 (비트)	니모닉
blkClipExtensionData() {		
TypeIndicator	8 * 4	bslbf
reserved	8 * 4	bslbf
ProgramInfoExtStartAddress	32	uimsbf
MakersPrivateDataStartAddress	32	uimsbf
reserved	192	bslbf
blkClipInfoExt()		
for(i=0; i<N1;i++){		
padding_word	16	bslbf
}		
blkProgramInfoExt()		
for(i=0; i<N2;i++){		
padding_word	16	bslbf
}		
blkMakersPrivateData()		
for(i=0; i<N3;i++){		
padding_word		
}	16	bslbf
}		

도면37

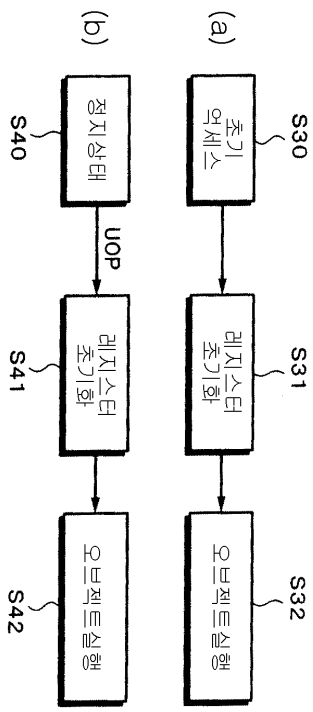


신택스	데이터길이(비트)	니모닉
blkProgramInfoExt() {		
Length	32	uimsbf
reserved	8	bslbf
NumberOfProgramSequences **/* Fixed to 1**	8	uimsbf
for (i=0; i<*NumberOfProgramSequences*; i++) {		
NumberOfStreamCodingInfoExtInPs[*i*]	8	uimsbf
reserved	8	bslbf
for (j=0; j<*NumberOfStreamCodingInfoExtInPs*[*i*]; j++) {		
StreamPID[*i*][*j*]	16	uimsbf
blkStreamCodingInfoExt(*i*,*j*)		
}		
}		
}		

