



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2014년02월19일

(11) 등록번호 10-1365339

(24) 등록일자 2014년02월13일

- (51) 국제특허분류(Int. Cl.)
G11B 20/10 (2006.01) *G11B 27/10* (2006.01)
H04N 5/85 (2006.01) *H04N 5/91* (2006.01)
- (21) 출원번호 10-2008-7006244
- (22) 출원일자(국제) 2007년07월25일
 심사청구일자 2012년06월05일
- (85) 번역문제출일자 2008년03월14일
- (65) 공개번호 10-2009-0033161
- (43) 공개일자 2009년04월01일
- (86) 국제출원번호 PCT/JP2007/065030
- (87) 국제공개번호 WO 2008/013308
 국제공개일자 2008년01월31일
- (30) 우선권주장

JP-P-2006-00203037 2006년07월26일 일본(JP)

- (56) 선행기술조사문현

JP2005524191 A*

WO2006033279 A1*

*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
소니 주식회사
 일본국 도쿄도 미나토구 코난 1-7-1
- (72) 발명자
이소베 유키오
 일본국 도쿄도 미나토구 코난 1-7-1 소니 가부시
 끼 가이샤 내
아리도메 켄이치로
 일본국 도쿄도 미나토구 코난 1-7-1 소니 가부시
 끼 가이샤 내
 (뒷면에 계속)

- (74) 대리인
최달용

전체 청구항 수 : 총 16 항

심사관 : 권영학

- (54) 발명의 명칭 기록 장치, 기록 방법 및 기록 프로그램, 및, 활상 장치, 활상 방법 및 활상 프로그램

(57) 요 약

기록 시작부터 정지의 사이에 생성된 AV 데이터를 파일로서 기록할 때에, 장시간의 연속 기록이나 연속 재생을 용이하게 한다. AV 데이터를 다중화한 스트림을 패킷 단위로 파일에 기록한다. 다음의 패킷의 기록에서 파일 사이즈가 2GB를 초과한 경우, 해당 파일의 오디오 데이터의 후단이 다음에 스트림이 기록되는 파일의 오디오 데이터의 선두와 시간적으로 대응하도록 기록 제어하여 해당 파일을 클로즈함과 함께, 클로즈한 파일과 다음에 스트림이 기록되는 파일을, 오디오 데이터의 중복이 없도록 심레스 접속하는 것이 지정된다. 신규로 파일을 작성하여 스트림을 계속적으로 기록한다. 정지 조작으로 파일의 기록을 정지한다. 장시간의 기록에 대해 파일이 자동적으로 분할된다. 재생시에는, 장시간 기록된 파일을 파일의 분할을 의식하는 일 없이 연속 재생할 수 있다.

(72) 발명자

모리모토 나오키

일본국 도쿄도 미나토구 코난 1-7-1 소니 가부시끼
가이샤 내

마에 아츠시

일본국 도쿄도 미나토구 코난 1-7-1 소니 가부시끼
가이샤 내

마에다 테츠히로

일본국 도쿄도 미나토구 코난 1-7-1 소니 가부시끼
가이샤 내

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 시작의 지시 및 기록 정지의 지시가 입력되는 기록 지시 입력부와,

상기 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록부와,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성부와,

상기 기록부 및 상기 관리 정보 생성부를 제어하는 CPU를 포함하는 제어부를 가지며,

상기 제어부는,

상기 기록 지시 입력부에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록부를 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록부를 제어하고,

상기 제어부는, 상기 기록 지시 입력부에 의한 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록부를 제어하는 것을 특징으로 하는 기록 장치.

청구항 2

제 1항에 있어서,

상기 제어부는,

상기 소정의 제약에 의거한 상기 스트림 파일의 클로즈에 따라, 클로즈되는 해당 스트림 파일에 기록되는 상기 오디오 데이터의 후단의 시각과, 상기 신규의 스트림 파일에 격납되는 상기 오디오 데이터의 선단의 시각이 대응하도록 상기 기록부에 의한 기록을 제어함과 함께,

상기 클로즈된 스트림 파일을 재생 구간으로서 지정하는 재생 구간 데이터 내에, 해당 클로즈된 스트림 파일에 기록되는 비디오 데이터 및 오디오 데이터와, 해당 신규의 스트림 파일에 기록되는 비디오 데이터 및 오디오 데이터가 연속적으로 재생되는 것을 나타내는 정보를 설정하여 상기 재생 구간 데이터를 상기 기록 매체에 기록하도록 상기 기록부를 제어하는 것을 특징으로 하는 기록 장치.

청구항 3

제 1항에 있어서,

상기 스트림 파일의 속성은, 파일 사이즈이고,

상기 소정의 제약은, 해당 파일 사이즈의 상한인 것을 특징으로 하는 기록 장치.

청구항 4

제 1항에 있어서,

상기 스트림 파일의 속성은, 상기 스트림 정보 파일에 격납되는, 상기 스트림 파일에 격납되는 상기 비디오 데이터의 시작을 나타내는 정보와, 해당 스트림 파일 내의 어드레스를 관련시키는 엔트리 포인트 정보이고, 상기 소정의 제약은, 하나의 상기 스트림 정보 파일에 대해 격납 가능하게 된 상기 엔트리 포인트 정보 수의 상한인 것을 특징으로 하는 기록 장치.

청구항 5

제 1항에 있어서,

상기 스트림 파일의 속성은, 상기 스트림 파일에 격납되는 상기 비디오 데이터의 시작을 나타내는 정보이고, 상기 소정의 제약은, 상기 시작을 나타내는 정보가 표현 가능한 시작의 상한인 것을 특징으로 하는 기록 장치.

청구항 6

제 1항에 있어서,

상기 제어부는, 기록된 상기 스트림 파일을 재생 구간으로서 지정하는 상기 재생 구간 데이터 내에, 해당 스트림 파일과 해당 스트림 파일에 대해 후속하는 스트림 파일이 프레임 타이밍에서 연속적으로 재생되는 것을 나타내는 정보를 설정하여 상기 재생 구간 데이터를 상기 기록 매체에 기록하도록 상기 기록부를 제어하는 것을 특징으로 하는 기록 장치.

청구항 7

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

데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터의 기록 시작의 지시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과,

상기 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과,

상기 기록의 스텝 및 상기 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며,

상기 제어의 스텝은,

상기 기록 지시 입력의 스텝에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록의 스텝을 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록의 스텝을 제어하고,

상기 제어의 스텝은, 상기 기록 지시 입력의 스텝에 의한 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록의 스텝을 제어하는 것을 특징으로 하는 기록 방법.

청구항 8

비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 방법을 컴퓨터 장치에 실행시키는 기록 프로그램이 기록된 기록매체에 있어서,

상기 기록 방법은,

데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터의 기록 시작의 지시 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과,

상기 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 차례로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과,

상기 기록의 스텝 및 상기 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며,

상기 제어의 스텝은,

상기 기록 지시 입력의 스텝에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록의 스텝을 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록의 스텝을 제어하고,

상기 제어의 스텝은, 상기 기록 지시 입력의 스텝에 의한 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록의 스텝을 제어하는 것을 특징으로 하는 기록 프로그램이 기록된 기록매체.

청구항 9

촬상부에서 피사체를 촬상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음하여 얻어진 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 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록부와,

상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에의 기록 시작의 지시 및 기록 정지의 지시를 위한 유저의 조작을 접수하는 조작부와,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성부와,

상기 기록부 및 상기 관리 정보 생성부를 제어하는 CPU를 포함하는 제어부를 가지며,

상기 제어부는,

상기 조작부에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록부를 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록부를 제어하고,

상기 제어부는, 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재

생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록부를 제어하는 것을 특징으로 하는 활상 장치.

청구항 10

제 9항에 있어서,

상기 제어부는,

상기 소정의 제약에 의거하는 상기 스트림 파일의 클로즈에 따라, 클로즈되는 해당 스트림 파일에 기록되는 상기 오디오 데이터의 후단의 시각과, 상기 신규의 스트림 파일에 격납되는 상기 오디오 데이터의 선단의 시각이 대응하도록 상기 기록부에 의한 기록을 제어함과 함께,

상기 클로즈되는 스트림 파일을 재생 구간으로서 지정하는 재생 구간 데이터 내에, 해당 클로즈되는 스트림 파일에 기록되는 비디오 데이터 및 오디오 데이터와, 해당 신규의 스트림 파일에 기록되는 비디오 데이터 및 오디오 데이터가 연속적으로 재생되는 것을 나타내는 정보를 설정하여 상기 재생 구간 데이터를 상기 기록 매체에 기록하도록 상기 기록부를 제어하는 것을 특징으로 하는 활상 장치.

청구항 11

제 9항에 있어서,

상기 스트림 파일의 속성은, 파일 사이즈이고,

상기 소정의 제약은, 해당 파일 사이즈의 상한인 것을 특징으로 하는 활상 장치.

청구항 12

제 9항에 있어서,

상기 스트림 파일의 속성은, 상기 스트림 정보 파일에 격납되는, 상기 스트림 파일에 격납되는 상기 비디오 데이터의 시각을 나타내는 정보와, 해당 스트림 파일 내의 어드레스를 관련시키는 엔트리 포인트 정보이고,

상기 소정의 제약은, 하나의 상기 스트림 정보 파일에 대해 격납 가능하게 된 상기 엔트리 포인트 정보 수의 상한인 것을 특징으로 하는 활상 장치.

청구항 13

제 9항에 있어서,

상기 스트림 파일의 속성은, 상기 스트림 파일에 격납되는 상기 비디오 데이터의 시각을 나타내는 정보이고,

상기 소정의 제약은, 상기 시각을 나타내는 정보가 표현 가능한 시각의 상한인 것을 특징으로 하는 활상 장치.

청구항 14

제 9항에 있어서,

상기 제어부는, 기록된 상기 스트림 파일을 재생 구간으로서 지정하는 상기 재생 구간 데이터 내에, 해당 스트림 파일과 해당 스트림 파일에 대해 후속하는 스트림 파일이 프레임 타이밍에서 연속적으로 재생되는 것을 나타내는 정보를 설정하여 상기 재생 구간 데이터를 상기 기록 매체에 기록하도록 상기 기록부를 제어하는 것을 특징으로 하는 활상 장치.

청구항 15

활상부에서 피사체를 활상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음하여 얻어진 오디오 데이터를 다중화하여 기록 매체에 기록하는 활상 장치의 활상 방법에 있어서,

피사체를 활상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 차례로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과,

조작부에 대한 상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에의 기록 시작의 지시 및 기록 정지의 지시를 위한 유저의 조작을 접수하는 스텝과,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함으로써 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과,

상기 기록의 스텝 및 상기 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며,

상기 제어의 스텝은,

상기 조작부에 대한 조작에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록의 스텝을 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록의 스텝을 제어하고,

상기 제어의 스텝은, 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록의 스텝을 제어하는 것을 특징으로 하는 활상 방법.

청구항 16

활상부에서 피사체를 활상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음하여 얻어진 오디오 데이터를 다중화하여 기록 매체에 기록하는 활상 장치의 활상 방법을 컴퓨터 장치에 실행시키는 활상 프로그램이 기록된 기록매체에 있어서,

상기 활상 방법은,

피사체를 활상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 해당 패킷마다 차례로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과,

조작부에 대한 상기 비디오 데이터 및 상기 오디오 데이터의 상기 기록 매체에의 기록 시작의 지시 및 기록 정지의 지시를 위한 유저의 조작을 접수하는 스텝과,

상기 기록 매체에 기록되는 상기 스트림 파일에 대해, 적어도, 해당 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 해당 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 상기 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과,

상기 기록의 스텝 및 상기 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며,

상기 제어의 스텝은,

상기 조작부에 대한 조작에 의한 상기 기록 시작의 지시 및 상기 기록 정지의 지시에 따라 상기 스트림의 상기 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 상기 기록의 스텝을 제어하고,

하나 또는 복수의 상기 패킷으로 이루어지는 소정 단위를 상기 스트림 파일에 기록함으로써 해당 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 해당 스트림 파일을 클로즈하여 신규의 상기 스트림 파일을 작성하고, 해당 신규의 스트림 파일에 대해 상기 스트림을 해당 소정 단위로부터 차례로, 기록하도록 상기 기록의 스텝을 제어하고,

상기 제어의 스텝은, 상기 기록 정지의 지시에 의거하여, 상기 스트림 파일 내에서, 상기 비디오 데이터의 선단의 재생시각과 상기 오디오 데이터의 선단의 재생시각이 일치하고, 상기 비디오 데이터의 종단의 재생시각보다도 상기 오디오 데이터의 종단의 재생시각이 후가 되도록 상기 기록의 스텝을 제어하는 것을 특징으로 하는 활상 프로그램이 기록된 기록매체.

명세서

기술분야

[0001] 본 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.

배경기술

[0002] 종래에는, 예를 들면 비디오 카메라 등에서, 촬상 소자로 촬상된 영상 신호를 디지털 비디오 데이터로 변환하고, 압축 부호화하여 기록하는 경우의 기록 매체로서, 자기 테이프가 이용되고 있다. 자기 테이프는, 대용량의 데이터를 연속적으로 기록하는데 적합하였다.

[0003] 한편, 근래에는, 기록 가능하고 기록 재생 장치로부터 페어냅이 가능하게 됨과 함께, 기록 용량이 비교적 크고, 비디오 데이터와 오디오 데이터로 이루어지는 AV(Audio/Video) 데이터를 기록하는데 적합한 기록 매체로서, 4.7GB(Giga Byte) 이상의 기록 용량을 갖는 DVD(Digital Versatile Disc)가 보급되어 있다. 특히 문헌 「일본 특개2004-350251」에는, 기록 가능한 타입의 DVD에 대해 DVD-Video 포맷으로 기록하는 촬상 장치가 기재되어 있다.

[0004] 이 기록 가능한 타입의 DVD는, 파일 시스템에 UDF(Universal Disk Format)가 이용되고 있고, UDF에 대응하는 컴퓨터 장치로 액세스가 가능하게 되어 있다. UDF는, ISO(International Organization for Standardization)9660에 의한 포맷을 포함하고 있고, 컴퓨터 장치에 이용되는 다양한 파일 시스템에서 액세스 가능하게 되어 있다. 이 기록 가능한 타입의 DVD에 대해, 비디오 데이터 및 오디오 데이터를 파일로서 기록함으로써, 컴퓨터 장치 등의 다른 장치와의 친화성이 늘어나고, 기록된 데이터를 보다 유효하게 활용하는 것이 가능해진다.

[0005] 또한, 근래에는, 비디오 카메라에 하드 디스크 드라이브를 내장하고, 촬영하여 얻어진 비디오 데이터 및 오디오 데이터를, 이 하드 디스크 드라이브에 파일로서 기록하는 제품도 출현하고 있다. 또한, 기록 매체로서 대용량의 반도체 메모리를 이용한 제품도 제안되어 있다.

[0006] 종래의, 기록 매체로서 자기 테이프를 이용한 경우에는, 기록 시작 조작부터 기록 정지 조작의 사이에 생성되는 비디오 데이터를 단위로 하여, 기록 매체에 대한 비디오 데이터 및 오디오 데이터의 기록을 행하는 것이 일반적으로 행하여지고 있다. 그래서, 비디오 데이터 및 오디오 데이터를 파일로서 기록하도록 한 경우도, 기록 시작 조작부터 기록 정지 조작의 사이에 생성되는 데이터를 단위로 하여 파일을 생성하면, 종래로부터의 기록 방식에도 익숙해지기 쉽고, 바람직하다.

[0007] 그런데, 연속적으로 공급되는 비디오 데이터 및 오디오 데이터를 파일로서 기록 매체에 기록하는 경우, 시스템 상의 제약에 의해, 기록 시작 조작부터 기록 정지 조작의 사이에 기록 가능한 기록시간으로 제한이 생기는 일이 있다. 시스템상의 제한으로서는, 예를 들면 기록 매체에 적용되는 파일 시스템상의 파일 사이즈의 제한이나, 비디오 데이터 및 오디오 데이터가 격납된 파일에서의 관리 정보에 관한 제한 등이 고려된다.

[0008] 파일 시스템상의 제한에 관해, 개략적으로 설명한다. 기록 매체에 기록된 파일을 컴퓨터 장치에서 취급하는 경우에는, 기록 매체에 기록된 파일의 형식이 컴퓨터 장치의 파일 시스템에 대응하고 있을 필요가 있다. 여기서, 컴퓨터 장치에 이용되는 보다 저위의 파일 시스템을 고려함으로써, 보다 많은 컴퓨터 장치에 대해 판독 가능하게 할 수 있다. 예를 들면, 컴퓨터 장치의 OS(Operating System)의 하나인 Windows(등록상표)에서 이용되는 파일 시스템인, FAT16(File Allocation Table 16)과 호환성을 갖게 하는 것이 고려된다.

[0009] FAT16에서는, 1파일의 최대 사이즈가 2GB(Giga Byte)로 제한되어 있다. 한편, UDF에서는, 1파일의 최대 사이즈는, 2GB에 비하여 훨씬 크다. 따라서 UDF의 규격에 준하여 기록된 파일의 사이즈가 2GB를 초과한 경우, 이 파일은, FAT16에서는 취급할 수 없는 것이 된다.

[0010] 이 때문에, 예를 들면 비디오 카메라를 생각한 경우, 유저는, 파일로서 기록된 디지털 비디오 데이터가 사용되는 컴퓨터 장치의 파일 시스템을 고려하면서, 기록 시작 및 정지를 행하여야 하고, 조작성을 현저하게 손상시켜 버린다는 문제점이 있다. 또한 이 때문에, 기록 매체에 기록 가능한 시간분의 기록이 행하여지지 않는 중에, 기

록을 정지하지 않을 수 없게 되는 것도 고려되어, 기록 용량에 따른 연속 기록시간을 확보할 수 있게 된다는 문제점이 있다.

[0011] 이 문제를 피하기 위해, 예를 들면, 기록중의 파일의 사이즈를 시스템이 감시하고, 파일 사이즈가 소정의 사이즈에 달하면 해당 파일을 클로즈하고, 신규로 파일을 작성하여 기록을 계속하는 것이 고려된다. 그러나, 이 경우에도, 재생시에, 연속적인 기록으로 생성된 복수의 파일을 연속 재생하기 위한 조작이 필요해지고, 유저에 있어서 부담이 되어 버린다는 문제점이 있다.

발명의 상세한 설명

[0012] 따라서, 본 발명의 목적은, 기록 시작부터 기록 정지까지의 사이에 생성된 비디오 데이터 및 오디오 데이터를 파일로서 기록하는 경우에 있어서, 장시간의 연속 기록 및 연속 재생이 용이한 기록 장치, 기록 방법 및 기록 프로그램, 및, 촬상 장치, 촬상 방법 및 촬상 프로그램을 제공하는 데 있다.

[0013] 상술한 과제를 해결하기 위해, 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 시작 및 기록 정지의 지시가 입력되는 기록 지시 입력부와, 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록부와, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성부와, 기록부 및 관리 정보 생성부를 제어하는 제어부를 가지며, 제어부는, 기록 지시 입력부에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록부를 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록부를 제어하는 것을 특징으로 하는 기록 장치이다.

[0014] 또한, 제 2의 발명은, 비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 방법에 있어서, 데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터의 기록 시작 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과, 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과, 기록의 스텝 및 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며, 제어의 스텝은, 기록 지시 입력의 스텝에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록의 스텝을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록의 스텝을 제어하는 것을 특징으로 하는 기록 방법이다.

[0015] 또한, 제 3의 발명은, 비디오 데이터와 오디오 데이터를 다중화하여 기록 매체에 기록하는 기록 방법을 컴퓨터 장치에 실행시키는 기록 프로그램에 있어서, 기록 방법은, 데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터의 기록 시작 및 기록 정지의 지시가 입력되는 기록 지시 입력의 스텝과, 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과, 기록의 스텝 및 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며, 제어의 스텝은, 기록 지시 입력의 스텝에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록의 스텝을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록의 스텝을 제어하는 것을 특징으로 하는 기록 방법이다.

보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록의 스텝을 제어하는 것을 특징으로 하는 기록 프로그램이다.

[0016] 또한, 제 4의 발명은, 활상부에서 피사체를 활상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음(收音)하여 얻어진 오디오 데이터를 다중화하여 기록 매체에 기록하는 활상 장치에 있어서, 피사체를 활상하여 비디오 데이터를 출력하는 활상부와, 음성을 수음하여 오디오 데이터를 출력하는 수음부와, 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록부와, 비디오 데이터 및 오디오 데이터의 기록 매체에의 기록 시작 및 기록 정지를 지시하는 유저 조작을 접수하는 조작부와, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성부와, 기록부 및 관리 정보 생성부를 제어하는 제어부를 가지며, 제어부는, 조작부에 대한 조작에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록부를 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록부를 제어하는 것을 특징으로 하는 활상 장치이다.

[0017] 또한, 제 5의 발명은, 활상부에서 피사체를 활상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음하여 얻어진 오디오 데이터를 다중화하여 기록 매체에 기록하는 활상 장치의 활상 방법에 있어서, 피사체를 활상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과, 조작부에 대한 비디오 데이터 및 오디오 데이터의 기록 매체에의 기록 시작 및 기록 정지를 지시하는 유저 조작을 접수하는 스텝과, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과, 기록의 스텝 및 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며, 제어의 스텝은, 조작부에 대한 조작에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록의 스텝을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록의 스텝을 제어하는 것을 특징으로 하는 활상 방법이다.

[0018] 또한, 제 6의 발명은, 활상부에서 피사체를 활상하여 얻어진 비디오 데이터와, 수음부에서 음성을 수음하여 얻어진 오디오 데이터를 다중화하여 기록 매체에 기록하는 활상 장치의 활상 방법을 컴퓨터 장치에 실행시키는 활상 프로그램에 있어서, 활상 방법은, 피사체를 활상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하는 기록의 스텝과, 조작부에 대한 비디오 데이터 및 오디오 데이터의 기록 매체에의 기록 시작 및 기록 정지를 지시하는 유저 조작을 접수하는 스텝과, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하는 관리 정보 생성의 스텝과, 기록의 스텝 및 관리 정보 생성의 스텝을 제어하는 제어의 스텝을 가지며, 제어의 스텝은, 조작부에 대한 조작에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록의 스텝을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록의 스텝을 제어하는 것을 특징으로 하는 활상 프로그램이다.

[0019] 상술한 바와 같이, 제 1, 제 2 및 제 3의 발명은, 데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화한 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하고, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보

파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하고, 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 제어하도록 하고 있기 때문에, 스트림 파일에 대한 소정의 제약을 의식하는 일 없이, 장시간의 연속적인 기록을 행할 수 있다.

[0020] 또한, 제 4, 제 5 및 제 6의 발명은, 피사체를 촬상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하고, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하고, 조작부에 대한 조작에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록을 제어하도록 하고 있기 때문에, 유저는, 스트림 파일에 대한 소정의 제약을 의식하는 일 없이, 장시간의 연속적인 촬영 및 촬영되어 얻어진 비디오 데이터 및 오디오 데이터의 기록을 행할 수 있다.

[0021] 제 1, 제 2 및 제 3의 발명은, 상술한 바와 같이, 데이터 입력으로 입력된 비디오 데이터 및 오디오 데이터를 패킷 단위로 다중화한 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하고, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하고, 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 제어하게 하고 있기 때문에, 스트림 파일에 대한 소정의 제약을 의식하는 일 없이, 장시간의 연속적인 기록을 행할 수 있는 효과가 있다.

[0022] 또한, 제 4, 제 5 및 제 6의 발명은, 피사체를 촬상하여 얻어진 비디오 데이터와, 음성을 수음하여 얻어진 오디오 데이터를 패킷 단위로 다중화하고, 다중화된 스트림을 패킷마다 순서로, 기록 매체상의 스트림 파일에 기록하고, 기록 매체에 기록되는 스트림 파일에 대해, 적어도, 스트림 파일의 재생시각 정보와 어드레스 정보를 대응시킨 스트림 정보 파일과, 스트림 파일에 대해 재생 시작점과 재생 종료점을 설정함에 의해 재생 구간을 지정하는 하나 이상의 재생 구간 데이터가 격납되고, 스트림 파일에 대한 재생시각 정보를 나타내는 마크 정보가 격납 가능한 재생 리스트 파일을 생성하고, 조작부에 대한 조작에 의한 기록 시작 및 기록 정지의 지시에 따라 스트림의 스트림 파일에 대한 기록의 시작 및 정지를 각각 행하도록 기록을 제어하고, 하나 또는 복수의 패킷으로 이루어지는 소정 단위를 스트림 파일에 기록함으로써 스트림 파일의 속성을 나타내는 정보가 소정의 제약을 충족시키지 않게 되는 경우에, 스트림 파일을 클로즈하여 신규의 스트림 파일을 작성하고, 신규의 스트림 파일에 대해 스트림을 소정 단위로 순서로, 기록하도록 기록을 제어하게 하고 있기 때문에, 유저는, 스트림 파일에 대한 소정의 제약을 의식하는 일 없이, 장시간의 연속적인 촬영 및 촬영되어 얻어진 비디오 데이터 및 오디오 데이터의 기록을 행할 수 있는 효과가 있다.

실시예

[0101] 이하, 본 발명의 실시의 한 형태를, 도면을 참조하면서 설명한다. 우선, 이해를 용이하게 하기 위해, 본 발명에 적용 가능한 한 예의 포맷(이하, AVCHD 포맷이라고 부른다)에 관해 설명한다. AVCHD 포맷은, 비디오 데이터와 오디오 데이터가 소정(所定)의 다중화된 AV(Audio/Video) 스트림을 기록 가능한 기록 매체에 기록하는 기록 포

맷으로서 현재 제안되어 있는 것으로, 기록 매체에 기록된 AV 스트림을, 클립 단위로 플레이 리스트를 이용하여 관리 가능하게 하고 있다.

[0102] 예를 들면 ITU-T(International Telecommunication Union-Telecommunication Standardization Sector) 권고 H.264 또는 ISO(International Organization for Standardization)/IEC(International Electrotechnical Commission) 국제표준 14496-10(MPEG-4 파트 10) Advanced Video Coding(이하, H.264|AVC라고 약칭한다)에 규정되는 부호화 방식으로 부호화되고, MPEG2 시스템에 따라 다중화된 비트 스트림은, 클립 AV 스트림(또는 AV 스트림)이라고 칭하여진다. 클립 AV 스트림은, 소정의 파일 시스템에 의해 파일로서 디스크에 기록된다. 이 파일을, 클립 AV 스트림 파일(또는 AV 스트림 파일)이라고 칭한다.

[0103] 클립 AV 스트림 파일은, 파일 시스템상에서의 관리 단위이고, 유저에 있어서 반드시 알기 쉬운 관리 단위라고는 한정할 수 없다. 유저의 편리성을 고려한 경우, 복수의 클립 AV 스트림 파일에 분할된 영상 컨텐츠를 하나로 정리하여 재생하는 구조나, 클립 AV 스트림 파일의 일부만을 재생하는 구조, 나아가서는, 특수 재생이나 두출(頭出) 재생을 원활하게 행하기 위한 정보 등을 데이터베이스로서 디스크에 기록하여 둘 필요가 있다.

[0104] 도 1은, 본 발명에 적용 가능한 AVCHD 포맷에 규정되는 데이터 모델을 개략적으로 도시한다. 이 AVCHD 포맷에 의하면, 데이터 구조는, 도 1에 도시되는 바와 같이 4층의 레이어로 이루어진다. 가장 최하층의 레이어는, 클립 AV 스트림이 배치되는 레이어이다(편의상, 클립 레이어라고 부른다). 그 위의 레이어는, 클립 AV 스트림에 대한 재생 개소를 지정하기 위한, 플레이 리스트(PlayList)와, 플레이 아이템(PlayItem)이 배치되는 레이어이다(편의상, 플레이 리스트 레이어라고 부른다). 또한 그 위의 레이어는, 플레이 리스트에 대해 재생 순서 등을 지정하는 커맨드로 이루어지는 무비 오브젝트(Movie Object) 등이 배치되는 레이어이다(편의상, 오브젝트 레이어라고 부른다). 최상층의 레이어는, 기록 매체에 격납되는 타이틀 등을 관리하는 인덱스 테이블이 배치된다(편의상, 인덱스 레이어라고 부른다).

