



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년11월12일
(11) 등록번호 10-1568680
(24) 등록일자 2015년11월06일

(51) 국제특허분류(Int. Cl.)

G06F 12/16 (2006.01)

(21) 출원번호 10-2014-0089780

(22) 출원일자 2014년07월16일

심사청구일자 2014년07월16일

(56) 선행기술조사문헌

JP2008243138 A

KR1020130081826 A

KR1020140008729 A

KR1020060054979 A

(73) 특허권자

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

이상진

서울특별시 종로구 사직로8길 20 101동 1503호(내수동, 파크펠리스)

박정흠

경기도 부천시 소사구 심곡로10번길 17 904호(심곡본동, 부천아름다운나날아파트)

(74) 대리인

특허법인충현

전체 청구항 수 : 총 14 항

심사관 : 이형일

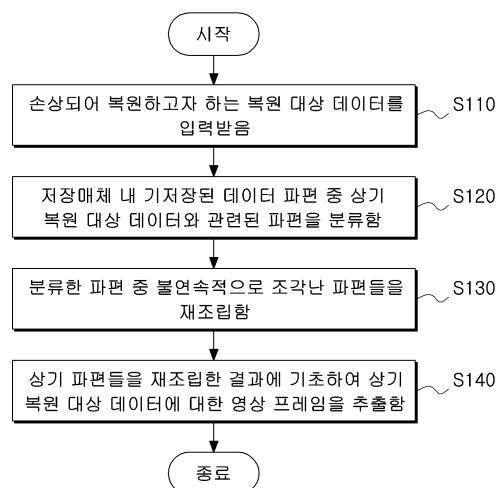
(54) 발명의 명칭 데이터 파편 분류를 이용한 데이터 복원 방법

(57) 요약

본 발명은 데이터 파편 분류를 이용한 데이터 복원 방법에 관한 것으로서, 보다 바람직하게는 입력부가 손상되어 복원하고자 하는 복원 대상 데이터를 입력받는 단계; 파편분류부가 저장매체 내 기저장된 데이터 파편 중 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계; 파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 재조립하는 단계; 및 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계;를 포함한다.

이러한 구성에 의해, 본 발명의 데이터 파편 분류를 이용한 데이터 복원 방법은 복원하고자 하는 데이터 파편과 관련있는 데이터 파편을 분류하고, 분류된 데이터 파편을 재조립하여 영상 프레임을 추출함으로써, 복원하고자 하는 데이터 파편이 불연속적으로 단편화되더라도 데이터를 용이하게 복원할 수 있는 효과가 있다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호 2013054816

부처명 미래창조과학부

연구관리전문기관 한국연구재단

연구사업명 원천기술개발사업-공공복지안전기술개발사업

연구과제명 대용량 저장장치 조사용 포렌식 시스템 개발

기 여 율 1/1

주관기관 고려대학교 산학협력단

연구기간 2013.09.01 ~ 2014.07.31

명세서

청구범위

청구항 1

입력부가 복원하고자 하는 복원 대상 데이터를 입력받는 단계;

파편분류부가 저장매체 내 기저장된 데이터 파편 중 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계;

파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 상기 복원 대상 데이터의 메타 정보와 프레임 구조에 대한 검증을 이용하여 재조립하는 단계; 및

영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계;

를 포함하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 2

제1항에 있어서,

상기 입력부가 대상 데이터를 입력받는 단계는

파편의 분류를 위한 기준 파편의 크기를 설정하는 것을 더 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 3

제2항에 있어서,

상기 기준 파편의 크기는

상기 저장매체의 종류에 따라 섹터, 클러스터, 블록, 페이지 중 적어도 하나에 해당하는 저장 단위인 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 4

제1항에 있어서,

상기 파편분류부가 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계는

상기 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정;

식별된 파편의 유형에 기초하여 상기 복원 대상 데이터와 관련된 파편이 저장매체 내 기저장된 데이터 파편 중에 존재하는지 검색하는 과정; 및

검색된 파편에 기초하여 상기 복원 대상 데이터에 대한 파편 맵을 생성하는 과정;

을 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 5

제4항에 있어서,

상기 파편분류부가 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계는

상기 파편의 유형을 식별하는 과정을 수행하기 전, 해쉬 알고리즘에 기초하여 상기 복원 대상 데이터 내 중복되는 파편을 제거하는 과정;

을 더 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 6

제4항에 있어서,

상기 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정은

상기 파편의 유형이 키 프레임(Key-frame), 델타 프레임(delta-frame), 디코딩 정보, 랜덤 중 적어도 하나인 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 7

제6항에 있어서,

상기 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정은

상기 파편의 유형이 키 프레임 또는 델타 프레임인 경우, 상기 키 프레임 또는 델타 프레임의 헤더 영역에 저장된 메타 정보에 기초하여 영상 프레임의 시작 부분이 포함된 파편을 분류하는 것을 더 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 8

제7항에 있어서,

상기 파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 상기 복원 대상 데이터의 메타 정보와 프레임 구조에 대한 검증을 이용하여 재조립하는 단계는