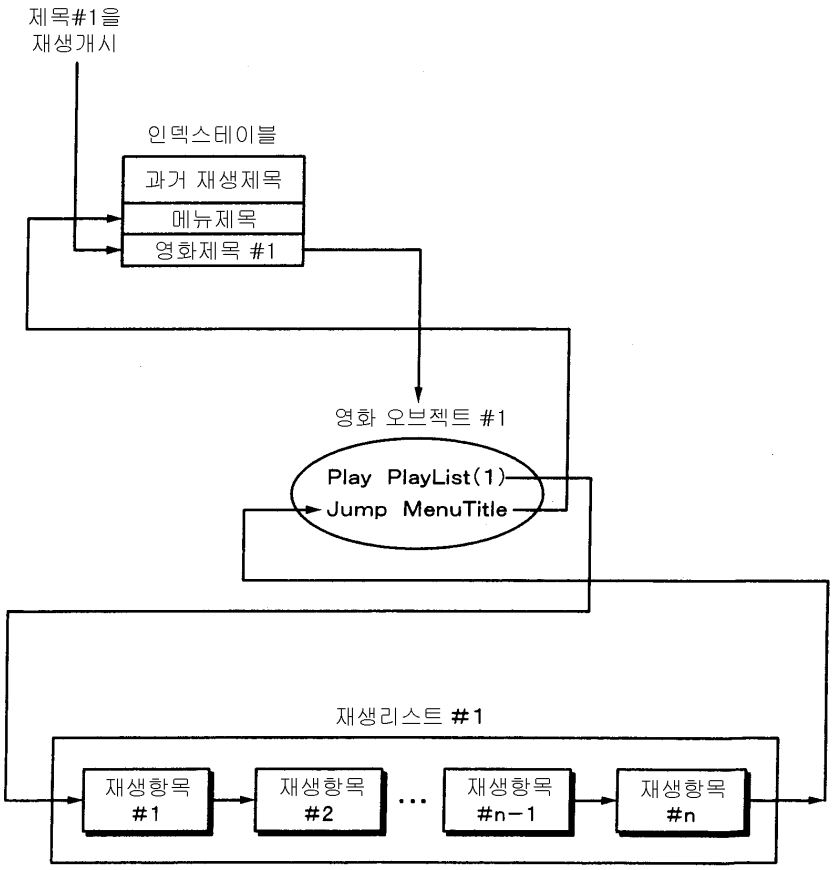
도면38

신택스	데이터길이(비트)	니모닉
blkStreamCodingInfoExt(*i*,*j*) {		
Length	8	uimsbf
StreamCodingType	8	bslbf
if (*StreamCodingType*==0x1B) {		
HorizontalSize	8	uimsbf
profile_idc	8	uimsbf
constraint_set0_flag	1	bslbf
constraint_set1_flag	1	bslbf
constraint_set2_flag	1	bslbf
constraint_set3_flag	1	bslbf
reserved	4	bslbf
level_idc	8	uimsbf
reserved	48	bslbf
}		
}		