[0105] 클립 레이어에 관해 설명한다. 클립 AV 스트림은, 비디오 데이터나 오디오 데이터가 MPEG2 TS(트랜스포트 스트림)의 형식 등으로 다중화된 비트 스트림이다. 이 클립 AV 스트림에 관한 정보가 클립 정보(Clip Information)로서 파일에 기록된다.

[0106] 또한, 클립 AV 스트림에는, 자막(字幕)을 표시하는 그래픽스 스트림인 OB 스트림(Overlay Bitmap stream)이나, 메뉴 표시 등에 이용되는 데이터(버튼 화상 데이터 등)를 스트림으로 한 MB 스트림(Menu Bitmap stream)을 다중화할 수 있다.

[0107] 클립 AV 스트림 파일과, 대응하는 클립 정보가 기록된 클립 정보 파일을 하나의 정리된 오브젝트라고 간주하고, 클립(Clip)이라고 칭한다. 즉, 클립은, 클립 AV 스트림과 클립 정보로 구성되는, 하나의 오브젝트이다.

[0108] 파일은, 일반적으로, 바이트열로서 취급된다. 클립 AV 스트림 파일의 컨텐츠는, 시간축상에 전개되고, 클립중의 엔트리 포인트는, 주로 시간 베이스로 지정된다. 소정의 클립에의 액세스 포인트의 타임 스템프가 주어진 경우, 클립 AV 스트림 파일중에서 데이터의 판독을 시작하여야 할 어드레스 정보를 찾기 위해, 클립 정보 파일을 이용할 수 있다.

[0109] 플레이 리스트 레이어에 관해 설명한다. 플레이 리스트는, 재생하는 AV 스트림 파일의 지정과, 지정된 AV 스트림 파일의 재생 개소를 지정하는 재생 시작점(IN점)과 재생 종료점(OUT점)의 모임으로 구성된다. 이 재생 시작점과 재생 종료점의 정보를 하나의 조(組)로 한 것은, 플레이 아이템(PlayItem)이라고 칭하여진다. 플레이 리스트는, 플레이 아이템의 집합으로 구성된다. 플레이 아이템을 재생한다는 것은, 그 플레이 아이템에 참조되는 AV 스트림 파일의 일부분을 재생한다는 것이 된다. 즉, 플레이 아이템 중의 IN점 및 OUT점 정보에 의거하여, 클립 중의 대응하는 구간이 재생된다.

[0110] 오브젝트 레이어에 관해 설명한다. 무비 오브젝트는, 내비게이션 커맨드 프로그램과, 무비 오브젝트를 연휴(連携)하는 터미널 인포메이션을 포함한다. 내비게이션 프로그램은, 플레이 리스트의 재생을 제어하기 위한 커맨드(내비게이션 커맨드 : navigation command)이다.

[0111] 인덱스 레이어에 관해 설명한다. 인덱스 레이어는, 인덱스 테이블(Index Table)로 이루어진다. 인덱스 테이블은, 기록 매체에 기록된 컨텐츠의 타이틀을 정의한, 톱 레벨의 테이블이다. 인덱스 테이블에 격납되어 있는 타이틀 정보에 의거하여, 플레이어에 상주하는 시스템 소프트웨어중의 모듈 매니저에 의해 기록 매체의 재생이 제어된다.

[0112] 즉, 도 2에 개략적으로 도시되는 바와 같이, 인덱스 테이블중의 임의의 엔트리는, 타이틀이라고 칭하여지고, 인

인덱스 테이블에 엔트리되는 퍼스트 플레이백 타이틀(First PlaybackTitle), 메뉴 타이틀(MenuTitle) 및 무비 타이틀(MovieTitle)(#1, #2, ...)은, 전부 타이틀이다. 각 타이틀은, 무비 오브젝트에 대한 링크를 나타낸다.

[0113] 이해를 용이하게 하기 위해 재생 전용의 기록 매체를 예로 들면, 예를 들면, 퍼스트 플레이백 타이틀은, 해당 기록 매체에 격납되는 컨텐츠가 영화라면, 영화 본편에 앞서서 영상이 나오는 영화회사의 선전용 영상(트레일러)에 대응한다. 메뉴 타이틀은, 예를 들면 컨텐츠가 영화인 경우, 본편 재생, 챕터 서치, 자막이나 언어 설정, 특정 영상 재생 등을 선택하기 위한 메뉴 화면에 대응한다. 또한, 무비 타이틀은, 메뉴 타이틀로부터 선택되는 각 영상이다. 타이틀이 또한 메뉴 화면인 구성도 가능하다.

[0114] 도 3은, 상술한 바와 같은 클립 AV 스트림, 클립 정보(Stream Attributes), 클립, 플레이 아이템 및 플레이 리스트의 관계를 도시하는 UML(Unified Modeling Language)도이다. 플레이 리스트는, 하나 또는 복수의 플레이 아이템에 대응시켜지고, 플레이 아이템은, 하나의 클립에 대응시켜진다. 하나의 클립에 대해, 각각 시작점 및/또는 종료점이 다른 복수의 플레이 아이템을 대응시킬 수 있다. 하나의 클립으로부터 하나의 클립 AV 스트림 파일이 참조된다. 마찬가지로, 하나의 클립으로부터 하나의 클립 정보 파일이 참조된다. 또한, 클립 AV 스트림 파일과 클립 정보 파일은, 1대1의 대응 관계를 갖는다. 이와 같은 구조를 정의함에 의해, 클립 AV 스트림 파일을 변경하는 일 없이, 임의의 부분만을 재생하는, 비파괴의 재생 순서 지정을 행하는 것이 가능해진다.

[0115] 또한, 도 4와 같이, 복수의 플레이 리스트로부터 동일한 클립을 참조하는 것도 가능하다. 또한, 하나의 플레이 리스트로부터 복수의 클립을 지정하는 것도 가능하다. 클립은, 플레이 리스트 중의 플레이 아이템에 나타나는 IN점 및 OUT점에 의해, 참조된다. 도 4의 예에서는, 클립(300)은, 플레이 리스트(310)의 플레이 아이템(320)으로부터 참조됨과 함께, 플레이 리스트(311)를 구성하는 플레이 아이템(321 및 322)중 플레이 아이템(321)으로부터, IN점 및 OUT점으로 나타나는 구간이 참조된다. 또한, 클립(301)은, 플레이 리스트(311)의 플레이 아이템(322)으로부터 IN점 및 OUT점으로 나타나는 구간이 참조됨과 함께, 플레이 리스트(312)의 플레이 아이템(323 및 324)중, 플레이 아이템(323)의 IN점 및 OUT점으로 나타나는 구간이 참조된다. 도 4의 예에서는, 클립(301)은, 또한 다른 플레이 리스트로부터도 참조되고 있다.

[0116] 다음에, AVCHD 포맷에 의한, 기록 매체에 기록되는 파일의 관리 구조에 관해, 도 5를 이용하여 설명한다. 파일은, 디렉토리 구조에 의해 계층적으로 관리된다. 기록 매체상에는, 우선, 하나의 디렉토리(도 5의 예에서는 루트(root) 디렉토리)가 작성된다. 이 디렉토리의 아래가, 하나의 기록 재생 시스템에서 관리되는 범위로 한다.

[0117] 루트 디렉토리의 아래에, 디렉토리 "BDMV"가 놓여진다. 또한 필요에 따라, 루트 디렉토리의 아래에 디렉토리 "AVCHDTN"가 놓여진다. 디렉토리 "AVCHDTN"에는, 예를 들면 클립의 대표 화상을 소정 사이즈로 축소한 썸네일 파일이 놓여진다. 디렉토리 "BDMV"에, 도 1을 이용하여 설명한 데이터 구조가 격납된다.

[0118] 디렉토리 "BDMV"의 바로 아래에는, 파일은, 파일 "index.bdmv" 및 파일 "MovieObject.bdmv"의 2개만을 놓을 수 있다. 또한, 디렉토리 "BDMV"의 아래에, 디렉토리 "PLAYLIST", 디렉토리 "CLIPINF", 디렉토리 "STREAM" 및 디렉토리 "BACKUP"가 놓여진다. 디렉토리 "BACKUP"는, 각 디렉토리 및 파일의 백업이 격납된다.

[0119] 파일 "index.bdmv"은, 디렉토리 "BDMV"의 내용에 관해 기술된다. 즉, 이 파일 "index.bdmv"이 상술한 최상층의 레이어인 인덱스 레이어에서의 인덱스 테이블에 대응한다. 또한, 파일 "MovieObject.bdmv"은, 하나 이상의 무비 오브젝트의 정보가 격납된다. 즉, 이 파일 "MovieObject.bdmv"이 상술한 오브젝트 레이어에 대응한다.

[0120] 디렉토리 "PLAYLIST"는, 플레이 리스트의 데이터베이스가 놓여지는 디렉토리이다. 즉, 디렉토리 "PLAYLIST"는, 플레이 리스트에 관한 파일인 파일 "xxxxx.mpls"을 포함한다. 파일 "xxxxx.mpls"은, 플레이 리스트의 각각에 대해 작성되는 파일이다. 파일명에 있어서, "."(피리어드) 앞의 "xxxxx"는, 5자릿수의 숫자로 되고, 피리어드의 뒤의 "mpls"는, 이 타입의 파일에 고정적이 되는 확장자이다.

[0121] 디렉토리 "CLIPINF"는, 클립의 데이터베이스가 놓여지는 디렉토리이다. 즉, 디렉토리 "CLIPINF"는, 클립 AV 스트림 파일의 각각에 대한 클립 인포메이션 파일인 파일 "zzzzz.clpi"을 포함한다. 파일명에 있어서, "."(피리어드) 앞의 "zzzzz"는, 5자릿수의 숫자로 되고, 피리어드의 뒤의 "clpi"는, 이 타입의 파일에 고정적이 되는 확장자이다.

[0122] 디렉토리 "STREAM"은, 실제로서의 AV 스트림 파일이 놓여지는 디렉토리이다. 즉, 디렉토리 "STREAM"은, 클립 인포메이션 파일의 각각에 대응하는 클립 AV 스트림 파일을 포함한다. 클립 AV 스트림 파일은, MPEG2(Moving Pictures Experts Group 2)의 트랜스포트 스트림(이하, MPEG2 TS라고 약칭한다)으로 이루어지고, 파일명이 "zzzzz.m2ts"가 된다. 파일명에 있어서, 피리어드 앞의 "zzzzz"는, 대응하는 클립 인포메이션 파일과 동일하게

함으로써, 클립 인포메이션 파일과 이 클립 AV 스트림 파일과의 대응 관계를 용이하게 파악할 수 있다.

[0123] 또한, 디렉토리 "AVCHDTN"는, 2종류의 섬네일 파일 "thumbnail.idx" 및 "thumbnail.tdt2"을 둘 수 있다. 섬네일 파일 "thumbnail.idx"은, 소정의 방식으로 암호화된 섬네일 화상이 격납된다. 섬네일 파일 "thumbnail.tdt2"은, 암호화되지 않은 섬네일 화상이 격납된다. 예를 들면 비디오 카메라로 유저가 촬영한 클립에 대응하는 섬네일 화상은, 카피 프리하여 암호화할 필요가 없다고 생각되기 때문에, 이 섬네일 파일 "thumbnail.tdt2"에 격납된다.

[0124] 도 5에서 도시한 각 파일중, 본 발명에 관계가 깊은 것에 관해, 보다 상세히 설명한다. 우선, 디렉토리 "BDMV"의 바로 아래에 놓여지는 파일 "index.bdmv"에 관해 설명한다. 도 6은, 이 파일 "index.bdmv"의 한 예의 구조를 표시하는 신택스를 도시한다. 여기서는, 신택스를 컴퓨터 장치 등의 프로그램의 기술 언어로서 이용되는 C언어의 기술법에 의거하여 나타낸다. 이것은, 다른 신택스를 도시하는 도면에서도, 마찬가지이다.

[0125] 도 6에서, 필드(TypeIndicator)는 32비트의 데이터 길이를 가지며, 이 파일이 인덱스 테이블인 것을 나타낸다. 필드(TypeIndicator2)는 32비트의 데이터 길이를 가지며, 이 파일 "index.bdmv"의 베전을 나타낸다. 필드(IndexesStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스 내에 있는 블록(blkIndexes())의 시작 어드레스를 나타낸다.

[0126] 필드(ExtensionDataStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스 내에 있는 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 블록(blkExtensionData())은 소정의 확장 데이터를 격납 가능하게 하기 위한 블록이다. 필드(ExtensionDataStartAddress)는, 이 파일 "index.bdmv"의 최초의 바이트로부터의 상대바이트수로, 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 상대바이트수는, "0"부터 시작된다. 만약, 이 필드(ExtensionDataStartAddress)의 값이 "0"이면, 이 파일 "index.bdmv" 내에, 블록(blkExtensionData())이 존재하지 않는 것을 나타낸다.

[0127] 필드(ExtensionDataStartAddress)에 계속해서, 데이터 길이가 192바이트의 영역(reserved)이 배치된다. 또한, 영역(reserved)은, 바이트 열라인먼트나, 장래적인 필드의 추가 등을 위한 영역이다. 이것은, 이하의 설명에서도 마찬가지이다. 블록(blkAppInfoBDMV())은, 내용 제작자가 임의의 정보를 기술할 수 있는 블록이고, 플레이어의 동작 등에는 영향을 주지 않는다.

[0128] 블록(blkIndexes())은, 이 파일 "index.bdmv"의 실질적인 내용이고, 이 블록(blkIndexes())에 기술된 내용에 의해, 디스크를 플레이어에 장전한 때에 재생되는 퍼스트 플레이백이나, 텁 메뉴로부터 호출된 타이틀(무비 오브젝트)이 지정된다. 인덱스 테이블에 의해 호출된 무비 오브젝트 등에 기술된 커맨드에 의거하여, 후술하는 플레이어 리스트 파일이 판독된다.

[0129] 도 7은, 블록(blkIndexes())의 한 예의 구조를 표시하는 신택스를 도시한다.

[0130] 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length) 직후부터 이 블록(blkIndexes())의 끝까지의 데이터 길이를 나타낸다. 계속해서, 블록(FirstPlaybackTitle()) 및 블록(MenuTitle())이 배치된다.

[0131] 블록(FirstPlaybackTitle())은, 퍼스트 플레이백에서 이용되는 오브젝트에 관한 정보가 기술된다. 블록(FirstPlaybackTitle())은, 1비트의 데이터 길이를 갖는 영역(reserved)에 계속해서 고정치"1"이 기술된다. 또한 31비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 고정치"1"이 기술된다. 그리고, 14비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 16비트의 데이터 길이를 갖는 필드(FirstPlaybackTitleMobjIDRef)가 배치된다. 이 필드(FirstPlaybackTitleMobjIDRef)에 의해, 퍼스트 플레이백 타이틀에서 이용되는 무비 오브젝트의 ID를 나타낸다.

[0132] 무비 오브젝트의 ID는, 예를 들면, 도 8 및 도 9를 이용하여 후술하는 무비 오브젝트의 신택스에 의거하여, 무비 오브젝트의 for루프 문자(文)에 있어서 루프 변수로서 이용되는 값(mobj_id)으로 나타난다. 이 예에서는, 필드(FirstPlaybackTitleMobjIDRef)는, 참조하는 무비 오브젝트에 대응하는 값(mobj_id)이 격납된다.

[0133] 또한, 블록(blkIndexes())에서의 블록(FirstPlaybackTitle()) 내의 필드(FirstPlaybackTitleMobjIDRef)는, 텁 메뉴의 무비 오브젝트를 가리키고 있어도 좋고, 타이틀을 가리키고 있어도 좋다.

[0134] 블록(MenuTitle())은, 텁 메뉴에서 이용되는 오브젝트에 관한 정보가 기술된다. 블록(MenuTitle())은, 1비트의 데이터 길이를 갖는 영역(reserved)에 계속해서 고정치"1"이 기술된다. 또한 31비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 고정치"1"이 기술된다. 그리고, 14비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 16비트의 데이터 길이를 갖는 필드(MenuTitleMobjIDRef)가 배치된다. 필드(MenuTitleMobjIDRef)는,

메뉴 타이틀에서 이용되는 무비 오브젝트의 ID를 나타낸다.

[0135] 블록(MenuTitle())의 다음의 필드(NumberOfTitles)는, 16비트의 데이터 길이를 가지며, 유저가 선택, 재생 가능한 타이틀의 수를 나타낸다. 다음의 for루프문자에 따라, 이 필드(NumberOfTitles)에 나타나는 회수만큼, 값(title_id)을 인수(引數)로 하여, 블록(MovieTitle[title_id]())이 기술된다. 블록(MovieTitle[title_id]())은, 타이틀마다의 정보가 기술된다. 값(title_id)은, "0"부터 필드(NumberOfTitles)로 나타나는 값까지의 수치이고, 타이틀을 식별한다.

[0136] 블록(MovieTitle[title_id]())에서, 1비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 고정치"1"이 기술되고, 또한, 46비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(MovieTitleMobjIDRef)가 기술된다. 필드(MovieTitleMobjIDRef)는, 16비트의 데이터 길이를 가지며, 이 타이틀에서 이용되는 무비 오브젝트의 ID를 나타낸다. 필드(MovieTitleMobjIDRef)의 뒤에, 32비트의 데이터 길이를 갖는 영역(reserved)이 배치된다.

[0137] 도 8은, 디렉토리"BDMV"의 바로 아래에 놓여지는 파일"MovieObject.bdmv"의 한 예의 구조를 표시하는 선택스를 도시한다. 필드(TypeIndicator)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 파일"MovieObject.bdmv"인 것을 나타낸다. 필드(TypeIndicator)는, ISO(International Organization for Standardization)646에 규정된 부호화 방식으로 부호화한 4문자로 이루어지는 문자열이 기술된다. 이 도 8의 예에서는, 필드(type_indicator)에 ISO646에 규정의 방식으로 부호화된 4문자의 문자열"MOBJ"가 기술되고, 이 파일이 파일"MovieObject.bdmv"인 것이 나타난다.

[0138] 필드(TypeIndicator2)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일"MovieObject.bdmv"의 버전 번호를 나타낸다. 이 파일"MovieObject.bdmv"에서는, 필드(TypeIndicator2)는, ISO646에 규정된 부호화 방식으로 부호화한 4문자의 문자열"0100"이어야 한다.

[0139] 필드(ExtensionDataStartAddress)는 32비트의 데이터 길이를 가지며, 이 선택스 내에 있는 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 필드(ExtensionDataStartAddress)는, 이 파일"MovieObject.bdmv"의 최초의 바이트로부터의 상대바이트수로, 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 상대바이트수는, "0"부터 시작된다. 만약, 이 필드(ExtensionDataStartAddress)의 값이 "0"이면, 이 파일"MovieObject.bdmv" 내에, 블록(blkExtensionData())이 존재하지 않는 것을 나타낸다.

[0140] 또한, 이 도 8에 도시하는 선택스 내의 필드(padding_word)는, 16비트의 데이터 길이를 가지며, 이 파일"MovieObject.bdmv"의 선택스에 따라 for루프문자에 값(N1) 또는 값(N2)으로 나타나는 회수만큼 삽입된다. 값(N1) 또는 값(N2)은, 0 또는 임의의 정(正)의 정수이다. 또한, 필드(padding_word)는, 임의의 값을 이용할 수 있다.

[0141] 필드(ExtensionDataStartAddress)에 계속해서 데이터 길이가 224비트의 영역(reserved)이 배치되고, 그 다음에, 이 파일"MovieObject.bdmv"의 본체인 블록(blkMovieObjects())이 격납된다.

[0142] 도 9는, 블록(blkMovieObjects())의 한 예의 구조를 표시하는 선택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 이 블록(blkMovieObjects())의 끝까지의 데이터 길이를 나타낸다. 32비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(NumberOfMobjs)가 배치된다. 필드(NumberOfMobjs)는, 직후의 for루프문자에 따라 격납되는 무비 오브젝트의 수를 나타낸다. for루프문자의 루프 변수로서 이용되는 값(mobj_id)으로, 무비 오브젝트가 일의적으로 특정된다. 값(mobj_id)은, "0"부터 시작되는 값이고, 무비 오브젝트는, for루프문자 중에 기술되는 순서에 의해 정의된다.

[0143] for루프문자 중의 블록(TerminalInfo())은, 고정치"1"이 기술되고, 다음에 15비트의 데이터 길이를 갖는 영역(reserved)이 배치된다. 그 다음에, 16비트의 데이터 길이를 갖는 필드(NumberOfNavigationCommands[mobj_id])가 배치된다. 이 필드(NumberOfNavigationCommands[mobj_id])는, 값(mobj_id)에 의해 지시되는 무비 오브젝트(MovieObject[mobj_id]())에 포함되는 내비게이션 커맨드(navigationcommand)의 수를 나타낸다.

[0144] 다음, 값(command_id)을 루프 변수로 하는 for루프문자에 의해, 필드(NumberOfNavigationCommands[mobj_id])에 도시되는 수만큼, 내비게이션 커맨드가 기술된다. 즉, 이 for루프문자중에 배치되는 필드(NavigationCommand[mobj_id][command_id])는, 값(mobj_id)에 의해 지시되는 블록(MovieObject[mobj_id]())에 포함되는, 값(command_id)으로 나타나는 순번의 내비게이션 커맨드(NavigationCommand)를 격납한다. 값(command_id)은, 0부터 시작되는 값이고, 내비게이션 커맨드(NavigationCommand)는, 이 for루프문자중에 기술

되는 순서로 정의된다.

[0145] 도 10은, 플레이 리스트 파일 "xxxxx.mpls"의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(TypeIndicator)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 플레이 리스트 파일인 것을 나타낸다. 필드(TypeIndicator2)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 플레이 리스트 파일의 버전을 나타낸다. 필드(PlayListStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스중의 블록(blkPlayList())의 시작 어드레스를 나타낸다.

[0146] 필드(PlayListMarkStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스중의 블록(blkPlayListMark())의 시작 어드레스를 나타낸다. 필드(ExtensionDataStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스중의 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 필드(ExtensionDataStartAddress)는, 블록(blkExtensionData())의 시작 어드레스를, 파일 "xxxxx.mpls"의 최초의 바이트로부터의 상대바이트수를 나타낸 값이다. 상대바이트수는, "0"부터 시작된다. 만약, 이 필드(ExtensionDataStartAddress)의 값이 "0"이면, 이 파일 "xxxxx.mpls" 내에, 블록(blkExtensionData())이 존재하지 않는 것을 나타낸다.

[0147] 160비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 블록(blkAppInfoPlayList())이 배치된다. 블록(blkAppInfoPlayList())은, 다음의 블록(blkPlayList())에 기술되는 플레이 리스트의 탑입, 재생 제한 등의 정보가 기술된다. 블록(blkPlayList())은, 플레이 리스트가 기술된다. 블록(blkPlayListMark())은, 챕터 점프 등으로 점프되는 포인트가 기술된다. 블록(blkExtensionData())은 소정의 확장 데이터를 격납 가능하게 하기 위한 블록이다.

[0148] 또한, 이 도 10에 도시하는 신택스 내의 필드(padding_word)는, 16비트의 데이터 길이를 가지며, 이 파일 "xxxxx.mpls"의 신택스에 따라 for루프문자에 값(N1), 값(N2) 및 값(N3)으로 나타나는 회수만큼 삽입된다. 값(N1), 값(N2) 또는 값(N3)은, 0 또는 임의의 정의 정수이다. 또한, 필드(padding_word)는, 임의의 값을 이용할 수 있다.

[0149] 도 11은, 블록(blkPlayList())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkPlayList())의 최후까지의 데이터 길이를 나타낸다. 필드(Length)에 계속해서 16비트의 데이터 길이를 갖는 영역(reserved)이 배치되고, 다음에 필드(NumberOfPlayItems)가 배치된다. 필드(NumberOfPlayItems)는, 16비트의 데이터 길이를 가지며, 이 블록(blkPlayList())에 포함되는 플레이 아이템의 수를 나타낸다. 필드(NumberOfSubPath)는, 이 블록(blkPlayList())에 포함되는 서브패스의 수를 나타낸다.

[0150] 다음의 for루프문자에 따라, 필드(NumberOfPlayItems)에서 나타나는 수만큼, 플레이 아이템이 기술되는 블록(blkPlayItem())이 기술된다. for루프문자에 의거한 카운트 수가 블록(blkPlayItem())의 식별자(PlayItem_id)가 된다. 또한 다음의 for루프문자에 따라, 필드(NumberOfSubPath)에서 나타나는 수만큼, 블록(blkSubPath())이 기술된다. for루프문자에 의거한 카운트 수가 블록(blkSubPath())의 식별자(SubPath_id)가 된다.

[0151] 또한, 서브패스는, 주로 재생되는 플레이 아이템에 대응하는 메인 패스에 대해, 서브 플레이 아이템에 대응하여 갖을 수 있다. 서브패스는, 예를 들면, 애프터레코딩용의 오디오 데이터의 지정이나, 2장의 영상을 합성할 때에, 플레이 아이템으로 지정되는 클립과 동기하여 재생하는 부영상의 지정한다는 목적으로 이용된다.

[0152] 도 12는, 블록(blkPlayItem())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는, 16비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkPlayItem())의 최후까지의 데이터 길이를 나타낸다.

[0153] 필드(ClipInformationFileName)는, 40비트(5바이트)의 데이터 길이를 가지며, 이 블록(blkPlayItem())이 참조하는 클립 인포메이션 파일의 파일명이 나타난다. 이 플레이 아이템에서, 필드(ClipInformationFileName[0])로 나타나는 파일명의 클립 인포메이션 파일이 판독된다. 필드(ClipCodecIdentifier[0])는, 32비트(4바이트)의 데이터 길이를 가지며, 이 블록(blkPlayItem())에 의한 플레이 아이템에서 이용되는 클립 AV 스트림의 코덱 방식을 나타낸다.

[0154] 12비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 필드(ConnectionCondition)가 배치된다. 필드(ConnectionCondition)는, 4비트의 데이터 길이를 가지며, 클립 사이의 접속 상태에 관한 정보를 나타낸다. 기록용도의 기록 매체에 대해서는, 필드(ConnectionCondition)의 값으로서 "1", "5" 또는 "6"이 이용된다. 필드(ConnectionCondition)의 값이 "1"에서, 그 플레이 아이템으로부터 참조되고 있는 클립과 바로앞의 플레이 아이템으로부터 참조되고 있는 클립이 심レス 접속하지 않는 것을 나타내고, 필드(ConnectionCondition)의 값이 "5" 또는 "6"에서, 그 플레이 아이템으로부터 참조되고 있는 클립과 바로앞의 플레이 아이템으로부터 참조되고 있는

클립이 심레스 접속하는 것을 나타낸다. 또한, 심레스 접속이란, 클립과 다음의 클립이 프레임 타이밍에서 연속적으로 재생되도록, 클립 사이의 재생 제어를 행하는 것을 말한다.

[0155] 필드(ConnectionCondition)의 값이 "5"에서, 해당 플레이 아이템이 참조하는 클립에서, 오디오 데이터의 기록 길이가 비디오 데이터의 기록 길이에 대해 길게 된다(도 13A 참조). 이로 인해, 클립과 클립을 접속할 때에, 오디오 데이터의 페이드아웃 처리가 가능하게 된다. 예를 들면, 유저에 의한 기록 정지 조작에 의해 클립이 클로즈되는 경우에, 필드(ConnectionCondition)의 값이 "5"가 된다. 이하, 이 필드(ConnectionCondition)의 값이 "5"로 나타나는 클립의 접속 방법을, 제 1의 심레스 접속이라고 부른다.

[0156] 필드(ConnectionCondition)의 값이 "6"에서, 해당 플레이 아이템이 참조하는 클립에서, 오디오 데이터의 기록 길이가 비디오 데이터의 기록 길이에 대해 같게 된다(도 13B 참조). 이로 인해, 클립과 클립과의 접속을 심레스하게 행하는 것이 가능하게 된다. 예를 들면, 유저 조작에 따른 기록 정지 이외의 이유, 예를 들면 시스템 요인에 의거하여 클립이 클로즈되는 경우에, 필드(ConnectionCondition)의 값이 "6"이 된다. 이하, 이 필드(ConnectionCondition)의 값이 "6"으로 나타나는 클립의 접속 방법을, 제 2의 심레스 접속이라고 부른다.

[0157] 필드(RefToSTCID[0])는, 8비트의 데이터 길이를 가지며, 시스템 타임 베이스(STC)의 불연속점에 관한 정보를 나타낸다. 필드(INTime) 및 필드(OUTTime)는, 각각 32비트의 데이터 길이를 가지며, 메인 클립 AV 스트림의 재생 범위를 나타낸다. 필드(INTime)가 시작점(IN점)을 나타내고, 필드(OUTTime)가 종료점(OUT점)을 나타낸다.

[0158] 블록(blkUOMaskTable())은, 유저 입력의 접수 제한이 설정되는 테이블이다. 1비트의 데이터 길이를 갖는 플래그(PlayItemRandomAccessFlag)는, 이 블록(blkPlayItem())에 의한 플레이 아이템에 대해 랜덤 액세스를 허가하는지의 여부를 규정한다. 계속해서, 7비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(StillMode)가 배치된다. 필드(StillMode)는, 8비트의 데이터 길이를 가지며, 블록(blkPlayItem())에 의한 플레이 아이템에서, 최후로 표시한 영상을 정지화로서 표시시키는지의 여부를 나타낸다. 필드(StillMode)의 값이 "0x01"(바이너리)이면, if문자(文)에 의거하여, 16비트의 데이터 길이를 갖는 필드(StillTime)에 의해 정지 시간이 나타난다. 필드(StillMode)의 값이 "0x01" 이외이면, 해당 16비트의 데이터 길이를 갖는 영역이 영역(reserved)이 된다.

[0159] 또한, 수치의 기술에서 "0x"는, 그 수치가 16진 표기되어 있는 것을 나타낸다. 이것은, 이하의 같은 표기에 관해 공통이다.

[0160] 블록(blkSTNTable())은, 이 블록(blkPlayItem())에 의한 플레이 아이템이 관리하고 있는 클립 AV 스트림의 속성, PID 번호, 기록 매체상에서의 기록 위치 등이 관리된다.

[0161] 도 14는, 블록(blkPlayListMark())의 한 예의 구조를 표시하는 선택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkPlayListMark())의 최후까지의 데이터 길이를 나타낸다.

[0162] 필드(NumberOfPlayListMarks)는, 16비트의 데이터 길이를 가지며, 이 블록(blkPlayListMark())에 포함되는 플레이 리스트 마크의 수를 나타낸다. 다음의 for루프문자에 따라, 필드(NumberOfPlayListMarks)에서 나타나는 수 만큼 플레이 리스트 마크의 정보가 기술된다.

[0163] for루프문자 내에서, 8비트의 데이터 길이를 갖는 영역 reserve에 계속해서 필드(MarkType)가 배치된다. 필드(MarkType)는, 8비트의 데이터 길이를 가지며, 마크의 타입을 나타낸다. 플레이 리스트 마크에는, 엔트리 마크(EntryMark) 및 링크 포인트(LinkPoint)의 2타입이 정의되어 있고, 이 필드(MarkType)에 의해, 어느 타입인지가 나타난다. 챕터를 정의하기 위해서는, 엔트리 마크를 이용한다. 링크 포인트는, 본 발명과 관련성이 희박하기 때문에, 설명을 생략한다. 상술한 필드(NumberOfPlayListMarks)는, 엔트리 마크 및 링크 포인트를 합계한 값을 나타낸다.

[0164] 필드(RefToPlayItemID)는, 16비트의 데이터 길이를 가지며, 마크가 찍히는 플레이 아이템을 참조하는 식별 정보(PlayItem_id)가 기술된다. 필드(MarkTimeStamp)는 32비트의 데이터 길이를 가지며, 마크가 찍히는 포인트를 나타내는 타임 스템프가 기술된다. 필드(EntryESPID)는, 16비트의 데이터 길이를 가지며, 마크에 의해 지시되는 엘리멘터리 스트림을 포함하고 있는 TS 패킷의 PID의 값을 나타낸다. 필드(Duration)는, 45kHz의 클록을 단위로 한 계측에 의한, 32비트의 데이터 길이를 갖는 부호없는 정수이다. 이 필드(Duration)에 격납된 값이 "0"이면, 이 필드(Duration)은, 의미를 이루지 않는다.

[0165] 도 15는, 클립 인포메이션 파일의 한 예의 구조를 표시하는 선택스를 도시한다. 필드(TypeIndicator)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 파일이 클립 인포메이션 파일인 것을 나타낸다.

필드(TypeIndicator2)는, 32비트(4바이트)의 데이터 길이를 가지며, 이 클립 인포메이션 파일의 버전을 나타낸다.

[0166] 이 클립 인포메이션 파일은, 블록(blkClipInfo()), 블록(blkSequenceInfo()), 블록(blkProgramInfo()), 블록(blkCPI()), 블록(blkClipMark()) 및 블록(blkExtensionData())을 가지며, 각각 32비트의 데이터 길이를 갖는 필드(SequenceInfoStartAddress), 필드(ProgramInfoStartAddress), 필드(CPIStartAddress), 필드(ClipMarkStartAddress) 및 필드(ExtensionDataStartAddress)는, 각각 대응하는 블록의 시작 어드레스를 나타낸다.

[0167] 필드(ExtensionDataStartAddress)는, 이 클립 인포메이션 파일의 최초의 바이트로부터의 상대바이트수로, 블록(blkExtensionData())의 시작 어드레스를 나타낸다. 상대바이트수는, "0"부터 시작된다. 만약, 이 필드(ExtensionDataStartAddress)의 값이 "0"이면, 이 파일"index.bdmv" 내에, 블록(blkExtensionData())이 존재하지 않는 것을 나타낸다.

[0168] 블록(blkClipInfo())은, 이들의 시작 어드레스를 나타내는 필드에 계속되는, 96비트의 데이터 길이를 갖는 영역(reserved)의 다음부터 시작된다. 블록(blkClipInfo())은, 이 클립 인포메이션 파일이 관리하는 클립 AV 스트림에 관한 정보가 기술된다. 블록(blkSequenceInfo())은, STC나 ATC(어라이벌 타임 베이스)가 연속하고 있는 시퀀스를 정리로서 관리하는 정보가 기술된다. 블록(blkProgramInfo())은, 이 클립 인포메이션 파일에 관리되는 클립 AV 스트림의 부호화 방식, 클립 AV 스트림중의 비디오 데이터의 애스펙트비 등의 정보가 기술된다. 블록(blkCPI())은, 랜덤 액세스 시작점 등의, AV 스트림중의 특징적인 개소를 나타내는 특징점 정보(CPI)에 관한 정보가 격납된다.

[0169] 또한, 블록(blkClipMark())은, 캡터 위치 등의, 클립에 부착된 두출을 위한 인덱스 점(점프 포인트)이 기술된다. 블록(blkExtensionData())은, 확장 데이터를 격납할 수 있는 영역이다. 또한, 이를 블록(blkClipMark()) 및 클립 인포메이션 파일 내의 블록(blkExtensionData())은, 본 발명과의 관련성이 희박하기 때문에, 상세한 설명을 생략한다.

[0170] 도 16은, 블록(blkClipInfo())의 한 예의 구조를 표시하는 선택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkClipInfo())의 최후까지의 데이터 길이를 나타낸다. 16비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 필드(ClipStreamType)가 배치된다.

[0171] 필드(ClipStreamType)는, 8비트의 데이터 길이를 가지며, 클립 AV 스트림의 종류별을 나타낸다. 이 필드(ClipStreamType)의 값은, 예를 들면 "1"로 고정적으로 된다. 필드(ApplicationType)는, 8비트의 데이터 길이를 가지며, 클립 AV 스트림(확장자가 「m2ts」인 파일)이 어떤 다중화에 의해 만들어지고 있는지를 나타낸다. 필드(ApplicationType)의 값이 "1"에서, 대응하는 클립 AV 스트림은, 통상의 동화가 재생된다. 계속해서 31비트의 데이터 길이를 갖는 영역(reserved)이 배치된다.

[0172] 데이터 길이가 1비트인 플래그(IsCC5)는, 플레이 리스트에서의 블록(blkPlayItem())에 의해, 대응하는 클립과 다음의 클립과의 접속을, 상술한 제 1의 심레스 접속, 즉 필드(ConnectionCondition)의 값이 "5"로 나타나는 방법으로 행하는지의 여부를 나타낸다. 플래그(IsCC5)의 값이 "1"(바이너리 값)이면, 클립 사이의 접속이 제 1의 심레스 접속에 의해 이루어지고 있는 것을 나타낸다.

[0173] 필드(TSRecordingRate)는, 클립 AV 스트림 파일의 기록 레이트를 바이트/초로 나타낸 것이다. 필드(NumberOfSourcePackets)는, 클립 AV 스트림에 포함되는 소스 패킷 수를 나타낸다. 1024비트의 데이터 길이의 영역(reserved)을 사이에 두고 블록(TSTypeInfoBlock())이 배치된다. 블록(TSTypeInfoBlock())은, 클립 AV 스트림이 격납되는 패킷의 타입을 나타내는 정보가 격납된다. 이 블록(TSTypeInfoBlock())은, 본 발명과의 관련성이 희박하기 때문에, 상세한 설명을 생략한다.

[0174] 다음의 if문자 이하의 정보는, 상술한 플래그(IsCC5)의 값이 "1"인 경우에 기술된다. if문자의 다음의 8비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(FollowingClipStreamType)가 배치되는 필드(FollowingClipStreamType)는, 8비트의 데이터 길이를 가지며, 이 클립 인포메이션 파일에 대응하는 클립의 다음의 클립의 타입이 기술된다. 32비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(FollowingClipInformationFileName)가 배치된다.

[0175] 필드(FollowingClipInformationFileName)는, 40비트(5바이트)의 데이터 길이를 가지며, 이 클립 인포메이션 파일에 대응하는 클립의 다음의 클립에 대응하는 클립 인포메이션 파일의 파일명이 기술된다. 다음의 필드(ClipCodecIdentifier)는, 32비트(4바이트)의 데이터 길이를 가지며, 해당 다음의 클립의 부호화 방식을 나타낸다.

다. 이 예에서는, 필드(ClipCodecIdentifier)는, ISO646에 규정의 방식으로 부호화된 4문자의 문자열치 "M2TS"에 고정적으로 된다. 다음에 8비트의 데이터 길이를 갖는 영역(reserved)이 배치된다.

[0176] 도 17은, 블록(blkSequenceInfo())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkSequenceInfo())의 최후까지의 데이터 길이를 나타낸다. 15비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 데이터 길이가 1비트로 고정치 "1"이 기술된다.

[0177] 다음의 필드(SPNATCStart)는 32비트의 데이터 길이를 가지며, 연속한 시간에 기록된 것을 나타내는 시퀀스(ATCSequence)라고 부른다)의 시작을 패킷 번호로 나타낸다. 이 도 17의 예에서는, 필드(SPNATCStart)는, 값을 "0"으로 하여 클립 AV 스트림 파일의 선두와 일치시키고 있다. 필드(NumberOfSTCSequence)는, 시퀀스(ATCSequence)상의 시퀀스(STCSequence)의 수를 나타낸다. 필드(NumberOfSTCSequence)는, 값이 "1" 이상이 된다.

[0178] 다음의 for루프문자에 따라, 필드(NumberOfSTCSequence)에서 나타나는 수만큼, 시퀀스(STCSequence)의 정보가 기술된다. 시퀀스(STCSequence)는, MPEG2 TS(Transport Stream)에서의 시간축의 기준인 PCR(Program Clock Reference)가 연속인 범위를 나타낸다. 시퀀스(STCSequence)에는, 클립 내에서 일의적인 번호(STC_id)가 할당된다. 이 시퀀스(STCSequence) 내에서는, 불연속이 없는 일관된 시간축을 정의할 수 있기 때문에, 플레이 아이템의 시작시각 및 종료시각을 일의적으로 정할 수 있다. 즉, 각 플레이 아이템의 시작점과 종료점은, 동일한 시퀀스(STCSequence)에 존재하여야 한다. 이 for루프문자에서는, 값(stc_id)에 의해 시퀀스(STCSequence)가 지정된다.

[0179] 필드(PCRPID[stc_id])는, 16비트의 데이터 길이를 가지며, MPEG2 TS에서, PCR(Program Clock Reference)가 포함되는 TS 패킷의 PID를 나타낸다. 필드(SPNSTCStart[stc_id])는 32비트의 데이터 길이를 가지며, 시퀀스(STCSequence)의 시작을 패킷 번호로 나타낸다. 필드(PresentationStartTime) 및 필드(PresentationEndTime)는, 각각 32비트의 데이터 길이를 가지며, 클립 AV 스트림중의 유효한 범위를 나타낸다. 필드(PresentationStartTime) 및 필드(PresentationEndTime)로 나타나는 범위가 플레이 아이템으로부터 참조할 수 있는 범위가 된다.

[0180] 도 18은, 블록(blkProgramInfo())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkProgramInfo())의 최후까지의 데이터 길이를 나타낸다. 15비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고, 데이터 길이가 1비트로 고정치 "1"이 기술된다.

[0181] 필드(SPNProgramSequenceStart)는 32비트의 데이터 길이를 가지며, 대응하는 클립 AV 스트림 파일에서, 프로그램 시퀀스가 시작되는 소스 패킷의 번호가 기술된다. 필드(ProgramMapPID)는, 16비트의 데이터 길이를 가지며, 프로그램 시퀀스에 적용 가능한 프로그램 맵 섹션을 포함한다고 되어 있는 TS 패킷의 PID의 값을 나타낸다. 필드(NumberOfStreamsInPS)는, 8비트의 데이터 길이를 가지며, 프로그램 시퀀스에 정의되는 엘리멘터리 스트림 수를 나타낸다. 필드(NumberOfStreamsInPS)에 계속해서, 8비트의 데이터 길이를 갖는 영역(reserved)이 배치된다.

[0182] 다음의 for루프문자에 따라, 값([stream_index])을 루프 변수로 하여, 필드(NumberOfStreamsInPS)에서 나타나는 수만큼, 필드(StreamPID[stream_index]) 및 블록(blkStreamCodingInfo(stream_index))의 조(組)가 격납된다. 필드(StreamPID[stream_index])는, 프로그램 시퀀스에 의해 참조된 PMT(ProgramMapTable)에 기술된 엘리멘터리 스트림 대응하는 PID의 값을 나타낸다. 다음의 블록(blkStreamCodingInfo(stream_index))은, 대응하는 필드(StreamPID[stream_index])에서 나타나는 엘리멘터리 스트림 부호화 방식에 관한 정보가 기술된다.

[0183] 도 19는, 블록(blkCPI())의 한 예의 구조를 표시하는 신택스를 도시한다. MPEG 스트림과 같은, 프레임 사이 압축을 행하고 있는 부호화 스트림에서는, 디코드 시작 가능한 개소는, GOP(Group Of Picture)의 선두 등 일부의 개소로 한정되어 있는 것이 많다. CPI(Characteristic Point Information)란, 그 디코드 가능한 시작점의 위치의 정보를 모은 데이터베이스로, 재생시각과, 파일 내 어드레스가 대응시켜진 테이블로 되어 있다. 즉, CPI는, 디코드 단위의 선두 위치를 나타내는 정보가 테이블화되어 있다.

[0184] 이와 같이 데이터베이스를 정함으로써, 예를 들면, 임의의 시작부터 재생하고 싶은 경우, 재생시각을 기초로 CPI를 참조함에 의해 재생 위치의 파일 내 어드레스를 알 수 있다. 이 어드레스는, 디코드 단위의 선두로 되어 있기 때문에, 플레이어는, 그곳부터 데이터를 판독하여 디코드하고, 재빠르게 화상을 표시할 수 있다.

[0185] 또한, 이 CPI에 격납되는, 디코드 단위의 선두 위치(이 예에서는 GOP의 선두 위치)를, EP(Entry Point) 엔트리

라고 칭한다.

[0186] 도 19에서, 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkCPI())의 최후까지의 데이터 길이를 나타낸다. 다음의 if문자에 따라, 필드(Length)의 값이 0이 아니면, 12비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(CPIType)가 배치된다. 필드(CPIType)는, 4비트의 데이터 길이를 가지며, CPI의 종류를 나타낸다. 다음의 블록(blkEPMMap())은, 대응하는 클립 AV 스트림 파일에서의 PTS값과 바이트 어드레스와의 관련시킴을 행하는 테이블이 격납된다.

[0187] 도 20은, 블록(blkEPMMap())의 한 예의 구조를 표시하는 신택스를 도시한다. 8비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(NumberofStreamPIDEEntries)가 배치된다. 필드(NumberofStreamPIDEEntries)는, 8비트의 데이터 길이를 가지며, 블록(blkEPMMap())에서의 블록(blkEPMMapForOneStreamPID)의 엔트리 수를 나타낸다. for루프문자에 따라, 값[k]을 루프 변수로 하여, 필드(NumberofStreamPIDEEntries)에 나타나는 수만큼, 엔트리 포인트에 관한 정보가 기술된다.

[0188] for루프문자 내에서, 필드(StreamPID[k])는, 16비트의 데이터 길이를 가지며, 블록(blkEPMMap())중에서 [k]번째에 엔트리되는 블록(blkEPMMapForOneStreamPID)(이하, [k]번째의 블록(blkEPMMapForOneStreamPID)이라고 기술한다)에 의해 참조되는 엘리멘터리 스트림을 전송하는 트랜스포트 패킷의 PID의 값을 나타낸다.

[0189] 10비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(EPStreamType[k])가 배치된다. 필드(EPStreamType[k])는, 4비트의 데이터 길이를 가지며, [k]번째의 블록(blkEPMMapForOneStreamPID)에 의해 참조되는 엘리멘터리 스트림의 타입을 나타낸다. 필드(NumberofEPCoarseEntries[k])는, 16비트의 데이터 길이를 가지며, [k]번째의 블록(blkEPMMapForOneStreamPID)중에 있는 거친 검색용의 서브테이블(EP coarse table)의 엔트리 수를 나타낸다. 필드(NumberofEPFineEntries[k])는, 18비트의 데이터 길이를 가지며, [k]번째의 블록(blkEPMMapForOneStreamPID)중에 있는 정밀한 검색용의 서브테이블(EP fine table)의 엔트리 수를 나타낸다. 필드(EPMMapForOneStreamPIDStartAddress[k])는 32비트의 데이터 길이를 가지며, 블록(blkEPMMap())중에서 [k]번째의 블록(blkEPMMapForOneStreamPID)이 시작되는 상대바이트 위치를 나타낸다. 이 값은, 블록(blkEPMMap())의 제1바이트째부터의 바이트수로 나타난다.

[0190] 상술한 for루프문자에 의한 기술의 후, 16비트의 정수배의 데이터 길이를 갖는 패딩 워드를 끼우고 기술되는 for루프문자에 따라, 값[k]을 루프 변수로 하여, 필드(NumberofStreamPIDEEntries)에 나타나는 수만큼, 블록(blkEPMMapForOneStreamPID(EPStreamType[k], NumberofEPCoarseEntries[k], NumberofEPFineEntries[k]))이 격납된다. 즉, 인수(NumberofEPCoarseEntries[k])는, 서브테이블(EP coarse table)에 격납되는 엔트리(PTSEPCoarse) 및 엔트리(SPNEPCoarse)의 수를 나타낸다. 마찬가지로, 인수(NumberofEPFineEntries[k])는, 서브테이블(EP fine table)에 격납되는 엔트리(PTSEPFine) 및 엔트리(SPNEPFine)의 수를 나타낸다. 이하에서는, 인수(NumberofEPCoarseEntries[k]) 및 인수(NumberofEPFineEntries[k])를, 각각 적절히, 엔트리 수(Nc) 및 엔트리 수(Nf)라고 부른다.

[0191] 도 21은, 블록(blkEPMMapForOneStreamPID(EP_stream_type, Nc, Nf))의 한 예의 구조를 표시하는 신택스를 도시한다. 블록(blkEPMMapForOneStreamPID(EP_stream_type, Nc, Nf))의 시멘틱스를 설명하기 위해, 우선, 블록(blkEPMMapForOneStreamPID(EP_stream_type, Nc, Nf))에 격납되는 데이터의 기초가 되는 엔트리인, 엔트리(PTSEPStart) 및 엔트리(SPNEPStart)의 의미에 관해 설명한다.

[0192] 엔트리(PTSEPStart)와, 엔트리(PTSEPStart)에 관련시켜진 엔트리(SPNEPStart)는, 각각 AV 스트림상의 엔트리 포인트를 가리킨다. 그리고, 엔트리(PTSEPFine)와, 엔트리(PTSEPFine)에 관련시켜진 엔트리(PTSEPCoarse)는, 동일한 엔트리(PTSEPStart)로부터 도출된다. 또한, 엔트리(SPNEPFine)와, 엔트리(SPNEPFine)에 관련시켜진 엔트리(SPNEPCoarse)는, 동일한 엔트리(SPNEPStart)로부터 도출된다.

[0193] 도 22는, 엔트리(PTSEPCoarse) 및 엔트리(PTSEPFine)의 한 예의 포맷에 관해 도시한다. PTS 즉 엔트리(PTSEPStart)는, 데이터 길이가 33비트인 값이다. MSB의 비트를 제 32비트, LSB의 비트를 제 0비트라고 할 때, 이 도 22의 예에서는, 대략적인 단위로 검색을 행할 때에 이용되는 엔트리(PTSEPCoarse)는, 엔트리(PTSEPStart)의 제 32비트부터 제 19비트까지의 14비트가 이용된다. 엔트리(PTSEPCoarse)에 의해, 해상도가 5.8초이고, 26.5시간까지의 범위에서 검색이 가능하다. 또한, 보다 정밀한 검색을 행하기 위한 엔트리(PTSEPFine)는, 엔트리(PTSEPStart)의 제 19비트부터 제 9비트까지의 11비트가 이용된다. 엔트리(PTSEPFine)에 의해, 해상도가 5.7밀리초이고, 11.5초까지의 범위에서 검색이 가능하다. 또한, 제 19비트는, 엔트리(PTSEPCoarse)와 엔트리(PTSEPFine)에서 공통으로 이용된다. 또한, LSB측의 제 0비트부터 제 8비트까지의 9비트는, 이용되지 않는다.

- [0194] 도 23은, 엔트리(SPNEPCoarse) 및 엔트리(SPNEPFine)의 한 예의 포맷에 관해 도시한다. 소스 패킷 번호 즉 엔트리(SPNEPStart)는, 데이터 길이가 32비트인 값이다. MSB의 비트를 제 31비트, LSB의 비트를 제 0비트라고 할 때, 이 도 23의 예에서는, 대략적인 단위로 검색을 행할 때에 이용되는 엔트리(SPNEPCoarse)는, 엔트리(SPNEPStart)의 제 31비트부터 제 0비트까지의 모든 비트가 이용된다. 또한, 보다 정밀한 검색을 행하기 위한 엔트리(SPNEPFine)는, 엔트리(SPNEPStart)의 제 16비트부터 제 0비트까지의 17비트가 이용된다. 엔트리(SPNEPFine)에 의해, 예를 들면 약 25MB(Mega Byte)의 AV 스트림 파일까지의 범위에서, 검색이 가능하다.
- [0195] 또한, 소스 패킷 번호의 경우에도, 엔트리(SPNEPCoarse)로서 MSB측의 소정비트 수의 값만 이용하도록 하여도 좋다. 예를 들면, 엔트리(SPNEPCoarse)로서, 엔트리(SPNEPStart)의 제 31비트부터 제 16비트까지의 17비트를 이용하고, 엔트리(SPNEPFine)는, 엔트리(SPNEPStart)의 제 16비트부터 제 0비트까지의 17비트를 이용한다.
- [0196] 상술한 것에 의거하여, 엔트리(PTSEPStart) 및 엔트리(SPNEPStart)는, 다음과 같이 정의된다.
- [0197] 엔트리(PTSEPStart)는, 도 22에서 도시한 바와 같이, 데이터 길이가 33비트의 부호없는 정수이고, AV 스트림중에서, 랜덤 액세스가 가능한 픽처(예를 들면 IDR(InstantaneousDecodingRefresh)픽처나 I(Intra)픽처)로부터 시작하는 비디오 액세스 유닛의 33비트 길이의 PTS를 나타낸다.
- [0198] 엔트리(SPNEPStart)는, 도 23에서 도시한 바와 같이, 32비트의 부호없는 정수이고, 엔트리(PTSEPStart)에 관련시켜진 비디오 액세스 유닛의 제 1바이트째를 포함하는 소스 패킷의, AV 스트림중에서의 어드레스를 나타낸다. 엔트리(SPNEPStart)는, 소스 패킷 번호의 단위로 표시되고, AV 스트림 파일중의 최초의 소스 패킷부터, 값"0"을 초기치로 하여, 소스 패킷마다 하나씩 증가하는 값으로서 카운트된다.
- [0199] 도 21을 참조하면, 블록(blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf))은, 제 1의 for루프문자에 의해 대략적인 단위로의 검색을 행하기 위한 서브테이블(EP coarse table)이 기술되고, 제 2의 for루프문자에 의해 서브테이블(EP coarse table)의 검색 결과에 의거하여 보다 상세한 검색을 행하기 위한 서브테이블(EP fine table)이 기술된다.
- [0200] 제 1의 for루프문자의 직전에, 필드(EPFineTableStartAddress)가 배치된다. 필드(EPFineTableStartAddress)는 32비트의 데이터 길이를 가지며, 최초의 제 2의 for루프에서의 필드(ReservedEPFine[EP_fine_id])의 제 1바이트째의 시작 어드레스를, 블록(blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf))의 제 1바이트째부터의 상대바이트수로 나타낸다. 상대바이트수는, 값"0"부터 시작한다.
- [0201] 제 1의 for루프문자는, 루프 변수[i]로써, 서브테이블(EP coarse table)의 엔트리 수(Nc)까지 반복되고, 엔트리 수(Nc)의 조(組) 수만큼 필드(RefToEPFineID[i]), 엔트리(PTSEPCoarse[i]) 및 엔트리(SPNEPCoarse[i])가 격납된다. 제 1의 for루프문자에서, 필드(RefToEPFineID[i])는, 18비트의 데이터 길이를 가지며, 필드(RefToEPFineID[i])에 계속된 필드(PTSEPCoarse[i])가 나타내는 엔트리(PTSEPCoarse)에 관련시켜지는 엔트리(PTSEPFine)를 갖는, 서브테이블(EP fine table) 내의 엔트리 번호를 나타낸다. 엔트리(PTSEPFine)와, 이 엔트리(PTSEPFine)에 관련시켜지는 엔트리(PTSEPCoarse)는, 동일한 엔트리(PTSEPStart)로부터 도출된다. 필드(RefToEPFineID[i])는, 제 2의 for루프문자중에서 기술된 순번으로 정의되는 루프 변수([EP_fine_id])의 값에 의해 주어진다.
- [0202] 제 1의 for루프문자의 후에, 패딩 워드를 끼우고 제 2의 for루프문자에 의한 기술이 이루어진다. 제 2의 for루프문자는, 루프 변수([EP_fine_id])로써, 서브테이블(EP fine table)의 엔트리 수(Nf)까지 반복되고, 엔트리 수(Nf)의 조 수만큼, 1비트의 데이터 길이를 갖는 필드(ReservedEPFine[EP_fine_id])와, 3비트의 데이터 길이를 갖는 필드(IEndPositionOffset[EP_fine_id])와, 11비트의 데이터 길이를 갖는 필드(PTSEPFine[EP_fine_id])와, 17비트의 데이터 길이를 갖는 필드(SPNEPFine[EP_fine_id])가 격납된다. 이 중, 필드(PTSEPFine[EP_fine_id]) 및 필드(SPNEPFine[EP_fine_id])는, 루프 변수([EP_fine_id])에 의거하여 서브테이블(EP fine table)로부터 참조되는 엔트리(PTSEPFine) 및 엔트리(SPNEPFine) 각각이 격납된다.
- [0203] 엔트리(PTSEPCoarse) 및 엔트리(PTSEPFine), 및, 엔트리(SPNEPCoarse) 및 엔트리(SPNEPFine)는, 다음과 같이 도출된다. 서브테이블(EP fine table)에, 관련되는 데이터(SPNEPStart)의 값의 승순(昇順)으로 나열되어 있는 Nf개의 엔트리가 있다고 한다. 각각의 엔트리(PTSEPFine)는, 대응하는 엔트리(PTSEPStart)로부터, 다음 식(1)과 같이 도출된다.
- [0204]
$$\text{PTSEPFine[EP_fine_id]} = (\text{PTSEPStart[EP_fine_id]} \gg 9) / 2^{11} \quad \dots \quad (1)$$