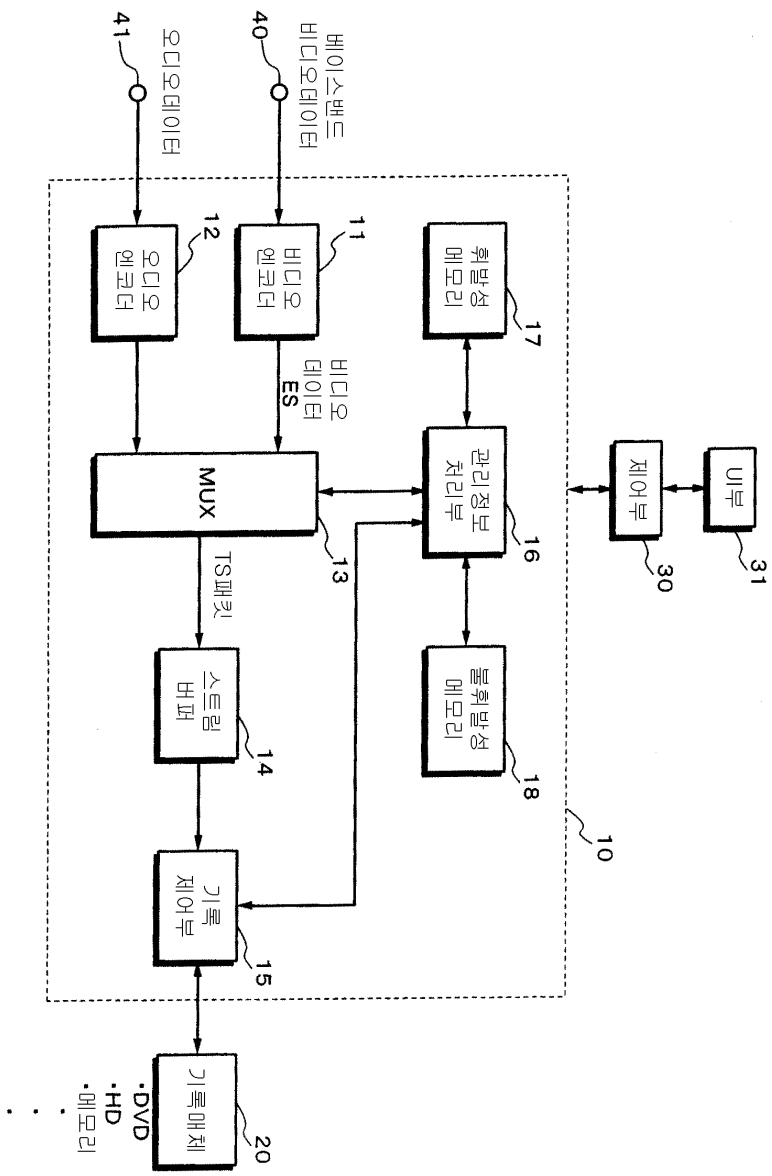
도면39


(a)
초기 억세스
S30
레지스터 초기화
S31
오브젝트실행
S32
(b)
정지상태
S40
UOP
레지스터 초기화
S41
오브젝트실행
S42

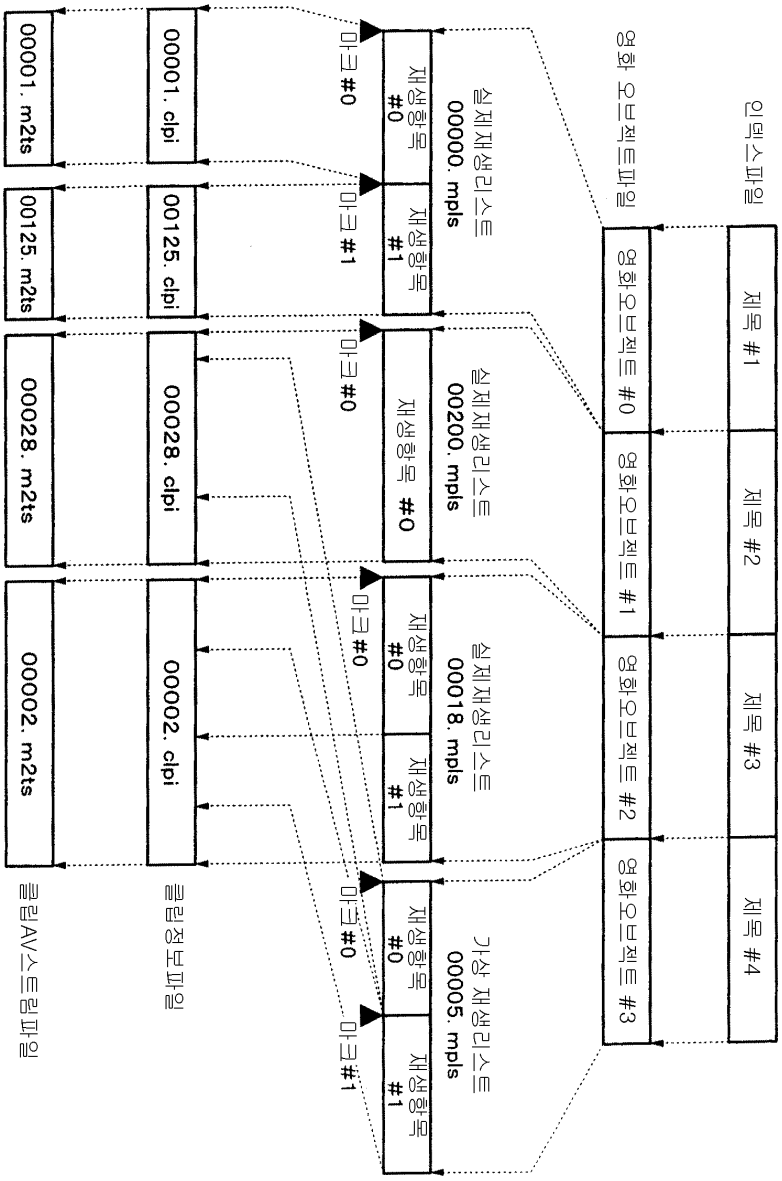
도면40


제목#1을
재생개시
인덱스테이블
과거 재생제목
메뉴제목
영화제목 #1
영화 오브젝트 #1
Play PlayList(1)
Jump MenuTitle
재생리스트 #1
재생항목
#1
재생항목
#2
...
재생항목
#n-1
재생항목
#n