- [0205] 엔트리(PTSEPCoarse)와, 대응하는 엔트리(PTSEPFine)와의 관계는, 다음 식(2), (3)과 같다.
- [0206] $\text{PTSEPCoarse}[i] = (\text{PTSEPStart}[\text{RefToEPFineID}[i]] >> 19) / 2^{14} \dots (2)$
- [0207] $\text{PTSEPFine}[\text{RefToEPFineID}[i]] = (\text{PTSEPStart}[\text{RefToEPFineID}[i]] >> 9) / 2^{11} \dots (3)$
- [0208] 각각의 엔트리(SPNEPFine)는, 대응하는 엔트리(SPNEPStart)로부터, 다음 식(4)과 같이 도출된다.
- [0209] $\text{SPNEPFine}[\text{EP_fine_id}] = \text{SPNEPStart}[\text{EP_fine_id}] / 2^{17} \dots (4)$
- [0210] 엔트리(SPNEPCoarse)와, 대응하는 엔트리(SPNEPFine)와의 관계는, 다음 식(5), (6)과 같다.
- [0211] $\text{SPNEPCoarse}[i] = \text{SPNEPStart}[\text{RefToEPFineID}[i]] \dots (5)$
- [0212] $\text{SPNEPFine}[\text{RefToEPFineID}[i]] = \text{SPNEPStart}[\text{RefToEPFineID}[i]] / 2^{17} \dots (6)$
- [0213] 또한, 상술한 식(1) 내지 (6)에서, 기호 「 $>>x$ 」 는, 데이터의 LSB측에서 x비트를 초과하는 자릿수로부터 비트를 이용하는 것을 의미한다.
- [0214] 다음에, 확장 데이터를 격납하기 위한 블록(blkExtensionData())에 관해 설명한다. 이 블록(blkExtensionData())은, 소정의 확장 데이터를 격납 가능하게 정의되고, 인덱스 테이블이 격납되는 파일"index.bdmv", 플레이 리스트가 격납되는 파일"xxxxx.mpls" 및 클립 인포메이션 파일"zzzzz.clpi"의 각 파일에 기술할 수 있다.
- [0215] 도 24은, 블록(blkExtensionData())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkExtensionData())의 끝까지의 데이터 길이를 바이트수로 나타낸다. 이 필드(Length)의 나타내는 데이터 길이가 "0"이 아니면, if문자 이하의 기술이 이루어진다.
- [0216] 필드(DataBlockStartAddress)는 32비트의 데이터 길이를 가지며, 이 신택스중의, 확장 데이터의 본체가 격납되는 블록(DataBlock())의 시작 어드레스를, 이 블록(blkExtensionData())의 선두바이트로부터의 상대바이트수로 나타낸다. 즉, 상대바이트수는, "0"부터 시작된다. 또한, 필드(DataBlockStartAddress)는, 다음에 나타내는 32비트 얼라인먼트의 조건을 충족시켜야 한다.
- [0217] DataBlockStartAddress%4=0
- [0218] 24비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(NumberOfExtDataEntries)가 배치된다. 필드(NumberOfExtDataEntries)는, 8비트의 데이터 길이를 가지며, 이 블록(blkExtensionData())의 블록(DataBlock())에 격납되는 확장 데이터의 엔트리 수를 나타낸다. 확장 데이터의 엔트리는, 확장 데이터의 본체를 취득하기 위한 정보가 격납된다. 이 예에서는, 확장 데이터의 엔트리는, 필드(ExtDataType), 필드(ExtDataVersion), 필드(ExtDataStartPosition) 및 필드(ExtDataLength)로 이루어지는 블록(ext_data_entry())이고, 블록(blkExtensionData())에서, 제 1의 for루프문자에 따라 이 필드(NumberOfExtDataEntries)에 나타나는 개수만큼, 이 블록(ext_data_entry())이 존재한다.
- [0219] 필드(ExtDataType)는, 16비트의 데이터 길이를 가지며, 이 블록(blkExtensionData())에 기술되는 확장 데이터가 기록 장치용의 확장 데이터인 것을 나타낸다. 이 필드(ExtDataType)의 값은, 확장 데이터를 식별하는 제 1의 값이고, 이 블록(blkExtensionData())을 포함하는 규격서의 라이센서(이용 인가자)가 할당된다고 정의할 수 있다. 필드(ExtDataVersion)는, 확장 데이터를 식별하는 제 2의 값이고, 이 확장 데이터의 버전 번호를 나타내는 것으로 정의할 수 있다. 또한, 이 블록(blkExtensionData())에서, 필드(ExtDataType) 및 필드(ExtDataVersion)의 값이 동일한 블록(ext_data_entry())이 2 이상, 존재해서는 않된다.
- [0220] 필드(ExtDataStartPosition)는 32비트의 데이터 길이를 가지며, 이 필드(ExtDataStartPosition)가 포함되는 확장 데이터의 엔트리(블록(ext_data_entry()))에 대응하는 확장 데이터의 시작 어드레스를 나타낸다. 필드(ExtDataStartPosition)는, 블록(blkExtensionData())의 선두바이트로부터의 상대바이트수로, 확장 데이터(ext_data)의 시작 어드레스를 나타낸다. 또한, 필드(ExtDataStartPosition)는, 다음에 나타내는 32비트 얼라인먼트의 조건을 충족시켜야 한다.
- [0221] ExtDataStartPosition%4=0
- [0222] 필드(ExtDataLength)는 32비트의 데이터 길이를 가지며, 이 필드(ExtDataStartPosition)가 포함되는 확장 데이터

터의 엔트리(블록(ext_data_entries()))에 대응하는 확장 데이터의 데이터 길이를 나타낸다. 데이터 길이는, 바이트수로 나타난다.

[0223] 필드(NumberOfExtDataEntries)에서 나타낸 개수만큼, 확장 데이터의 엔트리(블록(ext_data_entry()))가 기술되면, 각각 16비트의 데이터 길이를 갖고 임의의 데이터열로 이루어지는 필드(padding_word)가, 2필드를 조로 하여 임의의 회수(L1)만큼 반복된다. 그 후, 확장 데이터의 본체가 격납되는 블록(DataBlock())이 기술된다. 블록(DataBlock())은, 하나 이상의 확장 데이터가 격납된다. 각각의 확장 데이터(ext_data)는, 상술한 필드(ExtDataStartAddress) 필드(ExtDataLength)에 의거하여, 블록(DataBlock())으로부터 취출된다.

[0224] 도 25는, 블록(blkExtensionData())에서의 각 데이터의 참조 관계를 모식적으로 도시한다. 필드(Length)에 의해, 필드(Length) 직후의 위치부터 블록(blkExtensionData())의 최후까지의 데이터 길이가 나타난다. 필드(DataBlockStartAddress)에 의해, 블록(DataBlock())의 시작 위치가 나타난다. 필드(NumberOfExtDataEntries)에서 나타나는 개수만큼, 블록(ext_data_entry())이 기술된다. 최후의 블록(ext_data_entry)부터 블록(DataBlock()) 사이에는, 임의의 길이로 필드(padding_word)가 놓여진다.

[0225] 블록(DataBlock()) 내에는, 블록(ext_data_entry())에서 나타나는 확장 데이터(ext_data)가 놓여진다. 각각의 확장 데이터(ext_data)의 위치 및 데이터 길이는, 대응하는 블록(ext_data_entry()) 내의 필드(ExtDataStartAddress) 및 필드(ExtDataLength)에 의해 나타난다. 따라서 블록(DataBlock()) 내에서의 확장 데이터(ext_data)의 나열 순서는, 대응하는 블록(ext_data_entry())의 나열 순서와 일치하여 있지 않아도 좋다.

[0226] 이와 같이, 확장 데이터를, 확장 데이터의 본체가 격납되는 블록(DataBlock())과, 블록(DataBlock()) 내의 확장 데이터에 대한 액세스 정보 등이 격납되는 블록(ext_data_entry())에 의한 2층구조로 함으로써, 복수의 확장 데이터를 격납하는 것이 가능해진다.

[0227] 다음에, 상술한 확장 데이터의 한 예의 작성 방법 및 관독 방법에 관해 설명한다. 도 26은, 블록(blkExtensionData())에 데이터를 기록할 때의 한 예의 처리를 도시하는 플로우 차트이다. 이 도 26은, 블록(blkExtensionData())중의 (n+1)번째의 엔트리로서, 확장 데이터를 추가하고, 블록(blkExtensionData())을 재기록하는 경우의 예이다.

[0228] 우선, 스텝 S10에서, 기록하고자 하고 있는 확장 데이터의 데이터 길이를 취득하고, 필드(ExtDataLength[n+1])의 값을 세트한다. 또한, 「[n+1]」의 기술은, (n+1)번째의 엔트리의 번호에 대응한다. 다음에, 스텝 S11에서, 현재의 블록(blkExtensionData())에 열거되어 있는 블록(ext_data_entry())의 필드(ExtDataLength) 및 필드(ExtDataStartAddress)의 값을 조사하고, 블록(DataBlock())의 이용 상황을 취득한다.

[0229] 그리고, 다음의 스텝 S12에서, 블록(DataBlock())중에, 기록하고자 하고 있는 확장 데이터의 데이터 길이인 필드(ExtDataLength[n+1])에 나타나는 데이터 길이 이상의, 연속한 빈 영역이 있는지의 여부가 판단된다. 만약, 있다고 판단되면, 처리는 스텝 S14로 이행된다.

[0230] 한편, 필드(ExtDataLength[n+1])에 나타나는 데이터 길이 이상이 연속한 빈 영역이 없다고 판단되면, 처리는 스텝 S13으로 이행되고, 블록(blkExtensionData())에서의 필드(Length)의 값을 크게 하고, 필드(ExtDataLength[n+1])에 나타나는 데이터 길이 이상이 연속한 빈 영역을 블록(DataBlock()) 내에 만든다. 빈 영역이 만들어졌으면, 처리가 스텝 S14로 이행된다.

[0231] 스텝 S14에서는, 확장 데이터를 기록하는 영역의 선두 어드레스를 정하고, 그 선두 어드레스의 값을 필드(ExtDataStartAddress[n+1])로 한다. 다음의 스텝 S15에서, 필드(ExtDataStartAddress[n+1])로부터, 상술한 스텝 S10에서 세트된 필드(ExtDataLength[n+1])의 길이의 확장 데이터(ext_data)[n+1]를 기록한다.

[0232] 데이터의 기록이 종료되면, 스텝 S16에서, 블록(ext_data_entry())에 대해, 필드(ExtDataLength[n+1])와, 필드(ExtDataStartAddress[n+1])를 추가한다.

[0233] 또한, 상술한 것에 있어서, 재기록을 행하는 블록(blkExtensionData())은, 이미 디스크 등의 기록 매체로부터 판독되어 기록 장치의 메모리에 기억되어 있는 것으로 한다. 그 때문에, 스텝 S13에서의, 필드(Length)의 값의 변경에 의한 블록(blkExtensionData())의 확대는, 시스템에 맡겨지고, 시스템이 메모리 앤로케이션을 적절하게 행함으로써 이루어진다.

[0234] 도 27은, 블록(blkExtensionData())으로부터 확장 데이터를 판독할 때의 한 예의 처리를 도시하는 플로우 차트이다. 또한, 이 도 27의 플로우 차트에 의한 처리는, 재생 전용의 기록 매체와, 기록 가능한 기록 매체와의 양쪽에 적용 가능한 것이다. 우선, 최초의 스텝 S20에서, 판독하고자 하는 확장 데이터가 준거하는 규격으로부터,

필드(ExtDataType)의 값을 취득하고, 스텝 S21에서, 판독하고자 하는 확장 데이터의 종류별로부터, 필드(ExtDataVersion)의 값을 취득한다.

- [0235] 다음의 스텝 S22에서, 블록(blkExtensionData())에 열거되어 있는 블록(ext_data_entry())을 하나씩 순서로, 판독한다. 그리고, 스텝 S23에서, 판독한 블록(ext_data_entry())에 포함되는 필드(ExtDataType) 및 필드(ExtDataVersion)의 값이, 상술한 스텝 S20 및 스텝 S21에서 취득한 필드(ExtDataType) 및 필드(ExtDataVersion)의 값과 일치하는지의 여부가 판단된다.
- [0236] 일치하지 않는다고 판단되면, 처리는 스텝 S26으로 이행되고, 블록(blkExtensionData()) 내에 열거되는 블록(ext_data_entry())을 전부 다 판독하였는지 여부가 판단된다. 전부 다 판독하였다고 판단되면, 처리는 스텝 S27로 이행되고, 이 블록(blkExtensionData())에는, 판독하고자 하는 확장 데이터가 존재하지 않는다고 하여, 일련의 처리가 종료된다. 전부 다 판독하지 않았다고 판단되면, 처리는 스텝 S22로 되돌아와, 다음의 블록(ext_data_entry())이 판독된다.
- [0237] 상술한 스텝 S23에서, 블록(ext_data_entry())에 포함되는 필드(ExtDataType) 및 필드(ExtDataVersion)의 값이, 취득한 필드(ExtDataType) 및 필드(ExtDataVersion)의 값과 일치하다고 판단되면, 처리는 스텝 S24로 이행된다. 여기서는, 블록(blkExtensionData())중의 [i]번째의 엔트리에서 일치한 것으로 한다.
- [0238] 스텝 S24에서는, [i]번째의 엔트리의 블록(ext_data_entry())으로부터 필드(ExtDataLength[i])의 값과, 필드(ExtDataStartAddress[i])의 값을 판독한다. 그리고, 스텝 S25에서, 스텝 S24에서 판독한 필드(ExtDataStartAddress[i])에서 나타나는 어드레스로부터, 필드(ExtDataLength[i])에서 나타나는 데이터 길이만큼, 데이터를 판독한다.
- [0239] 다음에, 상술한, 인덱스 파일"index.bdmv", 무비 오브젝트 파일"MovieObject.bdmv", 플레이 리스트 파일"xxxxx.mpls" 및 클립 인포메이션 파일"zzzzz.clpi"에 각각 정의(定義) 가능한, 확장 데이터를 격납하는 확장 데이터 블록(blkExtensionData())에 대해 설명한다.
- [0240] 우선, 인덱스 파일"index.bdmv"에 대해 정의되는 한 예의 확장 데이터 블록에 대해 설명한다. 여기서는, 플레이 리스트마다 기록 가능한 기록 매체에 특유한 속성 정보를 부가하도록 하였다, 한 예의 확장 데이터 블록에 대해 설명한다. 도 28은, 이 플레이 리스트 속성을 기술하기 위한, 파일"index.bdmv" 내의 필드(blkExtensionData())에서의 블록(DataBlock())(도 24 참조)의 한 예의 구조를 표시하는 신택스를 도시한다. 이 도 28의 예에서는, 블록(DataBlock())이 블록(blkIndexExtensionData())으로서 기술되어 있다.
- [0241] 우선, 상술한 도 24를 참조하여, 블록(blkExtensionData())에서 필드(ExtDataType)를 값"0x1000", 필드(ExtDataVersion)를 값"0x0100"으로 한다. 이를 필드(ExtDataType) 및 필드(ExtDataVersion)에 기술된 값은, 예를 들면 재생 장치측에서, 미리 ROM(Read Only Memory) 등에 기억된 테이블이 참조되어 식별된다. 블록(DataBlock()) 내의 필드(ExtDataStartAddress) 및 필드(ExtDataLength)로 나타나는 영역에, 블록(blkIndexExtensionData())이 격납된다.
- [0242] 블록(blkIndexExtensionData())에서, 필드(TypeIndicator)는, 다음에 계속된 데이터의 종류를 나타낸다, ISO646에 규정된 부호화 방식으로 부호화한 4문자로 이루어지는 문자열이 기술된다. 이 도 28의 예에서는, 필드(TypeIndicator)에 ISO646에 규정의 방식으로 부호화된 4문자의 문자열"INDEX"이 기술되고, 다음에 계속된 데이터 종류가 인덱스 파일에서의 확장 데이터인 것이 나타난다.
- [0243] 필드(TypeIndicator)에 계속해서 32비트의 데이터 길이를 갖는 영역(reserved)이 배치되고, 그 다음에, 32비트의 데이터 길이를 갖는 필드(TableOfPlayListStartAddress)가 배치된다. 필드(TableOfPlayListStartAddress)는, 블록(blkTableOfPlayList())의, 이 블록(blkIndexExtensionData()) 선두를 기준으로 한 시작 어드레스가 나타난다.
- [0244] 필드(TableOfPlayListStartAddress)의 다음에, 32비트의 데이터 길이를 갖는 필드(MakersPrivateDataStartAddress)가 배치되고 블록(blkMakersPrivateData())의 이 블록(blkIndexExtensionData()) 선두를 기준으로 한 시작 어드레스가 나타나고, 192비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 블록(blkUIAppInfoAVCHD())이 배치된다. 16비트의 데이터 길이를 갖는 패딩 워드(padding_word)가 값(N1)으로 나타나는 회수만큼 반복되고, 다음에, 블록(blkTableOfPlayLists())이 배치된다. 또한 계속해서, 16비트의 데이터 길이를 갖는 패딩 워드(padding_word)가 값(N2)으로 나타나는 회수만큼 반복되고, 다음에 블록(blkMakersPrivateData())이 배치된다. 이 블록(blkMakersPrivateData())의 후에, 16비트의 데이터 길이를 갖는 필드(TableOfPlayListStartAddress)가 배치된다.

이터 길이를 갖는 패딩 워드(padding_word)가 값(N3)으로 나타나는 회수만큼 반복된다.

[0245] 또한, 블록(blkUIAppInfoAVCHD()) 및 블록(blkMakersPrivateData())은, 본 발명과 관련성이 희박하기 때문에, 설명을 생략한다.

[0246] 도 29는, 상술한 블록(blkTableOfPlayLists())의 한 예의 구조를 표시하는 신택스를 도시한다. 필드(Length)는 32비트의 데이터 길이를 가지며, 이 필드(Length)의 직후부터 블록(blkTableOfPlayLists())의 최후의 바이트까지의 데이터 길이를 바이트수로 나타낸다. 필드(Length)에 계속해서, 재생 타이틀을 재생하기 위한 플레이 리스트에 관한 정보가 기술되는 블록(blkFirstPlaybackTitlePlayLists())과, 메뉴 타이틀에 관한 정보가 기술되는 블록(blkMenuTitlePlayLists())이 배치된다. 이를 블록(blkFirstPlaybackTitlePlayLists()) 및 블록(blkMenuTitlePlayLists())은, 본 발명과 관련성이 희박하기 때문에, 설명을 생략한다.

[0247] 다음에, 16비트의 데이터 길이를 갖는 필드(NumberOfTitlePlaylistPair)가 배치된다. 필드(NumberOfTitlePlaylistPair)는, 플레이백 타이틀 및 메뉴 타이틀 이외의 타이틀을 재생하기 위한 플레이 리스트의 수가 기술된다. 다음의 for루프문자에 따라, 필드(NumberOfTitlePlaylistPair)에서 나타나는 수만큼, 블록(blkMovieTitlePlaylistPair())이 기술된다. 블록(blkMovieTitlePlaylistPair())은, 필드(PlaylistFileName), 필드(PlaylistAttribute) 및 필드(RefToTitleID)를 포함한다. 즉, 블록(blkMovieTitlePlaylistPair())은, 이 for루프문자로 나타나는 [i]번째의 플레이 리스트에 관해, 해당 플레이 리스트의 파일명, 해당 플레이 리스트에 부여된 속성, 및, 해당 플레이 리스트의 참조 타이틀 ID로 이루어지는 플레이 리스트의 정보를 구조화한 것이다.

[0248] 이 for루프문자에 의한 나열 순서는, 기록순이 된다. 즉, 하나의 플레이 리스트가 추가되면, 필드(NumberOfTitlePlaylistPair)의 값이 "1"만큼 잉크리먼트되고, 기존의 플레이 리스트의 정보의 뒤에, 추가된 플레이 리스트의 정보가 추가 기록된다.

[0249] 필드(PlaylistFileName)는, 40비트(5바이트)의 데이터 길이를 가지며, 플레이 리스트의 파일명이 ISO646로 규정된 부호화 방식으로 부호화되어 기술된다. 필드(PlaylistFileName)의 다음에, 6비트의 데이터 길이를 갖는 영역(reserved)을 사이에 두고 필드(PlaylistAttribute)가 배치된다. 필드(PlaylistAttribute)는, 2비트의 데이터 길이를 가지며, 해당 플레이 리스트에 부여된 속성을 나타낸다. 플레이 리스트는, 그 성인(成因)에 의거하여, 클립의 생성과 함께 생성되는 플레이 리스트에 대응하는 제 1의 종류와, 기존의 타이틀 또는 플레이 리스트의 일부 또는 전부를 이용하여 작성되는 플레이 리스트에 대응하는 제 2의 종류와, 메뉴를 재생하기 위해 이용하는 제 3의 종류의 3종류로 나눠지고, 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.

[0250] 또한, 이하에서는 적절히, 속성「Real」이 부여된 플레이 리스트를 리얼 플레이 리스트, 속성「Virtual」이 부여된 플레이 리스트를 베추얼 플레이 리스트, 속성「Menu」이 부여된 플레이 리스트를 메뉴 플레이 리스트라고 부른다.

[0251] 필드(RefToTitleID)는, 동일 루프 내의 필드(PlaylistFileName)에 나타나는 플레이 리스트가 작성시에 속하는 타이틀의 ID(번호)가 기술된다. 보다 구체적인 예로서는, 인덱스 파일"index.bdmv" 내의 블록(blkIndexes())에서의, 대응하는 값(title_id)이 기술된다. 또한, 해당 플레이 리스트가 퍼스트 플레이백 타이틀만으로부터 재생되는 경우, 필드(RefToTitleID)의 값은, 제 1의 고정치, 예를 들면 "0xFFFF"가 된다. 또한, 해당 플레이 리스트가 메뉴 타이틀만으로부터 재생되는 경우는, 필드(RefToTitle)의 값은, 제 2의 고정치, 예를 들면 "0xFFE"가 된다.

[0252] 다음에, 가상 플레이어에 관해, 개략적으로 설명한다. 상술한 바와 같은 데이터 구조를 갖는 디스크가 플레이어에 장전되면, 플레이어는, 디스크로부터 판독된 무비 오브젝트 등에 기술된 커맨드를, 플레이어 내부의 하드웨어를 제어하기 위한 고유의 커맨드로 변환할 필요가 있다. 플레이어는, 이와 같은 변환을 행하기 위한 소프트웨어를, 플레이어에 내장되는 ROM(Read Only Memory)에 미리 기억하고 있다. 이 소프트웨어는, 디스크와 플레이어를 중개하여 플레이어에 AVCHD 포맷의 규정에 따른 동작을 시키기 때문에, 가상 플레이어라고 칭하여진다.

[0253] 도 30A 및 도 30B는, 이 가상 플레이어의 동작을 개략적으로 도시한다. 도 30A는, 디스크의 로딩시의 동작의 예를 도시한다. 디스크가 플레이어에 장전되고 디스크에 대한 이니셜 액세스가 이루어지면(스텝 S30), 하나의 디스크에서 공유적으로 이용되는 공유 파라미터가 기억되는 레지스터가 초기화된다(스텝 S31). 그리고, 다음의 스텝 S32에서, 무비 오브젝트 등에 기술된 프로그램이 디스크로부터 판독되어 실행된다. 또한, 이니셜 액세스는, 디스크 장전시와 같이, 디스크의 재생이 처음으로 행하여지는 것을 말한다.

- [0254] 도 30B는, 플레이어가 정지 상태로부터 유저에 의해 예를 들면 플레이 키가 압하되어 재생이 지시된 경우의 동작의 예를 도시한다. 최초의 정지 상태(스텝 S40)에 대해, 유저에 의해, 예를 들면 리모트컨트롤 커맨더 등을 이용하여 재생이 지시된다(UO : User Operation). 재생이 지시되면, 우선, 레지스터 즉 공통 파라미터가 초기화되고(스텝 S41), 다음의 스텝 S42에서, 무비 오브젝트 실행 페이즈로 이행한다.
- [0255] 무비 오브젝트의 실행 페이즈에서의 플레이 리스트의 재생에 관해, 도 31을 이용하여 설명한다. UO 등에 의해, 타이틀 번호#1의 컨텐츠를 재생 시작하는 지시가 있는 경우에 관해 생각한다. 플레이어는, 컨텐츠의 재생 시작 지시에 따라, 상술한 도 2에 도시되는 인덱스 테이블(Index Table)을 참조하여, 타이틀#1의 컨텐츠 재생에 대응하는 오브젝트의 번호를 취득한다. 예를 들면 타이틀#1의 컨텐츠 재생을 실현하는 오브젝트의 번호가 #1이었다고 하면, 플레이어는, 무비 오브젝트#1의 실행을 시작한다.
- [0256] 이 도 31의 예에서는, 무비 오브젝트#1에 기술된 프로그램은 2행으로 이루어지고, 1행째의 커맨드가 "Play PlayList(1)"였다고 하면, 플레이어는, 플레이 리스트#1의 재생을 시작한다. 플레이 리스트#1은, 하나 이상의 플레이 아이템으로 구성되고, 플레이 아이템이 순서로 재생된다. 플레이 리스트#1중의 플레이 아이템의 재생이 종료되면, 무비 오브젝트#1의 실행으로 되돌아와, 2행째의 커맨드가 실행된다. 도 31의 예에서는, 2행째의 커맨드가 "jump MenuTitle"이고, 이 커맨드가 실행되어 인덱스 테이블에 기술된 메뉴 타이틀(MenuTitle)을 실현하는 무비 오브젝트의 실행이 시작된다.
- [0257] 다음에, 본 발명의 실시의 한 형태에 관해 설명한다. 본 발명에서는, 비디오 데이터 및 오디오 데이터의 기록에 즈음하여, 비디오 데이터 및 오디오 데이터가 격납되는 파일의 정보와 시스템상의 소정의 제약을 비교하고, 비교 결과에 의거하여, 기록의 계속에 즈음하여 해당 파일을 분할하는지의 여부를 판단한다. 예를 들면, 해당 파일의 정보가 시스템상의 소정의 제약을 초과한다고 판단되면, 생성중의 파일을 클로즈함과 함께 신규로 파일을 작성하고, 계속적으로 기록되는 비디오 데이터 및 오디오 데이터를 이 신규로 작성된 파일에 격납한다.
- [0258] 비디오 데이터 및 오디오 데이터가 격납되는 파일이 시스템상의 제약에 의거하여 자동적으로 분할되기 때문에, 유저는, 기록에 즈음하여 시스템상의 제약을 의식하는 일 없이 연속적인 기록을 행할 수 있다.
- [0259] 또한, 클로즈된 파일의 관리 정보에 대해, 신규로 작성된 파일을 나타내는 정보를 격납함과 함께, 클로즈된 파일에 격납된 비디오 데이터 및 오디오 데이터와, 신규로 작성된 파일에 격납된 비디오 데이터 및 오디오 데이터가 심レス하게 재생되는 것을 나타내는 식별 정보를 격납한다.
- [0260] 이것에 의하면, 재생시에, 연속적으로 기록된 비디오 데이터 및 오디오 데이터가 격납되는 파일이 시스템상의 제약에 의해 분할되어 있어도, 그것을 의식하는 일 없이 연속적인 재생이 가능해진다.
- [0261] 시스템상의 제약의 예로서는, 최대 파일 사이즈가 고려된다. 본 발명이 적용된 기록기에서 기록된 비디오 데이터 및 오디오 데이터가 격납된 파일을, 컴퓨터 장치 등의 외부 기기에서 취급하는 경우, 외부 기기가 이용하는 파일 시스템에서의 최대 파일 사이즈의 제한에 의거하여, 기록기에서 작성 가능한 파일 사이즈에 제한을 마련한다. 보다 많은 기기에서 대용 가능한 파일 시스템의 한 예로서, FAT16을 들 수 있다. FAT16에서는, 1파일의 최대 파일 사이즈가 2GB로 되어 있다. 예를 들면, 기록기의 시스템에서도, 이것에 대응하여 1파일의 최대 사이즈를 정하고, 기록되는 비디오 데이터 및 오디오 데이터가 격납되는 클립 AV 스트림 파일의 사이즈를 감시하고, 파일 사이즈가 2GB를 초과할 때에, 현재 기록중의 파일을 클로즈하여 신규로 파일을 작성하고, 신규로 작성된 파일에 대해 비디오 데이터 및 오디오 데이터를 기록한다.
- [0262] 시스템상의 제약의 다른 예로서, 어드레스 정보와 시각 정보와의 대응 관계를 나타내는 정보의 최대치가 고려된다. 즉, AVCHD에서는, 도 19 내지 도 23을 이용하여 설명한 EP 엔트리에 관해, 하나의 클립 인포메이션 파일에 격납 가능한 EP 엔트리 수에 상한이 마련되어 있다. 그래서, 비디오 데이터 및 오디오 데이터의 기록중에 이 EP 엔트리 수를 감시하고, EP 엔트리 수가 하나의 클립 인포메이션 파일에 격납 가능한 최대치를 초과할 때에, 현재 기록중의 파일을 클로즈하여 신규로 파일을 작성한다.
- [0263] 도 32는, 본 발명의 실시의 한 형태에 적용 가능한 기록 장치의 한 예의 구성을 개략적으로 도시한다. 이 기록 장치는, 입력된 디지털 비디오 데이터 및 디지털 오디오 데이터를, 소정의 방식에서 압축 부호화 및 다중화한 AV 스트림을 기록 매체에 기록하도록 하고 있다.
- [0264] 이 도 32에 예시되는 기록 장치는, 외부로부터 입력되는 비디오 데이터 및 오디오 데이터를 기록 매체에 기록하는, 단독의 기록 장치로서 이용할 수도 있고, 광학계나 촬상 소자 등을 구비한 카메라 블록과 조합시켜서, 촬상한 촬상 신호에 의거한 비디오 데이터를 기록 매체에 기록하는, 비디오 카메라 장치의 기록 블록으로서 이용할

수도 있다.

[0265] 적용 가능한 압축 부호화나 다중화의 방식으로서는, 다양하게 고려된다. 예를 들면, H. 264|AVC에 규정되는 방식을, 본 발명의 실시의 한 형태의 압축 부호화로서 적용할 수 있다. 이것으로 한정하지 않고, MPEG2 방식에 의거하여 압축 부호화를 행하도록 하여도 좋다. 또한, 다중화 방식은, 예를 들면 MPEG2 시스템을 적용할 수 있다. 이하에서는, 비디오 데이터의 압축 부호화를 H. 264|AVC에 규정된 방식에 준하여 행하고, 비디오 데이터 및 오디오 데이터의 다중화를, MPEG2 시스템에 규정되는 방식에 준하여 행하는 것으로 하여 설명한다.

[0266] 제어부(30)는, 예를 들면 CPU(Central Processing Unit), RAM(Random Access Memory) 및 ROM(Read Only Memory) 등으로 이루어지고(도시 생략), ROM에 미리 기억된 프로그램이나 데이터에 의거하여, RAM을 워크 메모리로서 이용하여 이 기록 장치의 기록부(10)의 각 부분을 제어한다. 또한, 제어부(30)와 기록부(10)의 각 부분을 접속하는 경로는, 번잡을 피하기 위해, 도 32에서는 생략하고 있다.

[0267] 제어부(30)상에서 동작하는 프로그램에 의해, 이 기록 장치에서 이용되는 파일 시스템이 제공된다. 예를 들면, 제어부(30)는, 이 파일 시스템에 의거하여, 데이터가 기록 매체(20)에 기록될 때의, 기록 매체(20)의 물리적인 어드레스와 해당 데이터가 격납되는 파일과의 관련시킴을 행함과 함께, 각 데이터가 격납되는 파일의 논리적인 관리 정보를 생성한다. 상술한 도 5에 도시하는 디렉토리 구조는, 파일의 논리적인 관리 정보의 한 예이다. 신규 파일의 작성이나 파일 오픈, 클로즈는, 파일 시스템에 의거하여 제어부(30)에 의해 제어된다.

[0268] UI(User Interface)부(31)는, 이 기록 장치의 동작을 유저가 조작하기 위한 조작자가 정해진 대로 마련되고, 조작자에 대한 조작에 따른 제어 신호를 출력한다. 이 제어 신호는, 제어부(30)에 공급된다. 제어부(30)는, 유저 조작에 따라 UI부(31)로부터 공급된 제어 신호에 의거하여 이루어지는 프로그램의 처리에 의해, 기록부(10)의 각 부분의 동작을 제어한다. 예를 들면, UI부(31)에 대해 이루어진 조작에 따라, 기록 장치에 의한 기록 동작의 시작 및 정지의 동작이 제어부(30)에 의해 제어된다.

[0269] 베이스밴드의 디지털 비디오 데이터가 단자(40)로부터 입력된다. 또한, 해당 디지털 비디오 데이터에 수반하여, 베이스밴드의 디지털 오디오 데이터가 단자(41)로부터 입력된다.

[0270] 디지털 비디오 데이터는 단자(40)로부터 기록부(10)에 입력되고, 비디오 인코더(11)에 공급된다. 비디오 인코더(11)는, 공급된 디지털 비디오 데이터를, 소정의 방식으로써 압축 부호화한다. H. 264|AVC에 규정되는 방식에 준하여 압축 부호화가 이루어지는 이 예에서는, 예를 들면, DCT(Discrete Cosine Transform)와 화면 내 예측에 의해 프레임 내 압축을 행함과 함께, 움직임 벡터를 이용한 프레임 사이 압축을 행하고, 또한 엔트로피 부호화를 행하여 압축 효율을 높인다. 비디오 인코더(11)에서 압축 부호화된 디지털 비디오 데이터는, H. 264|AVC의 엘리멘터리 스트림(ES)으로서, 멀티플렉서(MUX)13에 공급된다.

[0271] 디지털 오디오 데이터는 단자(41)로부터 기록부(10)에 입력되고, 오디오 인코더(12)에 공급된다. 오디오 인코더(12)는, 공급된 디지털 오디오 데이터를 소정의 압축 부호화 방식, 예를 들면 AC3(Audio Code number 3) 방식에 의해 압축 부호화한다. 오디오 데이터의 압축 부호화 방식은, AC3 방식으로 한정되는 것이 아니다. 오디오 데이터를 압축 부호화하지 않고, 베이스밴드의 데이터인 채로 이용하는 것도 고려된다. 압축 부호화된 디지털 오디오 데이터는, 멀티플렉서(13)에 공급된다.

[0272] 멀티플렉서(13)는, 각각 압축 부호화되어 공급된 디지털 비디오 데이터 및 디지털 오디오 데이터를 소정의 방식으로 다중화하고, 1개의 데이터 스트림으로서 출력한다. MPEG2 시스템에 준하여 다중화가 행하여지는 이 예에서는, MPEG2의 트랜스포트 스트림을 이용하여, 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.

[0273] 또한, H. 264|AVC 방식에서는, 비디오 데이터의 부호화를, 프레임 사이 상관을 이용한 프레임 사이 압축을 이용하여 행하고, 디코드 시작 가능한 위치에 의거한 디코드 단위로 부호화가 이루어진다. 이 디코드 단위는, 예를 들면 GOP이다. 클립 AV 스트림 파일에 격납되는 비디오 데이터도, 이 디코드 단위의 경계(境界)에 맞출 필요가 있고, 멀티플렉서(13)는, 이 디코드 단위의 경계에 맞추어서 TS 패킷을 작성한다.

[0274] 버퍼 메모리에 저장된 압축 비디오 데이터는, 소정 사이즈마다 분할되어 헤더가 부가(附加)되어, PES(Packetized Elementary Stream) 패킷화된다. 압축 오디오 데이터도 마찬가지로, 소정 사이즈마다 분할되어 헤더가 부가되어 PES 패킷화된다. 헤더에는, 패킷에 격납되는 데이터의 재생시각을 나타내는 PTS나 복호시각을 나타내는 DTS(Decoding Time Stamp)라는, MPEG2 시스템에 규정되는 소정의 정보가 격납된다. PES 패킷은, 더욱 분할되어 트랜스포트 패킷(TS 패킷)의 페이로드에 가득 채워진다. TS 패킷의 헤더에는, 페이로드에 가득 채

워진 데이터를 식별하기 위한 PID(Packet Identification)가 격납된다.

[0275] 멀티플렉서(13)는, TS 패킷에 대해, 다시 소정의 사이즈의 헤더를 부가하여 출력한다. 이, TS 패킷에 대해 소정의 헤더를 부가한 패킷을, 소스 패킷이라고 부른다. 소스 패킷의 각각에는, 소스 패킷의 각각을 식별함과 함께, 소스 패킷의 순번을 나타내는 소스 패킷 번호가 부여된다. 소스 패킷 번호는, 예를 들면 TS 패킷에 대해 부가되는 헤더에 격납할 수 있다. 멀티플렉서(13)로부터 출력된 소스 패킷은, 스트림 버퍼(14)에 일단 저장된다.

[0276] 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. 이것으로 한정하지 않고, 기록 매체(20)로서 하드 디스크 드라이브를 이용하여도 좋고, 반도체 메모리를 기록 매체(20)에 적용하는 것도 가능하다. 또한, 기록 매체(20)로서, 보다 대용량을 실현한 Blu-ray Disc(블루레이 디스크 : 등록상표)를 적용하는 것도 고려된다.

[0277] 기록 제어부(15)는, 스트림 버퍼(14)에 저장된 데이터량을 감시하고, 스트림 버퍼(14)에 소정량 이상의 데이터가 저장되면, 스트림 버퍼(14)로부터 기록 매체(20)의 기록 단위분의 데이터를 판독하여 기록 매체(20)에 기록한다. 데이터는, 파일 시스템에 의거한 제어부(30)의 제어에 의해, 소정의 파일에 대해 소스 패킷 단위로 차례로 추가 기록(追記)되도록 하여 기록 매체(20)에 기록된다.

[0278] 관리 정보 처리부(16)는, 예를 들면 CPU, 워크 메모리로서의 RAM 및 프로그램 소정의 데이터가 미리 기억되는 ROM으로 이루어진다(도시 생략). 이것으로 한정하지 않고, 관리 정보 처리부(16)는, 예를 들면 제어부(30)에서의 프로그램 처리로 관리 정보 처리부(16)의 기능을 실현하는 것도 가능하다. 이 경우, 예를 들면 제어부(30)가 갖는 RAM이 휘발성 메모리(17)로서 이용됨과 함께, 불휘발성 메모리(18)가 제어부(30)에 접속된다.

[0279] 관리 정보 처리부(16)는, 기록 데이터에 의거하여, 휘발성 메모리(17)를 워크 메모리로서 이용하여, 상술한 인덱스 파일 "index.bdmv", 무비 오브젝트 파일 "MovieObject.bdmv", 플레이 리스트 파일 "xxxxx.mpls" 및 클립 인포메이션 파일 "zzzzz.clpi"에 격납하기 위한 정보를 생성한다. 생성된 정보는, 소정의 타이밍에서 기록 매체(20)에 기록된다.

[0280] 한 예로서, 관리 정보 처리부(16)는, 멀티플렉서(13)로부터 기록 데이터의 시간 정보를 취득함과 함께, 기록 제어부(15)로부터 기록 데이터의 기록 매체(20)에 대한 어드레스 정보를 취득하고, 취득된 이들 시간 정보 및 어드레스 정보에 의거하여 EP 엔트리 정보가 생성된다. 또한, UI부(31)에 대한 기록 시작, 기록 종료의 조작에 따라 제어부(30)로부터 출력되는 제어 신호와, 멀티플렉서(13) 및 기록 제어부(15)로부터의 기록 데이터에 관한 정보에 의거하여, 플레이 리스트 파일 "xxxxx.mpls"의 생성 또는 갱신, 클립 인포메이션 파일 "zzzzz.clpi"의 생성 등이 행하여진다. 또한, 기록 매체(20)에 대해 신규로 기록이 행하여질 때에는, 인덱스 파일 "index.bdmv"이나 무비 오브젝트 파일 "MovieObject.bdmv"의 생성 또는 갱신이 행하여진다.

[0281] 다음에, 본 발명의 실시의 한 형태에 의한 클립의 기록 제어에 관해 설명한다. 도 33은, 본 발명의 실시의 한 형태에 의한 한 예의 기록 방법을 도시하는 플로우 차트이다. 또한, 이 실시의 한 형태에서는, 신규로 기록되는 클립의 정보는, 이미 존재하는 플레이 리스트에 대해 차례로, 추가 기록되도록 되어 있다. 또한, 이 플로우 차트에 의한 처리에 앞서서, 기록 매체(20)에는, 플레이 리스트 파일, 인덱스 파일 및 무비 오브젝트 파일이 이미 기록되어 있는 것으로 한다.

[0282] 스텝 S50에서 기록 시작 조작이 행하여지면, 다음의 스텝 S51에서, 제어부(30)의 제어에 의해, 신규로 클립 AV 스트림 파일이 작성된다. 예를 들면, 파일 시스템에 의해, 신규로 작성하는 클립 AV 스트림 파일의 논리 구조가 설정됨과 함께, 기록 매체(20)상의 물리 어드레스가 관련시켜진다. 다음의 스텝 S52에서, 스텝 S51에서 작성된 클립 AV 스트림 파일에 대한 클립 AV 스트림의 기록이 시작된다.

[0283] 기록 시작 조작은, 예를 들면 다음과 같이 이루어진다. 예를 들면, UI부(31)에, 기록 시작을 지시하는 기록 시작 스위치와, 기록 정지를 지시하는 기록 정지 스위치가 마련된다. 스텝 S50에서는, 유저에 의해 기록 시작 스위치가 조작된다. 이 조작에 따라, 기록 시작을 지시하는 제어 신호가 UI부(31)로부터 출력되고, 제어부(30)에 공급된다. 제어부(30)는, 이 기록 시작을 지시하는 제어 신호에 의거하여, 기록부(10)의 각 부분에 대해, 단자(40)로부터 입력되는 베이스밴드의 비디오 데이터와, 단자(41)로부터 입력되는 베이스밴드의 오디오 데이터를 클립 AV 스트림 파일에 기록하도록 제어한다.

[0284] 기록 시작에 관한 제어부(30)의 제어의 한 예로서, 기록 정지 상태에서는 비디오 인코더(11) 및 오디오 인코더(12)의 동작을 정지시켜 두고, 기록 시작의 지시에 따라, 이들 비디오 인코더(11) 및 오디오 인코더(12)의 동작을 시작시키는 것이 고려된다. 이것으로 한정하지 않고, 멀티플렉서(13)나 스트림 버퍼(14), 기록 제어부(15)의

동작의 시작, 정지를 제어함으로써, 기록 시작 및 정지의 제어를 행할 수도 있다.

[0285] 기록 시작의 제어에 따라, 클립 AV 스트림이 기록 매체(20)상의 클립 AV 스트림 파일에 기록된다(스텝 S52). 즉, 입력된 비디오 데이터 및 오디오 데이터가 비디오 인코더(11) 및 오디오 인코더(12)에서 각각 압축 부호화되고, 멀티플렉서(13)에서 패킷화되어 TS 패킷에 소정의 헤더가 부가된 소스 패킷이 되어 스트림 버퍼(14)에 공급된다. 스트림 버퍼(14)에 소정량 이상의 소스 패킷이 저장되면, 기록 제어부(15)에 의해 스트림 버퍼(14)로부터 소스 패킷이 판독된다. 판독된 소스 패킷은, 정해진대로 파일명이 부착된 클립 AV 스트림 파일에 관련시켜진 기록 매체(20)상의 어드레스에 의거하여, 순서로, 기록 매체(20)에 기록된다.

[0286] 신규로 작성되는 클립 AV 스트림 파일의 파일명은, 기록 매체(20)에 이미 기록되어 있는 다른 클립 AV 스트림 파일과 중복되지 않는 파일명이 선택된다.

[0287] 또한, 클립 AV 스트림의 기록 매체(20)에의 기록에 수반하여, 관리 정보 처리부(16)에 의해, 기록되는 데이터의 재생시간과 어드레스(클립 AV 스트림 파일 내의 논리 어드레스)와의 대응 관계를 나타내는 EP 엔트리 리얼타임으로 생성된다. 이 데이터는, 상술한 클립 인포메이션 파일 "zzzzz.clpi" 내의 블록(blkEPMMap())에 격납되는 데 이터로서, 휘발성 메모리(17)에 기억된다. 해당 데이터의 백업으로서, 불휘발성 메모리(18)에도 동일한 데이터를 기억할 수도 있다.

[0288] 다음의 스텝 S53에서, 현재 기록중의 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈, 예를 들면 2GB를 초과하는지의 여부가 판단된다. 한 예로서, 제어부(30)는, 현재 기록중의 클립 AV 스트림 파일의 파일 사이즈를 항상 감시하고, 예를 들면 스트림 버퍼(14)로부터 다음의 소스 패킷이 판독되어 해당 클립 AV 스트림 파일에 기록된 때에, 해당 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과하는지의 여부를 판단한다.

[0289] 보다 구체적인 예로서는, 제어부(30)는, 멀티플렉서(13)에서의 처리를 감시하고, 멀티플렉서(13)로부터 출력되는 소스 패킷의 용량을, 클립 AV 스트림 파일의 선두에 격납되는 소스 패킷부터 적산한다. 이 적산치에 의거하여, 다음에 멀티플렉서(13)로부터 출력되는 소스 패킷으로 데이터량이 소정치를 초과하는지의 여부를 판단한다.

[0290] 또한, 상술한 바와 같이, 클립 AV 스트림 파일은, 디코드 시작 가능한 위치에 의거한 디코드 단위의 경계에 맞추어서 기록된다. 따라서 스텝 S53에서의 판단은, 실제로는, 디코드 단위로 행하여진다. 예를 들면, 제어부(30)는, 스트림 버퍼(14)로부터 다음의 디코드 단위분의 소스 패킷이 판독되어 클립 AV 스트림 파일에 기록된 때에, 해당 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과하는지의 여부를 판단한다.

[0291] 스텝 S53에서, 다음의 소스 패킷이 클립 AV 스트림 파일에 기록된 때에, 해당 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과한다고 판단되면, 처리는 후술하는 스텝 S62로 이행된다. 이때, 제어부(30)는, 예를 들면, 클립 AV 스트림 파일에 대해 기록됨에 의해 파일 사이즈가 소정 사이즈를 초과한다고 된 해당 다음의 소스 패킷의 소스 패킷 번호를 보존한다.

[0292] 한편, 스텝 S53에서, 다음의 소스 패킷이 클립 AV 스트림 파일에 기록된 때에, 해당 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과하지 않는다고 판단되면, 처리는 스텝 S54로 이행되고, 기록 정지 조작이 행하여졌는지의 여부가 판단된다. 예를 들면, 유저에 의해 UI부(31)에 마련된 기록 정지 스위치가 조작되고, 기록이 정지되었다고 판단되면, 처리는 스텝 S55로 이행된다. 한편, 기록이 정지되지 않았다고 판단되면, 처리는 스텝 S52로 되돌아와, 클립 AV 스트림의 기록 매체(20)에의 기록이 계속된다. 또한, 이 스텝 S53 및 스텝 S54의 판단에 의거한 루프 처리는, 예를 들면 멀티플렉서(13)에서 소스 패킷이 출력될 때마다 행하여진다.

[0293] 스텝 S55에서는, 기록의 정지에 수반하여, 스트림 버퍼(14)에 저장되어 있는 스트림이 전부 기록 매체(20)에 기록된다. 예를 들면, 기록 제어부(15)는, 제어부(30)로부터의 기록 정지의 명령에 따라, 스트림 버퍼(14)에 저장되어 있는 스트림(소스 패킷)을 전부 판독하고, 기록 매체(20)상의 현재 기록중의 클립 AV 스트림 파일에 기록한다. 그리고, 다음의 스텝 S56에서, 파일 시스템에서 해당 클립 AV 스트림 파일이 클로즈된다.

[0294] 또한, 기록 정지의 명령에 따라, 예를 들면 비디오 인코더(11) 및 오디오 인코더(12)의 동작이 정지된다. 이때, 도 13A를 이용하여 설명한 제 1의 심레스 접속을 행하기 위해, 예를 들면, 오디오 인코더(12)의 동작이나 비디오 인코더(11)의 동작이 정지하고 나서 소정 시간 후에 정지되도록 제어된다.

[0295] 다음의 스텝 S57 내지 스텝 S61에서, 기록 매체(20)에 기록된 클립 AV 스트림 파일에 관한 클립 인포메이션 파일이 생성됨과 함께, 플레이 리스트 파일의 생성이 이루어진다.

[0296] 우선, 스텝 S57에서, 관리 정보 처리부(16)에 의해, 스텝 S56에서 클로즈된 클립 AV 스트림 파일에 대응하는 클립 인포메이션 파일 "zzzzz.clpi"가 생성된다. 파일명은, 스텝 S51에서 작성된 클립 AV 스트림 파일의 파일명과

대응하는 파일명이 되고, 해당 클립 AV 스트림 파일의 파일명이 "zzzzz.m2ts"이면, 이 클립 인포메이션 파일의 파일명은, 확장자보다 앞의 부분이 동일한 파일명"zzzzz.clpi"이 된다.

[0297] 클립 인포메이션 파일"zzzzz.clpi"에, 도 15 내지 도 21에 예시한 각 선택스에 따라, 각 필드나 플래그의 값이 정해진대로 설정되어 격납된다. 한 예로서, 소스 패킷에 관한 정보나, 재생시간(PTS)에 관한 정보는, 관리 정보 처리부(16)에 의해, 클립의 기록중에 멀티플렉서(13)로부터 취득된 정보에 의거하여 생성된다. 또한, 기록 매체(20)상의 기록 어드레스에 관한 정보는, 관리 정보 처리부(16)에 의해, 클립의 기록중에 기록 제어부(15)로부터 취득된 정보에 의거하여 생성된다. 시스템에 의해 고유의 값은, 예를 들면 미리 ROM(도시 생략) 등에 기억되어 있는 정보에 의거한다. 또한, 재생시간과 어드레스와의 대응 관계를 나타내는 상술한 블록(blkEPMap())의 정보가, 클립 인포메이션 파일"zzzzz.clpi"의 블록(blkCPI())에 격납된다.

[0298] 또한, 블록(blkClipInfo()) 내의 플래그(IsCC5)는, 유저 조작에 의해 클립의 기록이 정지된 경우, 값이 "1"(바이너리 값)이 된다. 그것에 수반하여, 블록(blkClipInfo()) 내의 if문자(도 16 참조)에서 나타나는 데이터가 정해진대로 설정된다.

[0299] 클립 인포메이션 파일의 작성이 완료되면, 처리는 다음의 스텝 S58로 이행한다. 스텝 S58 내지 스텝 S61의 처리는, 플레이 리스트 파일에 관한 처리이다. 이 스텝 S58 내지 스텝 S61의 처리에 의해, 이미 기록 매체(20)상에 존재하는 플레이 리스트 파일에 대해, 새롭게 기록된 클립 AV 스트림 파일"00002.m2ts"에 대응하는 플레이 아이템이 추가된다.

[0300] 우선, 스텝 S58에서, 플레이 리스트 파일 내의 블록(blkPlayItem())에서의 필드(ConnectionCondition)의 값이 "5"로 설정되고, 이 클립 AV 스트림 파일"00002.m2ts"이 다음에 기록되는 클립 AV 스트림 파일에 대해 제 1의 심레스 접속에 의해 접속되는 것이 나타난다(도 12 참조). 다음에 스텝 S59에서, 플레이 아이템 파일의 필드(NumberOfPlayItems)의 값이 "1"만큼 잉크리먼트되고, 해당 플레이 리스트에 대해 플레이 아이템이 하나, 추가되는 것이 나타난다(도 11 참조).

[0301] 다음의 스텝 S60에서, 블록(blkPlayItem())에서의 필드(ClipInformationFileName), 필드(INTime) 및 필드(OUTTime)가 각각 설정되고, 클립의 기록에 수반하여 추가되는 블록(blkPlayItem())이 작성된다. 필드(ClipInformationFileName)는, 상술한 스텝 S55에서 작성된 클립 인포메이션 파일의 파일명"zzzzz.clpi"이 격납된다. 실제로는, 클립 인포메이션 파일의 확장자는 고정적으로 되어 있기 때문에, 퍼리어드 앞의 부분"zzzz z"이 격납된다. 필드(INTime) 및 필드(OUTTime)는, 대응하는 클립 AV 스트림 파일"00002.m2ts"에 격납되는 비디오 스트림의 선두 및 종단의 시간을 나타내는 정보로서, 예를 들면 클립 인포메이션 파일"zzzzz.clpi" 내의 블록(blkCPI())에서의 블록(blkEPMap())의 정보에 의거한다.

[0302] 다음의 스텝 S61에서, 플레이 리스트 파일 내의 블록(blkPlayListMark())에서의 필드(NumberOfPlayListMarks)의 값이 "1"만큼 잉크리먼트되고, 그것에 수반하여 for루프문자 내에 추가된 필드(MarkTimeStamp)의 값이, 상술한 스텝 S60에서 설정된, 블록(blkPlayItem())에서의 필드(INTime)의 값으로 설정된다. 즉, 새롭게 기록된 클립 AV 스트림의 선두에 대응하는 시각에, 플레이 리스트 마크가 찍혀진다.

[0303] 이와 같이 하여, 새롭게 기록된 클립 AV 스트림 파일"zzzzz.m2ts"에 대해, 클립 인포메이션 파일"zzzzz.clpi"이 작성됨과 함께, 기존의 플레이 리스트 파일이 갱신된다.