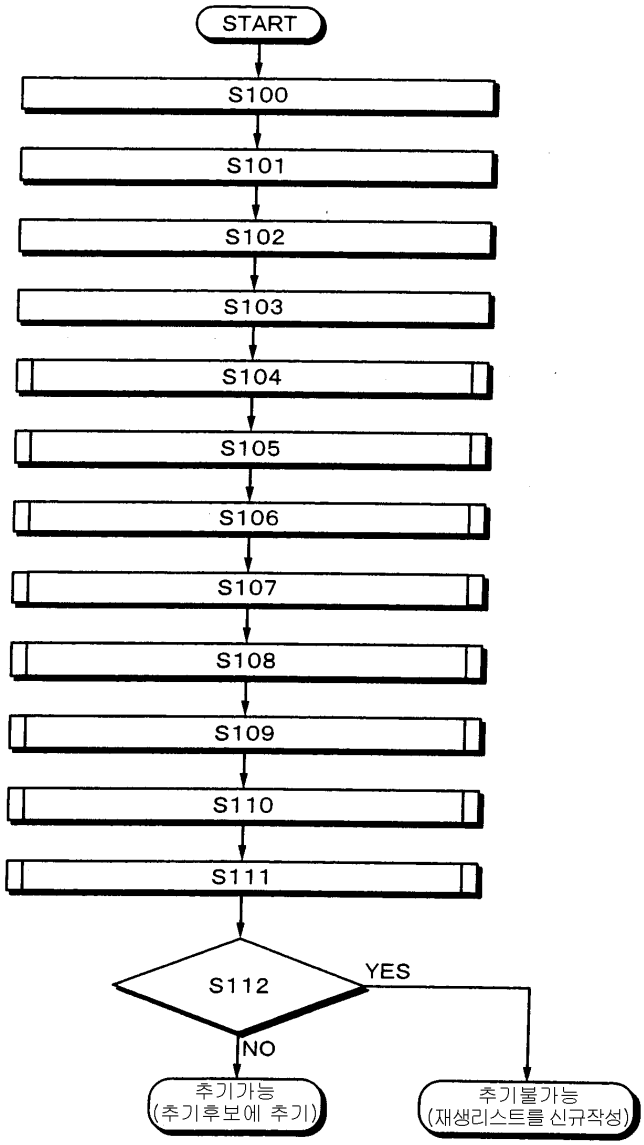
도면41


베이스밴드
비디오데이터
40
오디오데이터
41
비디오
엔코더
11
오디오
엔코더
12
휘발성
메모리
17
비디오
데이터
ES
MUX
13
관리정보
처리부
16
제어부
30
UI부
31
TS패킷
스트림
버퍼
14
불휘발성
메모리
18
10
기록
제어부
15
기록매체
20
·DVD
·HD
·메모리
· · ·

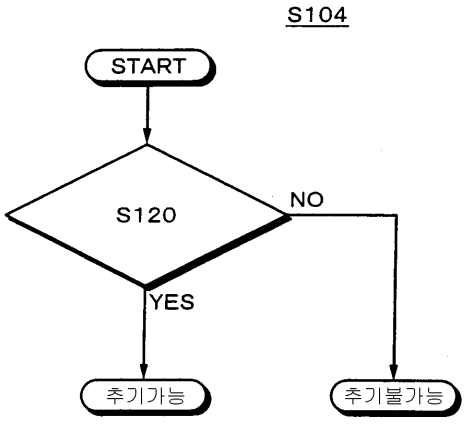
도면42


인덱스파일
제목 #1
제목 #2
제목 #3
제목 #4
영화 오브젝트파일
영화오브젝트 #0
영화오브젝트 #1
영화오브젝트 #2
영화오브젝트 #3
실제재생리스트
00000. mpls
재생항목 #0
재생항목 #1
마크 #0
마크 #1
실제재생리스트
00200. mpls
재생항목 #0
마크 #0
실제재생리스트
00018. mpls
재생항목 #0
재생항목 #1
마크 #0
가상 재생리스트
00005. mpls
재생항목 #0
재생항목 #1
마크 #0
마크 #1
00001. clpi
00125. clpi
00028. clpi
00002. clpi
클립정보파일
00001. m2ts
00125. m2ts
00028. m2ts
00002. m2ts
클립AV스트림파일