[0304] 기록이 정지된 상태로부터, UI부(31)에 대한 기록 시작 조작을 행함으로써, 재차 스텝 S50부터의 처리가 시작되고, 새로운 클립 AV 스트림 파일의 기록 매체(20)에의 기록과, 클립 AV 스트림 파일의 파일 사이즈에 의거한 판단, 대응하는 클립 인포메이션 파일의 작성, 작성된 클립 인포메이션 파일을 참조하는 플레이 아이템의 생성, 및, 생성된 플레이 아이템의 플레이 리스트에의 추가 처리가 마찬가지로 하여 행하여진다.

[0305] 또한, 상술한 스텝 S55에 의한 스트림 버퍼(14)에 저장된 데이터의 기록 매체(20)에의 기록 처리는, 스텝 S61의 처리의 후에 행하도록 하여도 좋다.

[0306] 상술한 스텝 S53의 판단에 의거하여, 본 발명의 실시의 한 형태에 의한, 시스템 사정에 의거하여 클립 AV 스트림 파일이 분할된 때의 처리에 관해 설명한다. 상술한 스텝 S53에서, 다음의 소스 패킷이 클립 AV 스트림 파일에 기록된 때에, 해당 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 넘는다고 판단된 경우, 처리는 스텝 S62로 이행된다. 스텝 S62에서는, 스트림 버퍼에 저장된 스트림이 클립 AV 스트림 파일"zzzzz.m2ts"에 기록된다.

[0307] 이때, 스텝 S53의 판단에 즈음하여 제어부(30)에 보존된 소스 패킷 번호에 의거하여, 클립 AV 스트림 파일

일"zzzzz.m2ts"에 대해 기록됨에 의해 파일 사이즈가 소정 사이즈를 초과한다고 된 해당 다음의 소스 패킷까지, 즉, 해당 다음의 소스 패킷보다 먼저 스트림 버퍼에 저장된 스트림이, 클립 AV 스트림 파일"zzzzz.m2ts"에 대해 기록된다.

- [0308] 다음의 스텝 S63에서, 파일 시스템에 의해, 현재 기록중의 클립 AV 스트림 파일이 클로즈된다. 파일이 클로즈되면, 스텝 S64에서, 관리 정보 처리부(16)에 의해, 스텝 S51에서 작성된 클립 AV 스트림 파일에 대응하는 클립 인포메이션 파일"zzzzz.clpi"이, 스텝 S57의 처리에서 설명한 것과 마찬가지로 하여, 정해진대로 생성된다.
- [0309] 다음의 스텝 S65에서, 플레이 리스트 파일 내의 블록(blkPlayItem())에서의 필드(ConnectionCondition)의 값이 "6"으로 설정되고, 이 클립 AV 스트림 파일"zzzzz.m2ts"이 다음에 기록되는 클립 AV 스트림 파일에 대해 제 2의 심레스 접속으로 접속된 것이 나타난다(도 12 참조).
- [0310] 스텝 S66에서, 플레이 리스트 파일의 필드(NumberOfPlayItems)의 값이 "1"만큼 잉크리먼트되고, 해당 플레이 리스트에 대해 플레이 아이템이 하나, 추가되는 것이 나타난다(도 11 참조). 그리고, 스텝 S67에서, 블록(blkPlayItem())에서의 필드(ClipInformationFileName)가 설정됨과 함께, 클립 인포메이션 파일 내의 블록(blkCPI())에서의 블록(blkEPMMap())의 정보에 의거하여 필드(INTime) 및 필드(OUTtime)가 각각 설정되고, 클립의 기록에 수반하여 추가되는 블록(blkPlayItem())이 작성된다.
- [0311] 다음의 스텝 S68에서, 플레이 리스트 파일 내의 블록(blkPlayListMark())에서의 필드(NumberOfPlayListMarks)의 값이 "1"만큼 잉크리먼트되고, 그것에 수반하여 for루프문자 내에 추가된 필드(MarkTimeStamp)의 값이, 상술한 스텝 S67에서 설정된, 블록(blkPlayItem())에서의 필드(INTime)의 값으로 설정된다. 즉, 새롭게 기록된 클립의 선두에, 플레이 리스트 마크가 찍혀진다.
- [0312] 스텝 S68의 처리가 종료되면, 처리는 스텝 S51로 되돌아와, 신규로 클립 AV 스트림 파일이 작성된다. 이 스텝 S53의 판단의 후의 스텝 S51에서 작성된 클립 AV 스트림 파일은, 해당 스텝 S53의 판단에 의해, 다음의 소스 패킷이 파일에 격납되면 해당 파일의 파일 사이즈가 소정 사이즈를 초과한다고 된, 해당 다음의 소스 패킷부터 순서로, 소스 패킷이 격납되는 파일이다.
- [0313] 여기서, 이때 작성되는 클립 AV 스트림 파일과, 하나 전에 작성된 클립 AV 스트림 파일은, 비디오 데이터 및 오디오 데이터가 각각 연속적으로 되도록 된다. 예를 들면 오디오 데이터에 관해서는, 하나 전에 작성된 클립 AV 스트림 파일에 기록된 오디오 데이터의 후단의 시작과, 여기서 작성된 클립 AV 스트림 파일에 기록되는 오디오 데이터의 선두의 시작이 대응하도록, 기록이 제어된다. 예를 들면, 제어부(30)에 의해 오디오 인코더(12), 멀티플렉서(13) 및 스트림 버퍼(14)를 정해진대로 제어함으로써, 이와 같은 기록 제어가 행하여진다.
- [0314] 또한, 상술에서는, 스텝 S68의 처리 후에 처리를 스텝 S5로에 되돌려서 신규 클립 AV 스트림 파일을 작성하도록 하고 있지만, 이것은 이 예로 한정되지 않는다. 즉, 스텝 S53의 판단에 의해, 시스템 사정에 의거하여 클립 AV 스트림 파일이 분할된 경우, 분할 전후의 클립 AV 스트림 파일이 제 2의 심레스 접속으로 접속되기 때문에, 스텝 S63에서 파일이 클로즈되고 나서 신규의 클립 AV 스트림 파일이 작성되기까지의 시간은, 가능한 한 짧은 쪽이 바람직하다.
- [0315] 그 때문에, 예를 들면, 스텝 S63에서 클립 AV 스트림 파일을 클로즈한 직후에 처리를 스텝 S51로 되돌려서, 신규로 클립 AV 스트림 파일을 작성하는 것이 고려된다. 이 경우, 스텝 S64 내지 스텝 S68의 처리는, 신규의 클립 AV 스트림 파일의 작성 처리 등과 병렬적으로 행하도록 한다. 플레이 리스트 파일이나, 스텝 S63에서 작성되는 클립 인포메이션 파일은, 휘발성 메모리(17) 및/또는 불휘발성 메모리(18)에 일시적으로 보존된다.
- [0316] 또한, 상술한 스텝 S61 및 스텝 S68에 의한 플레이 리스트 마크 설정의 처리는, 예를 들면, 스텝 S50의 기록 시작 조작에 수반하여 신규로 작성된 클립 AV 스트림 파일에 대한, 시스템 사정에 의거한 분할시에 있어서, 최초의 분할시 이외는, 생략할 수 있다. 한 예로서, 제 2의 심레스 접속으로 접속되는 클립 AV 스트림 파일의 선두에 대응하는 플레이 리스트 마크를 생략하는 것이 고려된다. 이것으로 한정하지 않고, 기록 정지 조작시에, 해당 기록 정지 조작에 대응하는 기록 시작 조작에 의해 기록이 시작된 시작에 대해, 플레이 리스트 마크를 설정하도록 하여도 좋다.
- [0317] 도 34는, 상술한 도 33의 순서에 따라 작성된 한 예의 파일 구조를 도시한다. 한 예로서, 도 35A에 도시되는 바와 같은 시간 경과에서, 기록이 이루어진 것으로 한다. 즉, 도 35A를 참조하여, 기록 매체(20)를 이용한 기록에 대한 기록 시작 조작이 시작(ST₁)에서 행하여지고, 대응하는 기록 정지 조작이 시작(ED₁)에서 행하여지고, 기록 구간(400)의 기록이 이루어진다. 같은 기록 매체(20)에 대해, 시작(ED₁)보다 후의 시작(ST₂)에서 재차 기록 시작

조작이 행하여지고, 시각(ED₂)에서 기록 정지 조작이 행하여지고, 기록 구간(401)의 기록이 이루어진다. 마찬가지로, 같은 기록 매체(20)에 대해, 시각(ED₂)보다 후의 시각(ST₃)에서 다시 재차 기록 시작 조작이 행하여지고, 시각(ED₃)에서 기록 정지 조작이 행하여지고, 기록 구간(402)의 기록이 이루어진다.

- [0318] 여기서, 클립 AV 스트림 파일에 대해 파일 사이즈의 상한으로서 마련된 소정 사이즈(예를 들면 2GB)에 상당하는 기록시간이, 예를 들면 도 35B에 도시되는 길이로 나타나는 것으로 한다. 이 경우, 기록 구간(400)은, 기록 시작시각(ST₁)부터 시점(時點)(a)까지에서, 소정 사이즈의 2GB에 대응하는 시간의 기록이 행하여지고, 또한 시점(a)부터 시점(b)까지에서, 소정 사이즈에 대응하는 시간의 기록이, 또한 행하여지고 있다. 시점(b)부터 기록 정지시각(ED₁)까지에 관해서는, 소정 사이즈에 대응하는 시간에 못미치는 기록시간으로 되어 있다.
- [0319] 한편, 기록 구간(401) 및 기록 구간(402)은, 각각, 파일 사이즈의 상한인 2GB에 대응하는 시간에 못미친다.
- [0320] 상술한 도 33에서 도시한 처리를, 이 도 35A에 도시되는 시간 경과로 기록이 행하여진 경우에 관해, 도 33의 플로우 차트 및 도 34의 파일 구조를 참조하면서 보다 구체적으로 설명한다.
- [0321] 우선, 시각(ST₁)에서의 기록 시작 조작에 수반하여, 스텝 S51에서 클립 AV 스트림 파일"00001.m2ts"이 작성된다. 작성된 클립 AV 스트림 파일"00001.m2ts"에 대해 스트림 기록되어 가고, 예를 들면 소스 패킷 단위로, 클립 AV 스트림 파일"00001.m2ts"의 파일 사이즈가 제어부(30)에 체크되고, 파일 사이즈가 소정 사이즈를 초과하는지의 여부가 판단된다. 실제로는, 상술한 바와 같이, 멀티플렉서(13)로부터 출력되는 소스 패킷의 데이터량이 적산된 결과에 의거하여, 판단이 이루어진다. 도 35A의 예에서는, 시점(a)의 직전에, 다음의 소스 패킷이 기록되면 클립 AV 스트림 파일"00001.m2ts"의 파일 사이즈가 소정 사이즈를 초과한다, 라고 판단되는 것으로 된다.
- [0322] 처리는 스텝 S62로 이행되고, 스트림 버퍼에 저장된 스트림이 정해진대로 판독되고, 클립 AV 스트림 파일"00001.m2ts"에 대해 기록된다. 그리고, 스텝 S63에서 클립 AV 스트림 파일"00001.m2ts"이 클로즈되고, 대응하는 클립 인포메이션 파일"00001.clpi"이 작성된다(스텝 S64). 또한, 상술한 바와 같이, 클립 AV 스트림 파일"00001.m2ts"이 클로즈되면, 스텝 S64부터의 처리와 병렬적으로, 처리를 스텝 S51로 되돌려서 다음의 클립 AV 스트림 파일의 작성을 행하여도 좋다.
- [0323] 다음의 스텝 S65부터, 클립 AV 스트림 파일"00001.m2ts"을 참조하는 플레이 아이템#1의 작성 및 플레이 리스트 #1의 생성이 이루어진다. 우선, 플레이 아이템#1 내의 필드(ConnectCondition)의 값이, 제 2의 심레스 접속이 나타내는 값"6"으로 설정되고(스텝 S65), 클립 인포메이션 파일"00001.clpi"에 의거하여, 예를 들면 시각(ST₁) 및 시각(a)을 각각 필드(INTTime 및 OUTTime)로 한 플레이 아이템#1이 정해진대로 작성된다(스텝 S66). 그리고, 이 플레이 아이템#1이 플레이 리스트#1에 대해 기술된다. 또한, 필드(MarkTimeStamp)의 값을 필드(INTTime)의 값 즉 시각(ST₁)이 되어, 플레이 리스트 마크(Mark#1)가 설정된다.
- [0324] 처리는 스텝 S51로 되돌아와, 시각(a)부터의 스트림을 기록하기 위해, 신규로 클립 AV 스트림 파일"00002.m2ts"이 작성된다. 이 클립 AV 스트림 파일"00002.m2ts"에 스트림이 기록된다(스텝 S52).
- [0325] 여기서, 상술한 바와 같이, 하나 전에 작성되어 기록된 클립 AV 스트림 파일"00001.m2ts"에 대응하는 플레이 아이템#1에서, 필드(ConnectionCondition)의 값이 "6"으로 설정되어, 제 2의 심레스 접속이 지정되어 있다. 그래서, 하나 전에 기록된 클립 AV 스트림 파일"00001.m2ts"에 기록된 오디오 데이터와, 여기서 작성되는 클립 AV 스트림 파일"00002.m2ts"에 기록된 오디오 데이터가 연속적으로 재생되도록, 기록 제어가 이루어진다. 예를 들면, 클립 AV 스트림 파일"00001.m2ts"에 기록된 오디오 데이터의 후단의 시각과, 클립 AV 스트림 파일"00002.m2ts"의 선두의 시각이 대응하도록, 기록 제어가 이루어진다.
- [0326] 다음에, 스텝 S53에서 클립 AV 스트림 파일"00002.m2ts"의 파일 사이즈가 소정 사이즈를 초과하는지의 여부가 판단된다. 도 35A의 예에서는, 시점(b)의 직전에, 다음의 소스 패킷이 기록되면 클립 AV 스트림 파일"00002.m2ts"의 파일 사이즈가 소정 사이즈를 초과한다, 라고 판단되는 것이 된다. 그러면, 처리는 스텝 S62로 이행하고, 상술한 바와 마찬가지로 하여, 스트림 버퍼에 저장된 스트림이 클립 AV 스트림 파일"00002.m2ts"에 대해 기록되고(스텝 S62) 해당 클립 AV 스트림 파일"00002.m2ts"이 클로즈된다(스텝 S63). 그리고, 해당 클립 AV 스트림 파일"00002.m2ts"에 대응하는 클립 인포메이션 파일"00002.clpi"이 작성된다(스텝 S64).
- [0327] 다음에, 클립 인포메이션 파일"00002.clpi"에 대응하는 플레이 아이템#2의 필드(ConnectCondition)가, 제 2

의 심레스 접속을 나타내는 값"6"으로 설정되고(스텝 S65), 다음의 스텝 S66에서, 시각(a) 및 시각(b)을 각각 필드(INTime) 및 필드(OUTTime)로 한 플레이 아이템#2이 정해진대로 작성되고, 플레이 리스트#1에 대해 추가가 롤된다. 그리고, 스텝 S68에서, 필드(INTime)의 값 즉 시각(a)을 나타내는 플레이 리스트 마크가 설정된다. 또한, 여기서의 플레이 리스트 마크 설정의 처리는, 생략 가능하다.

[0328] 처리는 재차 스텝 S51로 되돌아와, 시각(b)부터의 스트림을 기록하기 위해, 신규로 클립 AV 스트림 파일"00003.m2ts"이 작성된다. 이 클립 AV 스트림 파일"00003.m2ts"에 스트림이 기록된다(스텝 S52).

[0329] 여기서, 상술한 바와 같이, 하나 전에 작성되고 기록된 클립 AV 스트림 파일"00001.m2ts"에 대응하는 플레이 아이템#1에서, 필드(ConnectionCondition)의 값이 "6"으로 설정되고, 제 2의 심레스 접속이 지정되어 있다. 그래서, 상술한 바와 마찬가지로, 예를 들면, 클립 AV 스트림 파일"00002.m2ts"에 기록된 오디오 데이터의 후단의 시각과, 클립 AV 스트림 파일"00003.m2ts"의 선두의 시각이 대응하도록, 기록 제어가 이루어진다.

[0330] 다음에, 스텝 S53에서, 클립 AV 스트림 파일"00003.m2ts"의 파일 사이즈가 소정 사이즈를 초과하는지의 여부가 판단된다. 도 35A의 예에서는, 클립 AV 스트림 파일"00003.m2ts"의 파일 사이즈가, 파일 사이즈의 상한인 2GB를 초과하기 전에, 시각(ED₁)에서 기록 정지 조작이 이루어지고 있다. 따라서 스텝 S54의 판단에 의거하여, 처리가 스텝 S55로 이행된다. 스텝 S55에서, 스트림 버퍼에 저장된 스트림이 클립 AV 스트림 파일"00003.m2ts"에 기록되고, 스텝 S56에서 해당 AV 스트림 파일"00003.m2ts"이 클로즈되고, 스텝 S57에서 대응하는 클립 인포메이션 파일"00003.clpi"이 작성된다.

[0331] 다음에, 해당 클립 인포메이션 파일"00003.clpi"을 참조하는 플레이 아이템#3의 필드(ConnectionCondition)가, 제 1의 심레스 접속을 나타내는 값"5"으로 설정된다(스텝 S58). 그리고, 플레이 아이템#3이 정해진대로 작성되고, 해당 플레이 아이템#3이 플레이 리스트 파일#1에 추가 기록된다(스텝 S59, 스텝 S60). 또한, 플레이 리스트 파일#1에 대해, 플레이 아이템#3의 필드(INTime)에 대응하는 시각에 플레이 리스트 마크를 찍을 수 있다(스텝 S61). 또한, 여기서의 플레이 리스트 마크 설정의 처리는, 생략 가능하다.

[0332] 시각(ED₁)에서 기록 정지 조작이 이루어진 후에, 시각(ST₂)에서 재차 기록 시작 조작이 이루어지면(스텝 S50), 스텝 S51에서 신규로 클립 AV 스트림 파일"00004.m2ts"이 작성되고, 스트림이 해당 클립 AV 스트림 파일"00004.m2ts"에 기록된다(스텝 S52). 도 35A의 예에서는, 클립 AV 스트림 파일"00004.m2ts"의 파일 사이즈가 소정 사이즈를 초과하기 전에, ED₂에서 기록 정지 조작이 이루어지고 있다. 따라서 스텝 S54의 판단에 의거하여, 처리가 스텝 S55로 이행된다.

[0333] 스텝 S55에서, 스트림 버퍼에 저장된 스트림이 클립 AV 스트림 파일"00004.m2ts"에 기록되고, 스텝 S56에서 해당 AV 스트림 파일"00004.m2ts"이 클로즈되고, 스텝 S57에서 대응하는 클립 인포메이션 파일"00004.clpi"이 작성된다. 스텝 S58에서, 해당 클립 인포메이션 파일"00004.clpi"을 참조하는 플레이 아이템#4의 필드(ConnectionCondition)의 값이, 제 1의 심레스 접속을 나타내는 값"5"으로 설정된다. 그리고, 플레이 아이템#4이 정해진대로 작성되고, 해당 플레이 아이템#4이 플레이 리스트#1에 추가 기록된다(스텝 S59, 스텝 S60). 또한, 플레이 리스트#1에 대해, 플레이 아이템#4의 필드(INTime)에 대응하는 시각에서 플레이 리스트 마크(Mark#2)가 설정된다(스텝 S61).

[0334] 또한, 도 35A에서의 다음의 기록 시작시각(ST₃)부터 기록 정지시각(ED₃)에 의한 처리는, 상술한 기록 시작시각(ST₂)부터 기록 정지시각(ED₂)에 의한 처리와 동등하기 때문에, 번잡을 피하기 위해 생략한다.

[0335] 또한, 최초의 클립 AV 스트림 파일"00001.m2ts"의 기록에 수반하여, 플레이 리스트#1을 호출하는 무비 오브젝트 #1가 기술되는 무비 오브젝트 파일과, 무비 오브젝트#1를 타이틀#1로서 선택 및 재생시키기 위한 인덱스 파일을 작성하도록 할 수 있다. 무비 오브젝트 파일 및 인덱스 파일은, 템플릿을 미리 작성하여 두고, 그것을 갱신하도록 할 수 있다. 이를 무비 오브젝트 파일 및 인덱스 파일의 작성에 관해서는, 본 발명의 주지와 관련성이 희박하기 때문에, 상세한 설명을 생략한다.

[0336] 도 34 및 도 35A를 이용하여 설명한 바와 같이 하여 클립 AV 스트림 파일이 기록된 기록 매체(20)의 한 예의 재생에 관해, 개략적으로 설명한다. 또한, 여기서는, 기록 매체(20)는, 기록 가능한 타입의 DVD인 것으로 한다. 기록 매체(20)가 재생 장치에 장전되면, 인덱스 파일이 판독되고, 또한 무비 오브젝트 파일이 판독된다. 인덱스 파일에 대해 타이틀#1의 재생이 지시되면, 무비 오브젝트 파일 내의 대응하는 무비 오브젝트#1가 참조되고, 플레이 리스트#1가 기술되는 플레이 리스트 파일이 기록 매체(20)로부터 판독된다.

- [0337] 플레이 리스트#1에 따라, 플레이 아이템#1, 플레이 아이템#2, 플레이 아이템#3, 플레이 아이템#4, …의 순서로 재생이 이루어진다. 플레이 아이템#1으로부터 클립 인포메이션 파일 "00001.clpi"이 참조되어, 클립 AV 스트림 파일 "00001.m2ts"이 재생되고, 플레이 리스트#1의 기술에 따라 다음에 플레이 아이템#2으로부터 클립 인포메이션 파일 "00002.clpi"이 참조되어, 클립 AV 스트림 파일 "00002.m2ts"이 재생된다.
- [0338] 플레이 아이템#1의 필드(ConnectionCondition)의 값이 "6"이 되어 제 2의 심레스 접속이 지정되어 있기 때문에, 클립 AV 스트림 파일 "00001.m2ts"과 클립 AV 스트림 파일 "00002.m2ts"은, 심레스하게 접속되어 재생된다. 이때, 오디오 데이터는, 클립 AV 스트림 파일 "00001.m2ts"의 후단의 시작과, 클립 AV 스트림 파일 "00002.m2ts"의 선두의 시작이 대응하도록 기록 제어되어 있기 때문에, 클립 AV 스트림 파일 "00001.m2ts"에 의한 오디오와, 클립 AV 스트림 파일 "00002.m2ts"에 의한 오디오는, 연속한 오디오로서 재생된다.
- [0339] 플레이 리스트#1의 기술에 따라, 플레이 아이템#2의 다음에 플레이 아이템#3에 의한 재생이 이루어진다. 즉, 플레이 아이템#3으로부터 클립 인포메이션 파일 "00003.clpi"이 참조되어, 클립 AV 스트림 파일 "00003.m2ts"이 재생된다.
- [0340] 플레이 아이템#2의 필드(ConnectionCondition)의 값이 "6"이 되어 제 2의 심레스 접속이 지정되어 있기 때문에, 클립 AV 스트림 파일 "00002.m2ts"과 클립 AV 스트림 파일 "00003.m2ts"은, 심레스하게 접속되어 재생된다. 이 경우에도, 오디오 데이터는, 클립 AV 스트림 파일 "00002.m2ts"의 후단의 시작과, 클립 AV 스트림 파일 "00003.m2ts"의 선두의 시작이 대응하도록 기록 제어되어 있기 때문에, 클립 AV 스트림 파일 "00002.m2ts"에 의한 오디오와, 클립 AV 스트림 파일 "00003.m2ts"에 의한 오디오는, 연속한 오디오로서 재생된다.
- [0341] 따라서 연속적인 기록에 의한 기록 구간(400)(도 35A 참조)의 기록 시작시각(ST₁)부터 기록 정지시각(ED₁)까지의 사이에, 소정 사이즈로 3파일로 분할되어 기록된 클립 AV 스트림 파일 "00001.m2ts", "00002.m2ts" 및 "00003.m2ts"은, 재생시에는, 유저에게 파일 분할을 의식시키는 일 없이, 마치 하나의 연속한 클립 AV 스트림 파일에 기록된 비디오 데이터 및 오디오 데이터인 것 같이, 연속적으로 재생된다.
- [0342] 한편, 시각(ED₁)에서의 기록 정지 조작의 후에, 시각(ST₂)에서 기록 시작 조작이 이루어져서 기록된 클립 AV 스트림 파일 "00003.m2ts"은, 플레이 리스트#1에 따라, 상술한 플레이 아이템#3에 의한 재생의 다음에, 플레이 아이템#4에 의거하여 재생된다. 이때, 플레이 아이템#3의 필드(ConnectionCondition)에는, 제 1의 심레스 접속을 나타내는 값 "5"이 설정되어 있다. 그 때문에, 플레이 아이템#3에 의해 클립 AV 스트림 파일 "00003.m2ts"의 후단에서, 오디오 데이터가 비디오 데이터보다 약간 길게 기록되어 있다.
- [0343] 예를 들면, 이 부분에 페이드아웃 등 소망하는 음향 효과를 가하여, 다음의 플레이 아이템#4에 의한 클립 AV 스트림 파일 "00004.m2ts"의 선두 부분의 오디오 데이터와 믹스하여, 클립 AV 스트림 파일 "00003.m2ts"과 클립 AV 스트림 파일 "00004.m2ts"을 연속적으로 재생한다. 이와 같이 함으로써, 장면이 다른 2개의 클립 AV 스트림 파일을, 위화감을 억제하면서 심레스하게 재생할 수 있다.
- [0344] 다음에, 본 발명의 실시의 한 형태의 다른 예에 관해 설명한다. 상술에서는, 본 발명이 단체(單體)의 기록 장치에 적용된 예에 관해 설명하였다(도 32 참조). 이에 대해, 이 실시의 한 형태의 다른 예에서는, 본 발명을, 활상 소자와, 피사체로부터의 광을 활상 소자에 입사시키는 광학계를 가지며, 활상 소자로 활상된 활상 신호에 의거하여 비디오 데이터를 기록 매체에 기록하도록 한, 비디오 카메라 장치에 적용하였다.
- [0345] 도 36은, 본 발명의 실시의 한 형태의 다른 예에 의한 비디오 카메라 장치(100)의 한 예의 구성을 도시한다. 비디오 카메라 장치(100)에서, 기록계의 구성은, 도 32를 이용하여 설명한 기록 장치의 구성을 개략 그대로 적용할 수 있기 때문에, 도 32와 공통되는 부분에는 동일한 부호를 붙이고, 상세한 설명을 생략한다.
- [0346] 도 36의 구성에서, 카메라부(50)는, 영상 신호에 관한 구성으로서, 광학계(51), 활상 소자(52), 활상 신호 처리부(53), 카메라 제어부(54) 및 표시부(55)를 가지며, 음성 신호에 관한 구성으로서, 마이크로폰(MIC)(56) 및 음성 신호 처리부(57)를 갖는다. 제어부(30)는, 카메라부(50)의 각 부분과의 사이에서 각종 제어 신호나 정보의 교환을 행하고, 카메라부(50)의 동작을 제어한다. 또한, 제어부(50)는, 유저 조작에 따라 UI부(31)로부터 공급되는 제어 신호에 의거하여, 카메라부(50)의 동작을 제어한다.
- [0347] 또한, 비디오 카메라 장치(100)로서 구성되는 경우, 기록 시작 조작 및 기록 정지 조작은, 예를 들면, UI부(31)에 마련된 단일의 기록 스위치를 이용하고, 해당 기록 스위치가 압하될 때마다 기록 시작 및 기록 정지가 교대로 지시되도록 이루어지는 것이 일반적이다. 또한, 이 비디오 카메라 장치(100)에서는, 기록 매체(20)로서, Blu-ray Disc나 기록 가능한 타입의 DVD라는, 디스크 기록 매체를 적용하는 것으로 한다.

- [0348] 물론, 이들로 한정하지 않고, 비디오 카메라 장치(100)에 하드 디스크 드라이브를 내장시키고(도시 생략), 이 하드 디스크 드라이브를 기록 매체(20)로서 이용하여도 좋다. 또한, 소정의 연속 기록시간분의 기록을 행한 경우의 데이터가 기억 가능한 용량을 갖는, 불휘발성의 반도체 메모리를 비디오 카메라 장치(100)에 내장 또는 착탈 가능하게 마련하고, 이 불휘발성 메모리를 기록 매체(20)로서 이용할 수도 있다.
- [0349] 카메라부(50)에 있어서, 광학계(51)는, 피사체로부터의 광을 활상 소자(52)에 유도하기 위한 렌즈계, 조리개 조정 기구, 포커스 조정 기구, 줌 기구, 셔터 기구 등을 구비한다. 조리개 조정 기구, 포커스 조정 기구, 줌 기구 및 셔터 기구의 동작은, 제어부(30)로부터 공급되는 제어 신호에 의거하여, 카메라 제어부(54)에 의해 제어된다.
- [0350] 활상 소자(52)는, 예를 들면 CCD(Charge Coupled Device)로 이루어지고, 광학계(51)를 통하여 조사된 광을 광전변환에 의해 전기신호로 변환하고, 소정의 신호 처리를 시행하여 활상 신호로서 출력한다. 활상 신호 처리부(53)는, 활상 소자로부터 출력된 활상 신호에 대해 소정의 신호 처리를 시행하고, 베이스밴드의 디지털 비디오 데이터로서 출력한다.
- [0351] 예를 들면 활상 신호 처리부(53)는, 활상 소자(52)로부터 출력된 활상 신호에 대해, CDS(Correlated Double Sampling) 회로에 의해 화상 정보를 갖는 신호만을 샘플링함과 함께, 노이즈를 제거하고, AGC(Auto Gain Control) 회로에 의해 게인을 조정한다. 그리고, A/D 변환에 의해 디지털 신호로 변환한다. 또한, 활상 신호 처리부(53)는, 이 디지털 신호에 대해 검파계의 신호 처리를 시행하여, R(적색), G(녹색) 및 B(청색) 각 색의 성분을 추출하고, γ 보정이나 화이트 밸런스 보정 등의 처리를 행하여, 최종적으로 1개의 베이스밴드의 디지털 비디오 데이터로서 출력한다.
- [0352] 또한, 활상 신호 처리부(53)는, 활상 소자(52)로부터 출력된 활상 신호의 정보를 제어부(30)에 보낸다. 제어부(30)는, 이 정보에 의거하여 광학계(51)를 제어하기 위한 제어 신호를 생성하고, 카메라 제어부(54)에 공급한다. 카메라 제어부(54)는, 이 제어 신호에 의거하여 포커스 조정 기구나 조리개 조정 기구 등의 제어를 행한다.
- [0353] 또한, 활상 신호 처리부(53)는, 활상 소자(52)로부터 출력된 활상 신호에 의거하여, 예를 들면 LCD(Liquid Crystal Display)를 표시 소자로서 이용한 표시부(55)에 영출시키는 영상 신호를 생성한다.
- [0354] 한편, 마이크로폰(56)은, 주위의 음성을 수음하여 전기신호로 변환하여 출력한다. 마이크로폰(56)으로부터 출력된 음성 신호는, 음성 신호 처리부(57)에 공급된다. 음성 신호 처리부(57)는, 공급된 음성 신호를, 리미터를 통하고 나서 A/D 변환을 시행하여 디지털 오디오 데이터로 하고, 노이즈 제거나 음질 보정 등 소정의 음성 신호 처리를 시행하여 베이스밴드의 디지털 오디오 데이터로서 출력한다.
- [0355] 카메라부(50)의 활상 신호 처리부(53)로부터 출력된 베이스밴드의 디지털 비디오 데이터는, 기록부(10)의 단자(40)에 공급된다. 또한, 음성 신호 처리부(57)로부터 출력된 베이스밴드의 디지털 오디오 데이터는, 기록부(10)의 단자(41)에 공급된다.
- [0356] 기록 정지 상태로부터 UI부(31)에 마련된 기록 스위치가 압하되면, 기록 시작을 지시하는 제어 신호가 UI부(31)로부터 제어부(30)에 공급되고, 제어부(30)의 제어에 의거하여 카메라부(50)로부터 출력된 베이스밴드의 디지털 비디오 신호 및 디지털 오디오 데이터의 기록 매체(20)에의 기록이 시작된다.
- [0357] 즉, 이미 설명한 바와 같이, 제어부(30)의 제어에 의거하여 비디오 인코더(11) 및 오디오 인코더(12)의 동작이 시작되고, 비디오 데이터 및 오디오 데이터가 각각 비디오 인코더(11) 및 오디오 인코더(12)에서 압축 부호화되고, 멀티플렉서(13)에서 정해진대로 패킷화되고 다중화되어 AV 스트림 데이터가 된다. AV 스트림 데이터는, 스트림 버퍼(14)를 통하여, 기록 제어부(15)에 공급되고, 클립 AV 스트림 파일로서 기록 매체(20)에 기록된다.
- [0358] 기록 시작 조작부터 소정 시간 이상, 기록이 연속되고, 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과할 때에, 이미 설명한 바와 같이, 기록중의 클립 AV 스트림 파일을 클로즈하여 대응하는 클립 인포메이션 파일을 작성하고, 해당 클립 인포메이션 파일을 참조하는 플레이 아이템이 생성된다. 여기서 생성되는 플레이 아이템은, 제 2의 심레스 접속이 지정된다. 생성된 플레이 아이템은, 플레이 리스트에 추가기록된다. 또한, 신규로 클립 AV 스트림 파일이 작성되고, 계속되는 기록에 의해 공급되는 AV 스트림 데이터가, 신규로 작성된 이 클립 AV 스트림 파일에 기록된다.
- [0359] UI부(31)의 기록 스위치가 압하되면, 기록이 정지되고, 클립 인포메이션 파일의 작성이나, 플레이 리스트 파일의 갱신이 행하여진다. 관리 정보 처리부(16)는, 멀티플렉서(13) 및 기록 제어부(15)로부터의 정보에 의거하여,

기록 매체(20)에 기록되는 클립 AV 스트림 파일에 대응하는 클립 인포메이션 파일을 작성한다. 또한, 관리 정보 처리부(16)는, 해당 클립 인포메이션 파일을 참조하는 플레이 아이템을 생성하고, 이미 플레이 리스트가 존재하는 경우에는, 생성된 플레이 아이템을 해당 플레이 리스트에 대해 추가함과 함께, 플레이 리스트에 대해 플레이 리스트 마크를 찍는다.

[0360] 이 상태로부터 재차 기록 스위치가 압하되면, 재차 기록 시작이 지시되어, 새로운 클립 AV 스트림 파일의 기록 매체(20)에의 기록이 시작됨과 함께, 대응하는 클립 인포메이션 파일의 작성이나, 해당 클립 인포메이션 파일을 참조하는 플레이 아이템의 생성, 생성된 플레이 아이템의 플레이 리스트에의 추가 등의 처리가 행하여진다. 클립 AV 스트림 파일 사이의 재생은, 제 1의 심레스 접속에 의해 접속된다.

[0361] 이 실시의 한 형태의 다른 예와 같이, 본 발명을 비디오 카메라 장치(100)에 적용하는 경우, 장시간의 연속 촬영을 행하고, 기록 시작 조작부터 기록 정지 조작까지의 사이에 기록된 클립 AV 스트림 파일의 파일 사이즈가 소정 사이즈를 초과할 때에, 클립 AV 스트림 파일이 자동적으로 분할되고, 1파일의 사이즈가 소정 사이즈를 초과하지 않도록 된다. 또한, 분할시에, 분할 전후의 클립 AV 스트림 파일의 접속 방법으로서 제 2의 심레스 접속이 설정되기 때문에, 재생시에는, 소정 사이즈에 의거한 클립 AV 스트림 파일의 분할을 의식하는 일 없이, 1회의 연속 촬영으로 기록된 비디오 데이터 및 오디오 데이터를 연속적으로 재생하는 것이 가능해진다.

[0362] 또한, 상술에서는, 클립 AV 스트림 파일의 분할을 행하는지의 여부를, 파일 사이즈에 의거하여 판단하는 예에 관해 설명하였지만, 이것은 이 예로 한정되지 않는다. 즉, 클립 AV 스트림 파일에 대한 관리 정보에 의거하여 파일의 분할을 행하는지의 여부를 판단하는 것도 고려된다. 예를 들면, 이미 설명한 바와 같이, 클립 AV 스트림 파일의 어드레스 정보와, 해당 파일에 기록되는 비디오 데이터 및 오디오 데이터의 시각 정보와의 대응 관계를 나타내는 EP 엔트리 정보의 상한에 의거하여, 클립 AV 스트림 파일의 분할을 행하는지의 여부를 판단한다.

[0363] EP 엔트리는, 도 20 내지 도 23을 이용하여 이미 설명한 바와 같이, 소스 패킷에 의거한 엔트리와, PTS에 의거한 엔트리를 각각 가지며, 또한, 각각의 엔트리는, 거친 검색을 행하는 엔트리와 정밀한 검색을 행하는 엔트리로 이루어진다. 각각의 엔트리는, 도 22 및 도 23에 예시한 바와 같이, 포맷에 의해 엔트리를 표현하기 위한 비트 수가 규정되어 있기 때문에, 이 비트 수로 표현 가능한 최대치가 엔트리 수의 최대 수에 대응한다. 한 예로서, 제어부(30)는, 관리 정보 처리부(16)로부터의 엔트리 포인트에 관한 정보에 의거하여, 현재 기록중의 클립 AV 스트림 파일에 관해, 엔트리 포인트 수가 이 최대치에 달하였는지의 여부를 판단한다.

[0364] 한 예로서, 도 33의 플로우 차트를 참조하면, 제어부(30)는, 관리 정보 처리부(16)에서 생성되는 EP 엔트리 정보를 감시하고, 하나의 클립 AV 스트림 파일에 대해 생성되는 EP 엔트리의 수가 소정수로 되었는지의 여부를 판단한다. 그리고, EP 엔트리 수가 소정수가 되었다고 판단되면, 처리를 스텝 S62로 이행시키고, 스트림 베퍼(14)에 저장되어 있는 스트림을 정해진대로 판독하여 클립 AV 스트림 파일에 기록하고, 해당 클립 AV 스트림 파일을 클로즈한다(스텝 S63). 그리고, 해당 클립 AV 스트림 파일에 대응하는 클립 인포메이션 파일을 작성하고(스텝 S64), 대응하는 플레이 아이템 내의 필드(ConnectionCondition)의 값을 제 2의 심레스 접속을 나타내는 값"6"으로 설정한다(스텝 S65). 그리고, 스텝 S66 및 스텝 S67에서 플레이 아이템의 정보를 생성함과 함께, 생성된 플레이 아이템을 플레이 리스트에 대해 추가 기록한다. 필요하면, 스텝 S68에서 플레이 리스트 마크의 설정을 행한다.

[0365] 또한, 클립 AV 스트림 파일을 분할하는지의 여부의 판단은, EP 엔트리 수로 한정하지 않고, 예를 들면 PTS의 상한에 의거하여 행하여도 좋다.

[0366] 또한, 상술에서는, 도 32에 도시하는 기록 장치나 도 36에 도시하는 비디오 카메라 장치(100)의 기록부(10)가 하드웨어적으로 구성되도록 설명하였지만, 이것은 이 예로 한정되지 않는다. 즉, 기록부(10)는, 소프트웨어로서 구성하는 것도 가능하다. 이 경우, 소프트웨어는, 예를 들면 제어부(30)가 갖는 도시되지 않은 ROM에 미리 기억된다. 이것으로 한정하지 않고, 기록부(10)를, 퍼스널 컴퓨터 등의 컴퓨터 장치상에 구성하는 것도 가능하다. 이 경우에는, 기록부(10)를 컴퓨터 장치에 실행시키는 소프트웨어는, CD-ROM이나 DVD-ROM이라는 기록 매체에 기록되어 제공된다. 컴퓨터 장치가 네트워크 접속 가능한 경우, 인터넷 등의 네트워크를 통하여 해당 소프트웨어를 제공하는 것도 가능하다.

도면의 간단한 설명

[0023] 도 1은 본 발명에 적용 가능한 AVCHD 포맷에 규정되는 데이터 모델을 개략적으로 도시하는 간략선도,

[0024] 도 2는 인덱스 테이블을 설명하기 위한 간략선도,

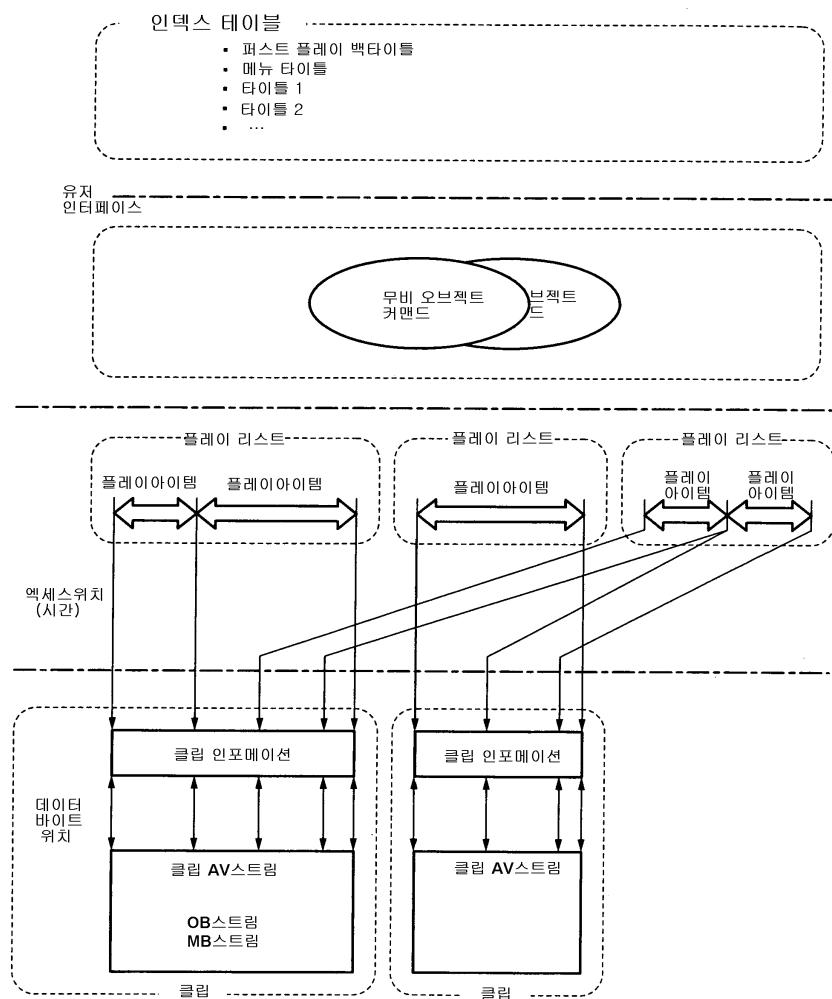
- [0025] 도 3은 클립 AV 스트림, 클립 정보, 클립, 플레이 아이템 및 플레이 리스트의 관계를 도시하는 UML도,
- [0026] 도 4는 복수의 플레이 리스트로부터 동일한 클립을 참조하는 방법을 설명하기 위한 간략선도,
- [0027] 도 5는 기록 매체에 기록되는 파일의 관리 구조를 설명하기 위한 간략선도,
- [0028] 도 6은 파일"index.bdmv"의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0029] 도 7은 블록(blkIndexes())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0030] 도 8은 파일 "MovieObject.bdmv"의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0031] 도 9는 블록(blkMovieObjects())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0032] 도 10은 플레이 리스트 파일"XXXXX.mpls"의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0033] 도 11은 블록(blkPlayList())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0034] 도 12는 블록(blkPlayItem())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0035] 도 13A 및 도 13B는 제 1 및 제 2의 심레스 접속을 설명하기 위한 간략선도,
- [0036] 도 14는 블록(blkPlayListMark())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0037] 도 15는 클립 인포메이션 파일의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0038] 도 16은 블록(blkClipInfo())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0039] 도 17은 블록(blkSequenceInfo())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0040] 도 18은 블록(blkProgramInfo())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0041] 도 19는 블록(blkCPI())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0042] 도 20은 블록(blkEPMap())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0043] 도 21은 블록(blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf))의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0044] 도 22는 엔트리(PTSEPCoarse) 및 엔트리(PTSEPFine)의 한 예의 포맷에 관해 도시하는 간략선도,
- [0045] 도 23은 엔트리(SPNEPCoarse) 및 엔트리(SPNEPFine)의 한 예의 포맷에 관해 도시하는 간략선도,
- [0046] 도 24는 블록(blkExtensionData())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0047] 도 25는 블록(blkExtensionData())에서의 각 데이터의 참조 관계를 모식적으로 도시하는 간략선도,
- [0048] 도 26은 블록(blkExtensionData())에 데이터를 기록할 때의 한 예의 처리를 도시하는 플로우 차트,
- [0049] 도 27은 블록(blkExtensionData())으로부터 확장 데이터를 판독할 때의 한 예의 처리를 도시하는 플로우 차트,
- [0050] 도 28은 파일"index.bdmv" 내의 필드(blkExtensionData())에서의 블록(DataBlock())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0051] 도 29는 블록(blkTableOfPlayList())의 한 예의 구조를 표시하는 신택스를 도시하는 간략선도,
- [0052] 도 30A 및 도 30B는 가상 플레이어의 동작을 개략적으로 도시하는 플로우 차트,
- [0053] 도 31은 가상 플레이어의 동작을 개략적으로 도시하는 간략선도,
- [0054] 도 32는 본 발명의 실시의 한 형태에 적용 가능한 기록 장치의 한 예의 구성을 개략적으로 도시하는 블록도,
- [0055] 도 33은 본 발명의 실시의 한 형태에 의한 한 예의 기록 방법을 도시하는 플로우 차트,
- [0056] 도 34는 본 발명의 실시의 한 형태의 방법에 의해 기록된 한 예의 파일 구조를 도시하는 간략선도,
- [0057] 도 35A 및 도 35B는 기록의 시간 경과의 예를 설명하기 위한 간략선도,
- [0058] 도 36은 본 발명의 실시의 한 형태의 다른 예에 의한 비디오 카메라 장치의 한 예의 구성을 도시하는 블록도.

- [0059] (도면의 주요 부분에 대한 부호의 설명)
- [0060] 10 : 기록부 11 : 비디오 인코더
- [0061] 12 : 오디오 인코더 13 : 멀티플렉서
- [0062] 14 : 스트림 베퍼 15 : 기록 제어부
- [0063] 16 : 관리 정보 처리부 17 : 휘발성 메모리
- [0064] 18 : 불휘발성 메모리 20 : 기록 매체
- [0065] 30 : 제어부 31 : 유저 인터페이스부
- [0066] 50 : 카메라부 100 : 비디오 카메라 장치
- [0067] S10 : 기록하고자 하는 데이터의 길이를 얻어서, ExtDataLength(n+1)의 값에 세트한다
- [0068] S11 : 현재의 blkExtensionData()의 열거되어 있는 ext_data_entry()로부터 ExtDataLength와 ExtDataStartAddress를 조사하고, DataBlock중의 사용 상황을 안다
- [0069] S12 : DataBlock중에 ExtDataLength(n+1) 이상이 연속한 빈 곳이 있는지?
- [0070] S13 : blkExtensionData()의 Length값을 크게 하여, ExtDataLength(n+1) 이상이 연속한 빈 영역을 만든다
- [0071] S14 : 데이터를 격납하는 영역의 선두 어드레스를 정하고, ExtDataStartAddress(n+1)로 한다
- [0072] S15 : ExtDataStartAddress(n+1)로 나타나는 어드레스로부터, ExtDataLength(n+1)의 길이의 데이터를 기록한다
- [0073] S16 : ext_data_entry()에 ExtDataLength(n+1)와 ExtDataStartAddress(n+1)를 추가한다
- [0074] S20 : 판독하고자 하는 데이터가 준거하는 규격으로부터, ExtDataType을 안다
- [0075] S21 : ExtDataType을 기초로, 판독하고자 하는 데이터의 종류별로부터 ExtDataVersion을 안다
- [0076] S22 : blkExtensionData()에 열거되어 있는 ext_data_entry()를 순서로 하나 판독한다
- [0077] S23 : ExtDataType, ExtDataVersion이 일치 한지?
- [0078] S24 : ExtDataLength(i)와 ExtDataStartAddress(i)를 판독한다
- [0079] S25 : ExtDataStartAddress(i)로 나타나는 어드레스로부터, ExtDataLength(i)의 길이의 데이터를 판독한다
- [0080] S26 : ext_data_entry()를 다 판독하였는지?
- [0081] S27 : 판독하고자 하는 데이터는 존재하지 않는다
- [0082] S50 : 기록 시작 조작
- [0083] S51 : 신규 클립 AV 스트림 파일 작성
- [0084] S52 : 스트림을 클립 AV 스트림 파일에 기록
- [0085] S53 : 2GB 초과하는지?
- [0086] S54 : 기록 정지 조작?
- [0087] S55 : 스트림 베퍼에 저장되어 있는 스트림을 파일에 기록한다
- [0088] S56 : 클립 AV 스트림 파일을 클로즈한다
- [0089] S57 : 클립 인포메이션 파일 작성
- [0090] S58 : ConnectionCondition=5로 설정
- [0091] S59 : NumberOfPlayItems를 + 1
- [0092] S60 : ClipInformationFileName, INTIME, OUTTIME의 설정, blkPlayItem()의 작성
- [0093] S61 : MarkTimeStamp를 INTIME으로 하여 blkPlayListMark를 하나 추가

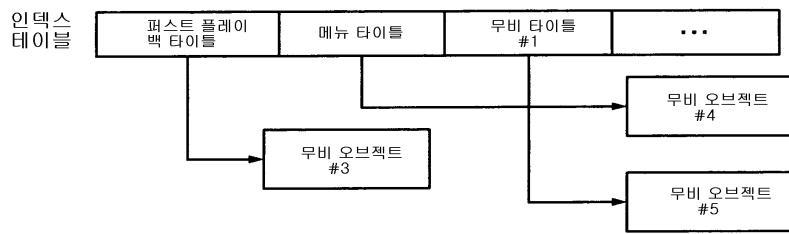
- [0094] S62 : 스트림 버퍼에 저장되어 있는 스트림를 파일에 기록한다
- [0095] S63 : 클립 AV 스트림 파일을 클로즈한다,
- [0096] S64 : 클립 인포메이션 파일 작성
- [0097] S65 : ConnectionCondition=6으로 설정
- [0098] S66 : NumberOfPlayItems를 + 1
- [0099] S67 : ClipInformationFileName, INTIME, OUTTIME의 설정, blkPlayItem()의 작성
- [0100] S68 : (플레이 리스트 마크 추가)

도면

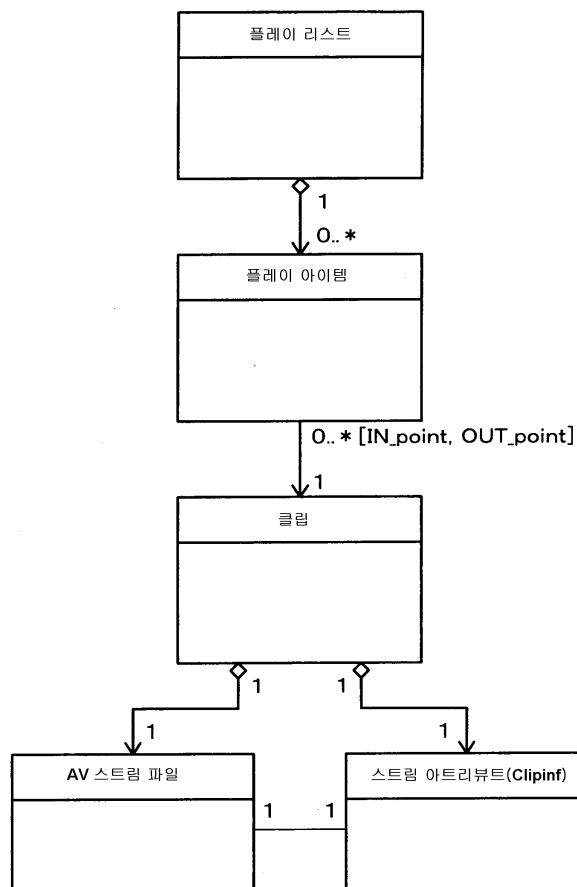
도면1



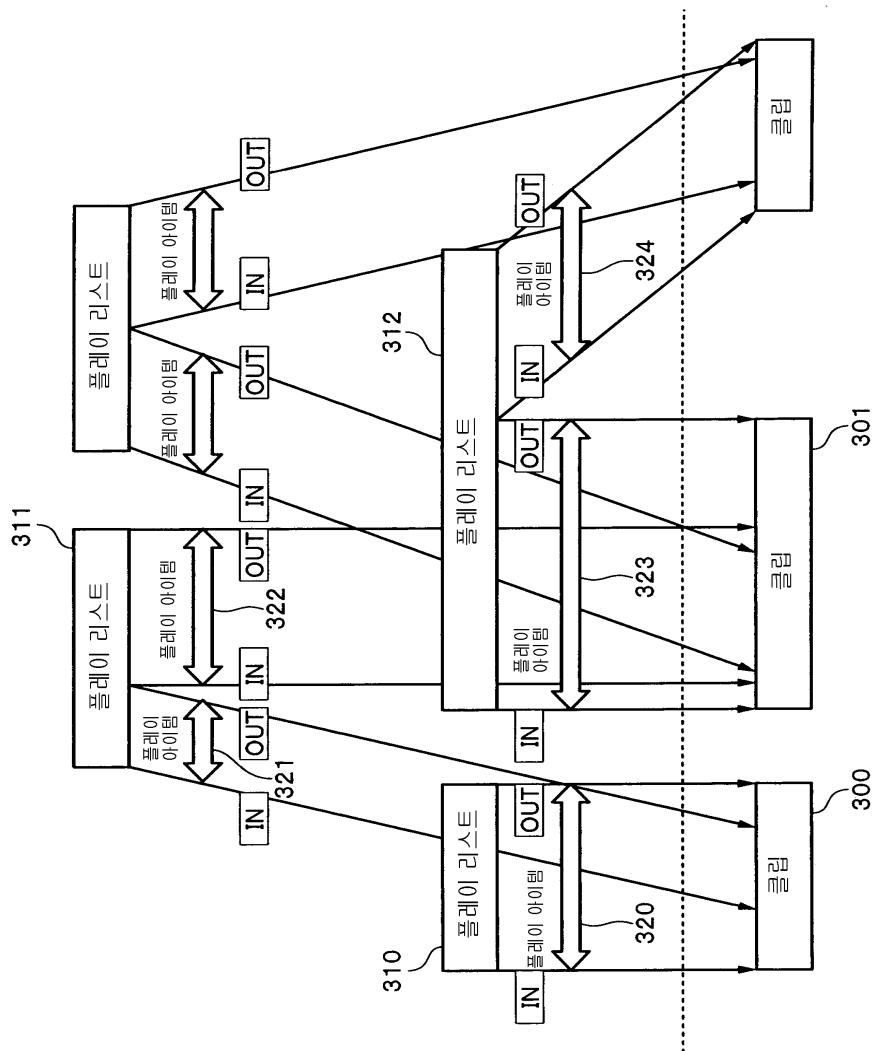
도면2



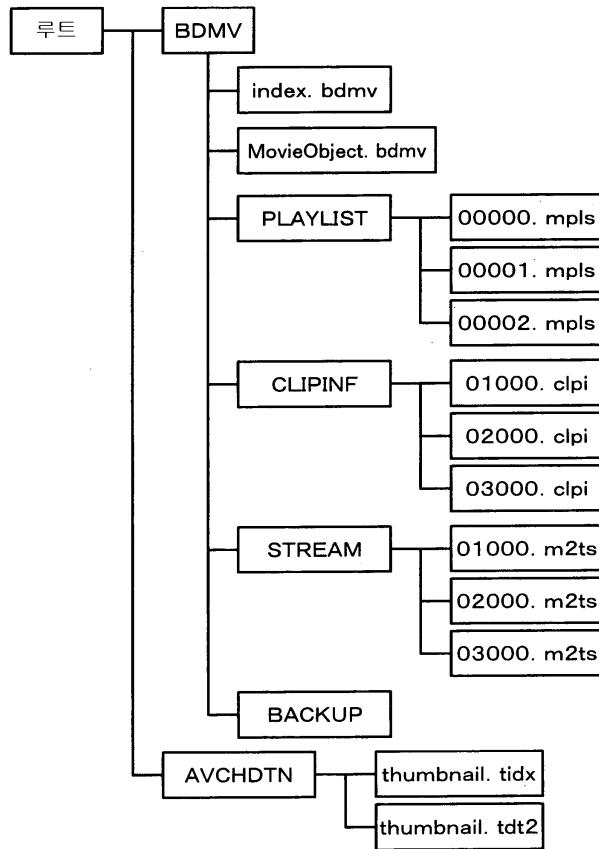
도면3



도면4



도면5



도면6

신택스	데이터 길이 (비트)	나모닉 (mnemonic)
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