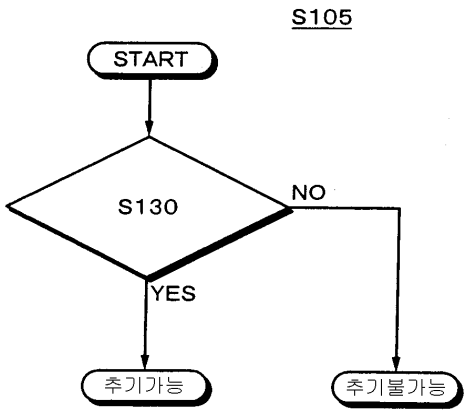
도면43


START
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
YES
NO
추기가능
(추기후보에 추기)
추기불가능
(재생리스트를 신규작성)

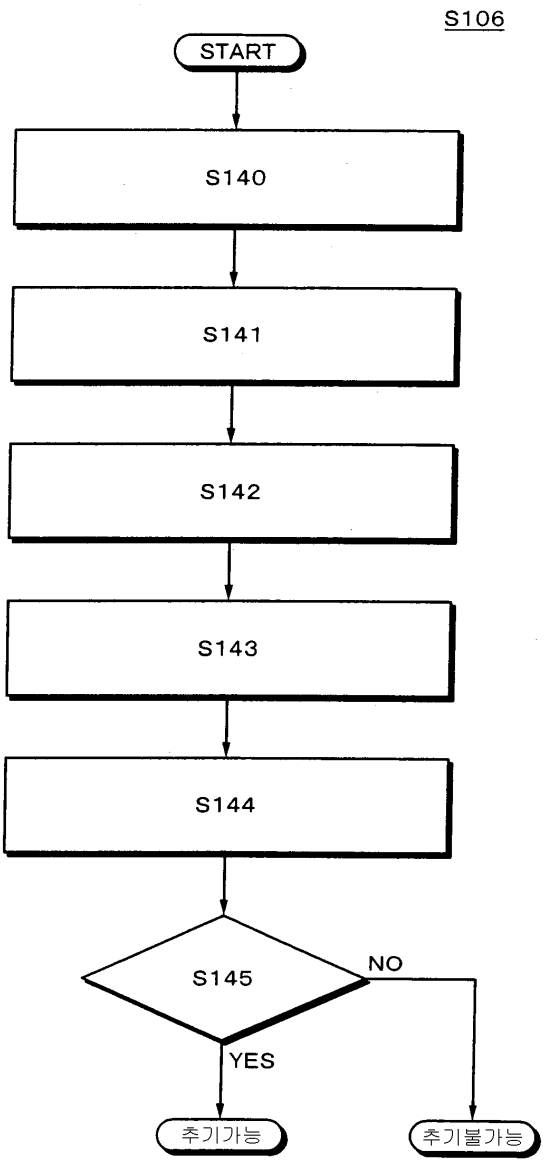
도면44


S104
START
S120
NO
YES
추기가능
추기불가능

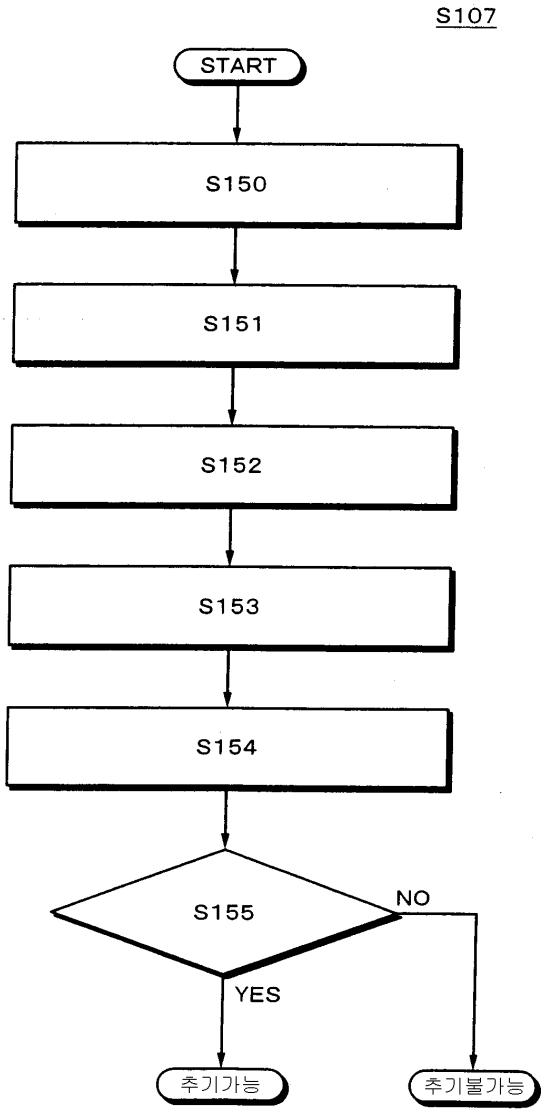
도면45



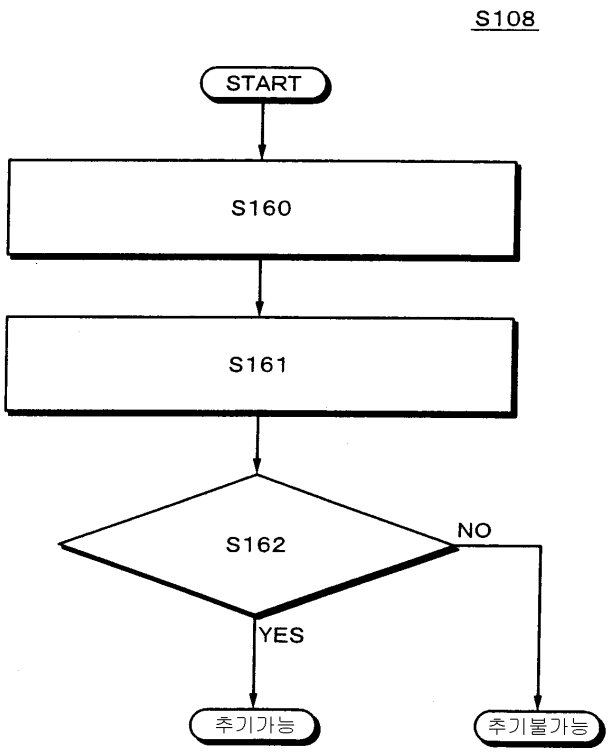
도면46



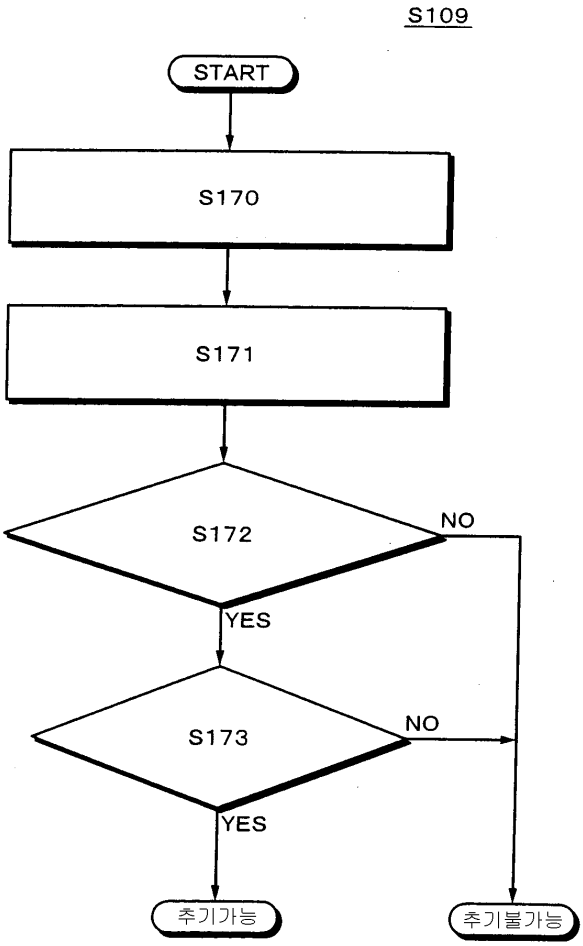
도면47