선택스	데이터 길이 (비트)	나모닉
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 (<i>title_id</i> =0; <i>title_id</i> < NumberOfTitles; <i>title_id</i> ++) {		
MovieTitle[title_id]()		
reserved	1	bslbf
'1'	1	bslbf
reserved	46	bslbf
MovieTitleMobjIDRef[title_id]	16	uimsbf
reserved	32	bslbf
}		
}		
}		

도면8

선택스	데이터 길이 (비트)	나모닉
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

선택스	데이터 길이 (비트)	나모닉
blkMovieObjects()		
Length	32	uimsbf
reserved	32	bslbf
NumberOfMobjs	16	uimsbf
for (<i>mobj_id</i> =0; <i>mobj_id</i> < <i>NumberOfMobjs</i> ; <i>mobj_id</i> ++) {		
MovieObject[<i>mobj_id</i>]()		
TerminalInfo()		
'1'	1	bslbf
reserved	15	bslbf
}		
NumberOfNavigationCommands[<i>mobj_id</i>]	16	uimsbf
for (<i>command_id</i> =0; <i>command_id</i> < <i>NumberOfNavigationCommands[<i>mobj_id</i>]</i> ; <i>command_id</i> ++) {		
NavigationCommand[<i>mobj_id</i>][<i>command_id</i>]	96	bslbf
}		
}		
}		
}		

도면10

선택스	데이터 길이 (비트)	나모닉
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>i</i> =0; <i>i</i> < <i>N1</i> ; <i>i</i> ++) {		
padding_word	16	bslbf
}		
blkPlayList()		
for (<i>i</i> =0; <i>i</i> < <i>N2</i> ; <i>i</i> ++) {		
padding_word	16	bslbf
}		
blkPlayListMark()		
for (<i>i</i> =0; <i>i</i> < <i>N3</i> ; <i>i</i> ++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (<i>i</i> =0; <i>i</i> < <i>N4</i> ; <i>i</i> ++) {		
padding_word	16	bslbf
}		
}		
}		

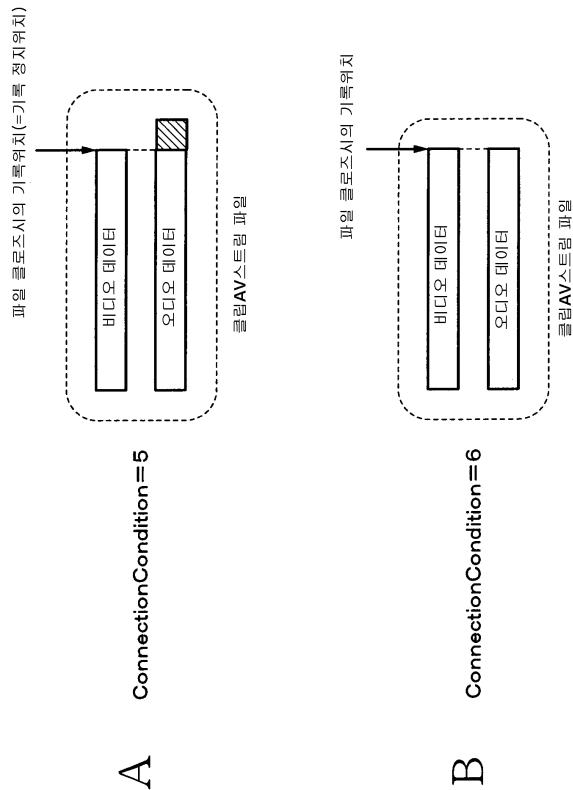
도면11

신택스	데이터길이 (비트)	나모닉
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)	나모닉
blkPlayItem()		
Length	16	uimsbf
ClipInformationFileName	8 * 5	bslbf
ClipCodeIdentifier	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

선택스	데이터길이 (비트)	나모닉
blkPlayListMark() {		
Length	32	uimsbf
NumberOfPlayListMarks	16	uimsbf
for(<i>PL_mark_id</i> =0; <i>PL_mark_id</i> < NumberOfPlayListMarks ; <i>PL_mark_id</i> ++) {		
reserved	8	bslbf
MarkType	8	bslbf
RefToPlayItemID	16	uimsbf
MarkTimeStamp	32	uimsbf
EntryESPID	16	uimsbf
Duration	32	uimsbf
}		
}		

도면15

선택스	데이터 길이 (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

선택스	데이터 길이 (비트)	나모닉
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 == 1b) {		
reserved	8	bslbf
FollowingClipStreamType	8	bslbf
reserved	32	bslbf
FollowingClipInformationFileName	8 * 5	bslbf
ClipCodeIdentifier	8 * 4	bslbf
reserved	8	bslbf
}		
}		

도면17

선택스	데이터 길이 (비트)	나모닉
blkSequenceInfo() {		
Length	32	uimsbf
reserved	15	bslbf
'1'	1	bslbf
SPNATCStart	32	uimsbf
NumberOfSTCSequences	8	uimsbf
reserved	8	uimsbf
for (stc_id=0; stc_id < NumberOfSTCSequences; stc_id++) {		
PCRVID[stc_id]	16	uimsbf
SPNSTCStart[stc_id]	32	uimsbf
PresentationStartTime[stc_id]	32	uimsbf
PresentationEndTime[stc_id]	32	uimsbf
}		
}		

도면18

선택스	데이터 길이 (비트)	나모닉
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)		
}		
}		

도면19

선택스	데이터 길이 (비트)	나모닉
blkCPI() {		
Length	32	uimsbf
if(Length !=0) {		
reserved	12	bslbf
CPIType	4	bslbf
blkEPMap()		
}		
}		

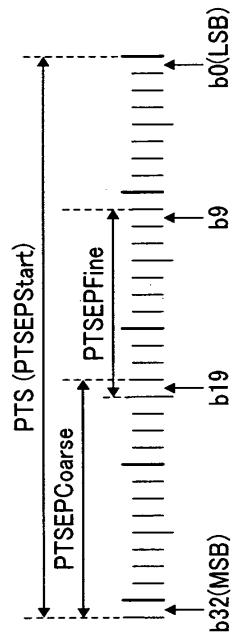
도면20

선택스	데이터 길이 (비트)	나모닉
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
 }		
 }		
}		

도면21

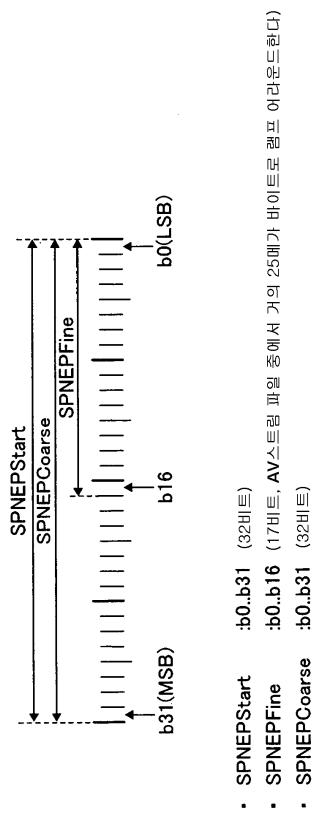
선택스	데이터 길이 (비트)	나모닉
blkEPMapForOneStreamPID(EP_stream_type, Nc, Nf) {		
 EPFineTableStartAddress	32	uimsbf
 for (i=0; i<Nc; i++) {		
// EP coarse table		
 RefToEPFinelD[i]	18	uimsbf
 PTSEPCoarse[i]	14	uimsbf
 SPNEPCoarse[i]	32	uimsbf
 }		
 for (i=0; i<X; i++) {		
 padding_word	16	bslbf
 }		
 for (EP_fine_id =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
 }		
}		

도면22



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

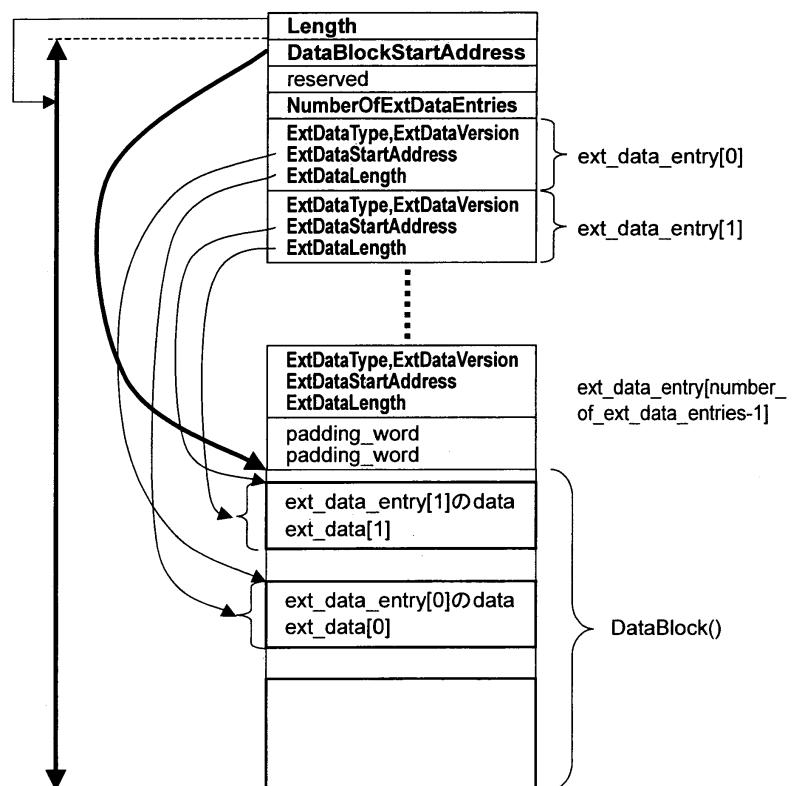
도면23



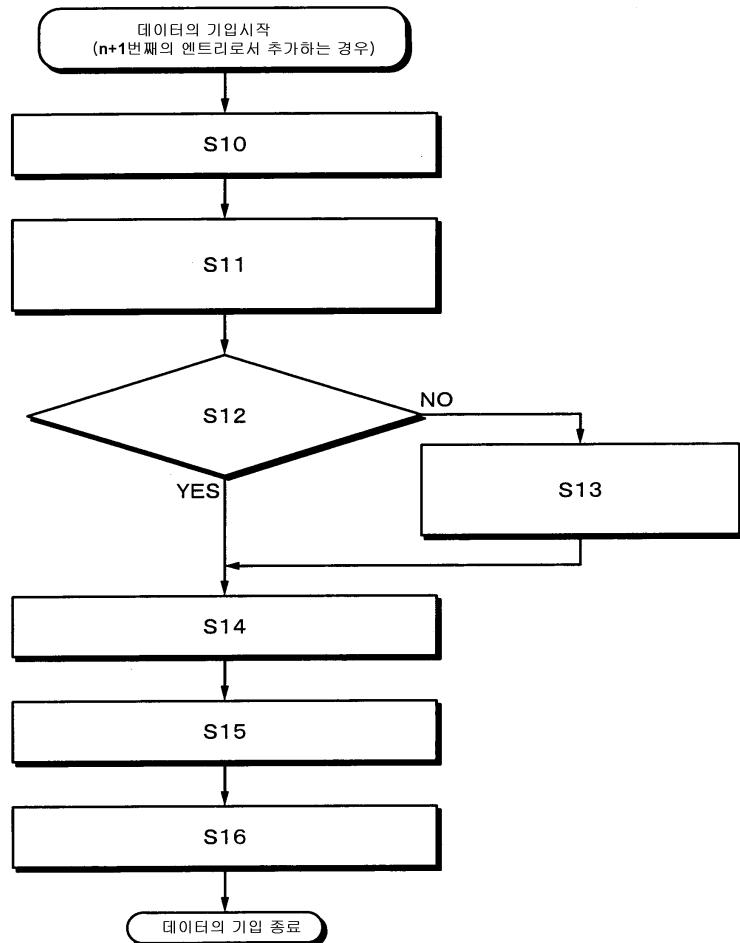
도면24

선택스	데이터 길이 (비트)	나모닉
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)	
}		
}		

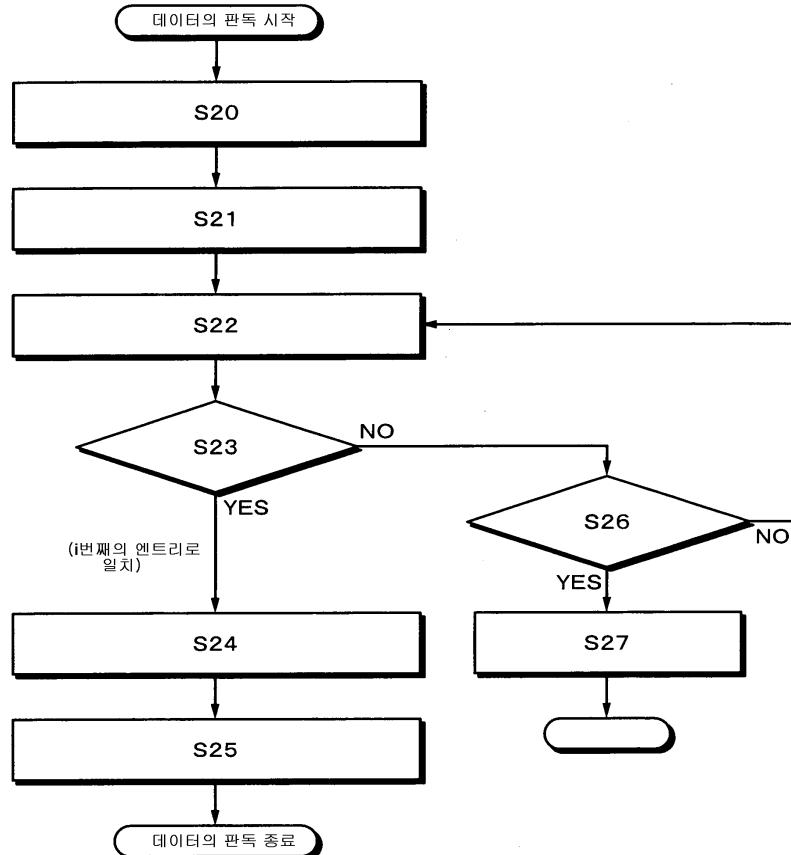
도면25



도면26



도면27



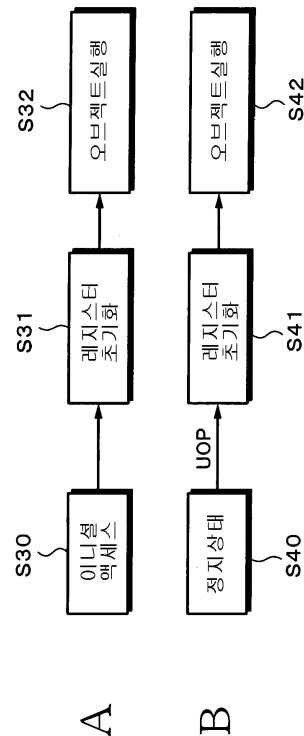
도면28

선택스	데이터 길이 (비트)	나모닉
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
}		
}		

도면29

선택스	데이터 길이 (비트)	나모닉
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
}		
}		
}		

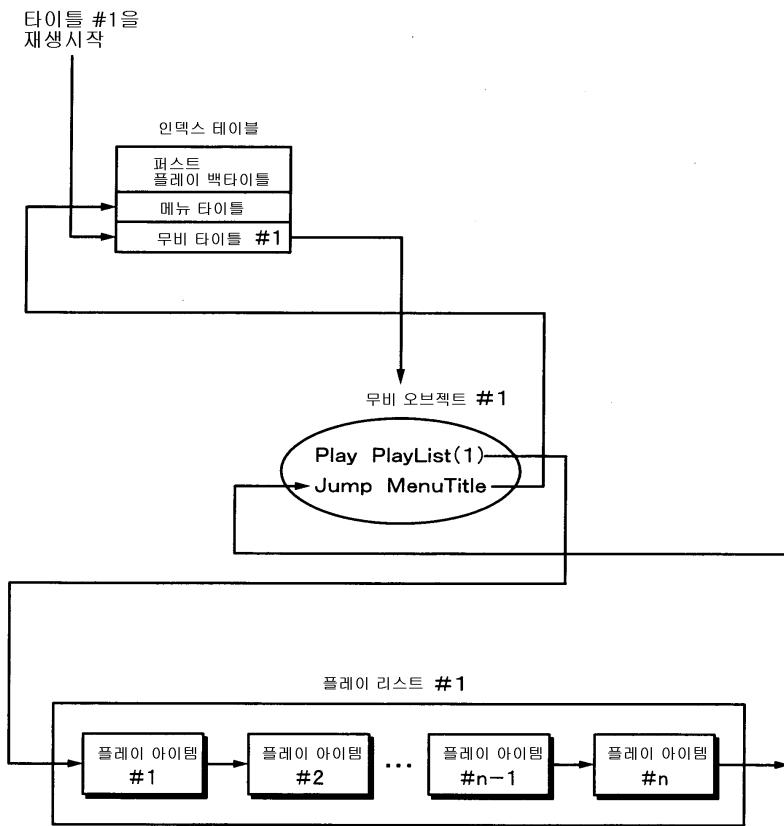
도면30



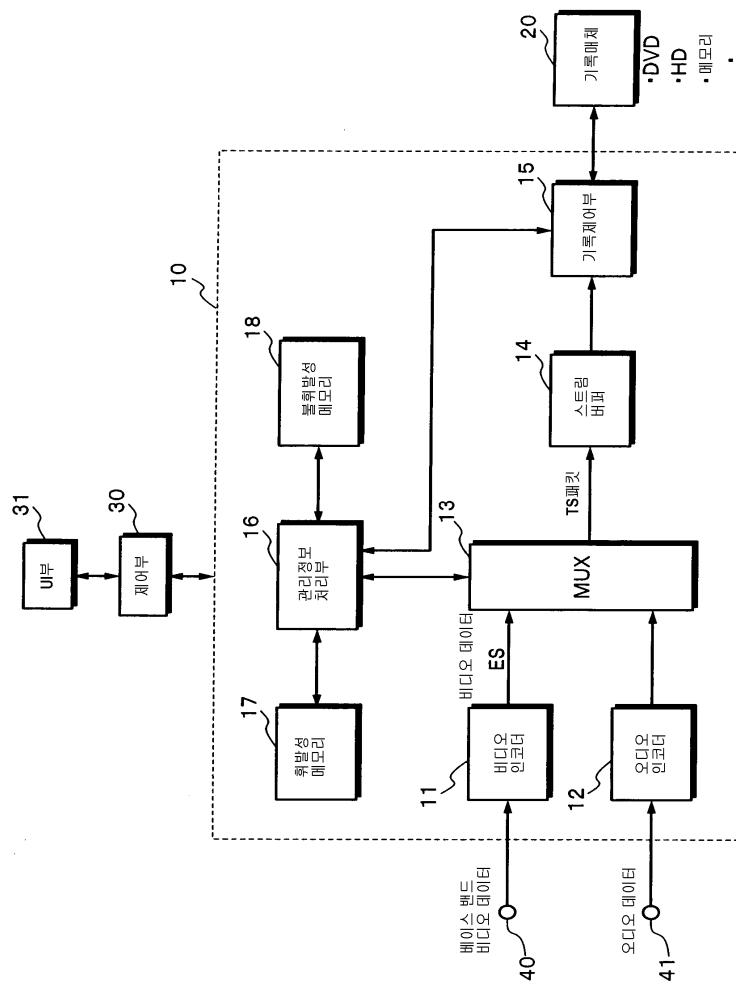
A

B

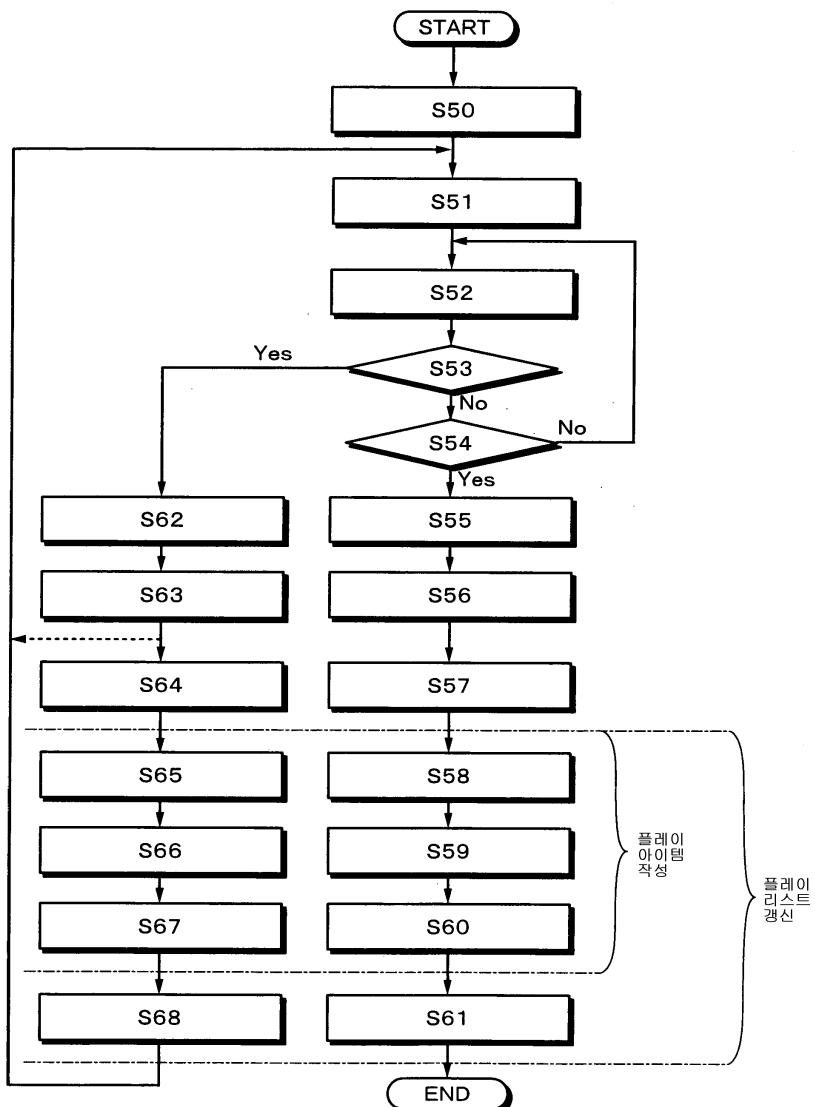
도면31



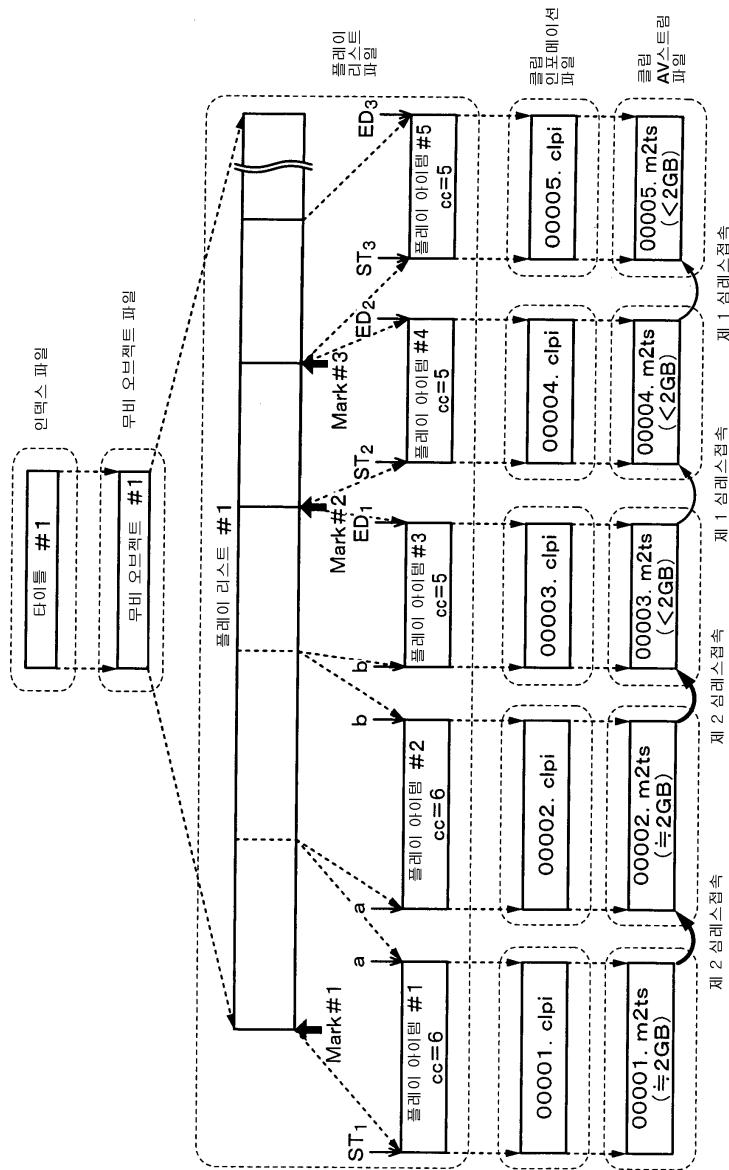
도면32



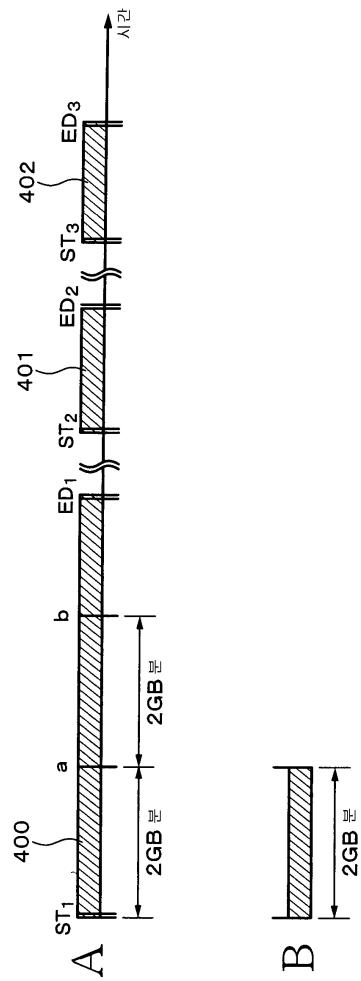
도면33



도면34



도면35



도면36