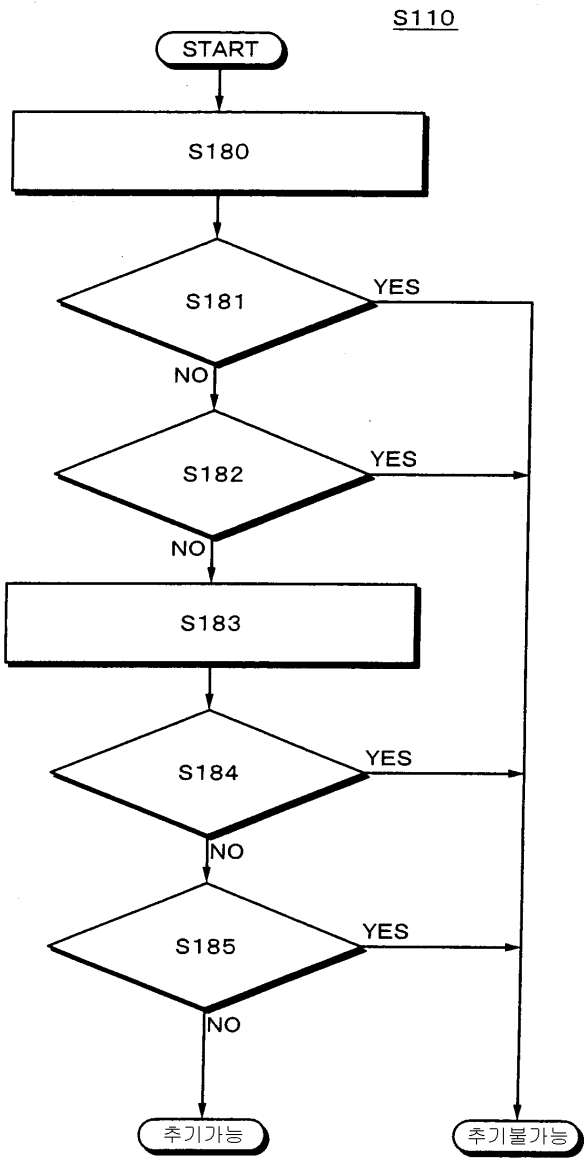
도면48



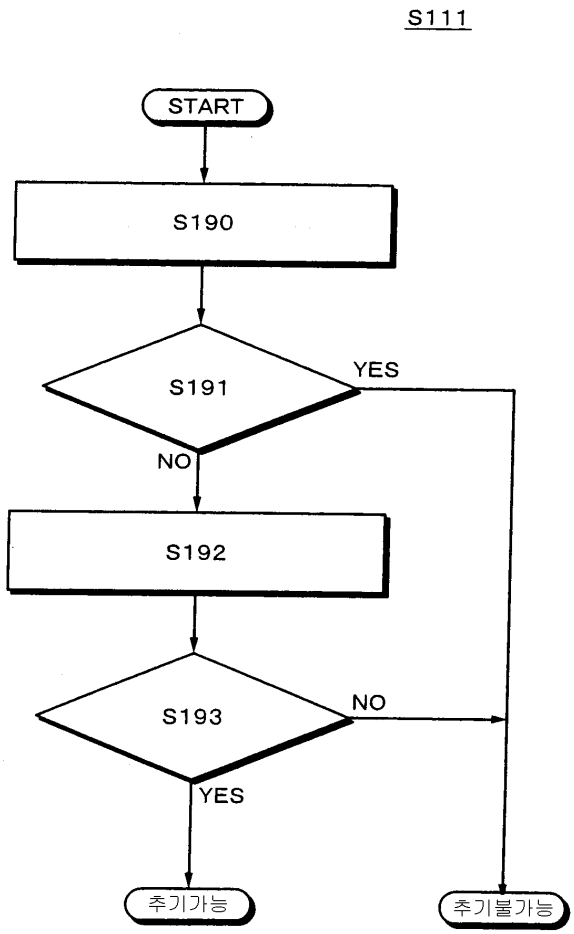
도면49



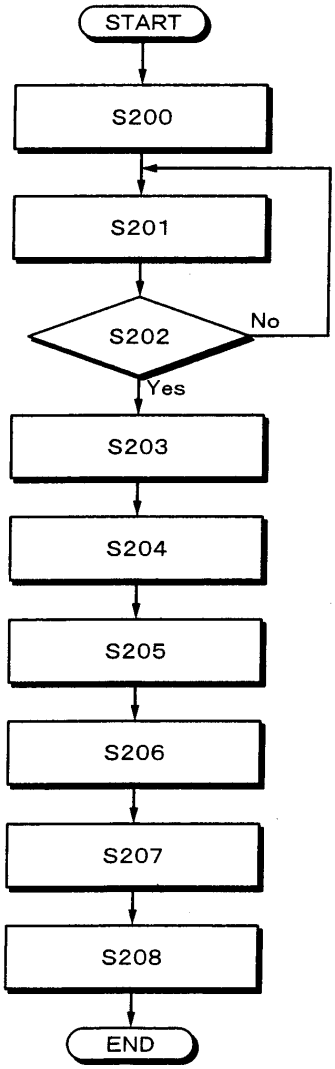
도면50


S110
START
S180
S181
YES
NO
S182
YES
NO
S183
S184
YES
NO
S185
YES
NO
추기가능
추기불가능

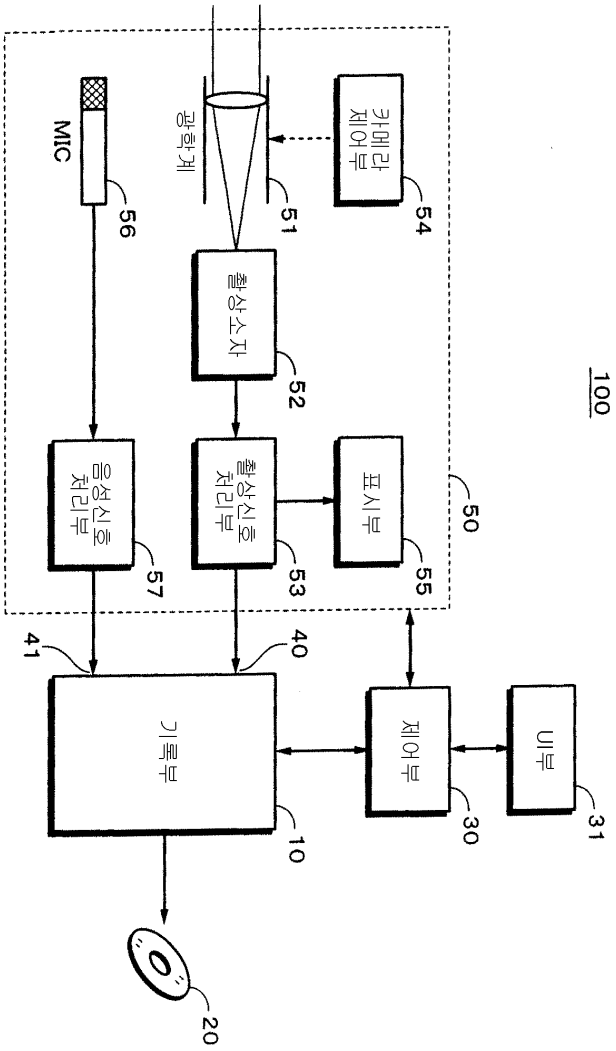
도면51



도면52



도면53


100
50
MIC
56
광학계
51
카메라 제어부
54
촬상소자
52
음성신호 처리부
57
촬상신호 처리부
53
표시부
55
41
40
기록부
10
제어부
30
UI부
31
20