상기 복원 대상 데이터에 대한 파편 맵을 입력받는 과정;

상기 복원 대상 데이터의 헤더 영역에 저장된 메타 정보의 유효성을 검증하는 과정;

상기 복원 대상 데이터의 영상 프레임에 대한 구조를 검증하는 과정; 및

상기 파편의 메타 정보에 대한 유효성 및 영상 프레임의 구조에 대한 검증 결과에 기초하여 영상 프레임 체인을 재조립하는 과정;

을 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 9

제8항에 있어서,

상기 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계는

상기 영상 프레임 체인에 대한 메타 정보가 존재하는지 검색하는 과정;

상기 영상 프레임 체인에 대한 메타 정보가 존재하는 경우, 상기 메타 정보를 획득하는 과정;

추출하고자 하는 영상 프레임의 이름을 설정하는 과정; 및
 획득한 상기 메타 정보에 기초하여 이미지 파일 형태의 영상 프레임을 추출하는 과정;
 을 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 10

제9항에 있어서,
 상기 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계는
 상기 이미지 파일을 시계열 순으로 나열하여 동영상 파일 형태의 영상 프레임을 추출하는 과정;
 을 더 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 11

제9항에 있어서,
 상기 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계는
 상기 영상 프레임 체인에 대한 메타 정보가 존재하지 않는 경우, 상기 영상 프레임 체인의 데이터를 디코딩하여 이미지 파일 형태의 영상 프레임을 추출하는 과정;
 추출된 상기 이미지 파일에 대하여 이미지 프로세싱 기법을 적용하여 적어도 하나의 이미지 집합을 분류하는 과정;
 을 더 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 12

제1항에 있어서,
 상기 복원 대상 데이터는
 파일 시스템의 비할당 영역, 미사용 영역, 단일 파일의 비할당 영역, 미사용 영역, 손상된 파일, 로우 덤프(raw dump) 중 적어도 하나인 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 13

제1항에 있어서,
 상기 복원 대상 데이터는
 컨테이너 메타 정보, NAL 헤더 정보, 슬라이스 헤더 정보, 프레임 메타 정보_중 적어도 하나를 포함하는 것을 특징으로 하는 데이터 파편 분류를 이용한 데이터 복원 방법.

청구항 14

제1항 내지 제13항 중 어느 한 항에 따른 방법을 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체.

발명의 설명

기술 분야

[0001] 본 발명은 데이터 파편 분류를 이용한 데이터 복원 방법에 관한 것으로, 특히 저장매체 내 삭제되거나, 손상된 데이터를 용이하게 복원할 수 있는 데이터 파편 분류를 이용한 데이터 복원 방법에 관한 것이다.

배경 기술

[0002] IT 기술이 급격하게 발전함에 따라, 다양한 분야에서 IT 기술이 융합되어 널리 사용되고 있다. 특히, 많은 사람들이 언제 어디서나 유무선 인터넷에 접속하여 디지털 기기를 이용함에 따라, 보다 많은 멀티 미디어 데이터를 저장매체에 저장하는 다양한 기술이 요구되고 있다.

[0003] 특히, 저장매체를 효과적으로 사용하기 위해서는 데이터를 체계적으로 저장, 삭제, 수정 등의 관리하는 파일 시스템이 요구된다.

[0004] 도 1은 종래기술에 따라 파일 시스템을 이용하여 디지털 기기에서 저장매체 내 데이터를 기록하는 방법을 나타낸 도면이다.

[0005] 도 1(a)에 도시된 바와 같이, 파일 시스템이 사용하는 메타 데이터 영역에는 데이터 또는 파일의 이름, 크기, 위치 등의 부가 정보가 저장되고, 실제 데이터는 저장소 영역에 저장된다.

[0006] 이처럼 파일 시스템에서는 데이터를 저장매체에 기록하기 위해 특정 크기의 단위를 사용하는데, 이를 데이터 파편(fragment)이라고 한다. 예를 들어, 하나의 파일(File 1)이 총 5개의 데이터 파편(3, 4, 5, 6, 7)으로 구성되고, 모든 파편이 연속적으로 저장되었다면, 메타 데이터 영역의 정보를 이용하여 3 → 4 → 5 → 6 → 7번 파편의 순서대로 접근하여 상기 파일 1의 내용을 확인한다.

[0007] 또는 다른 파일(File 2)이 총 3개의 데이터 파편(11, 25, 17)으로 구성되고, 모든 파편이 불연속적으로 단편화되어 저장된 경우에도 메타 데이터 영역의 정보를 이용할 수 있으며, 11 → 25 → 17번 파편의 순서로 접근하여 내용을 확인한다.

[0008] 하지만 이때, 상기 도 1(a)에 저장된 두 개의 파일(File 1, File 2)이 임의의 값으로 데이터영역을 덮어쓰는 완전 삭제인 경우를 제외하고, 일반적으로 삭제된 경우에는 도 1(b)에 도시된 바와 같이, 저장소 영역의 실제 데이터는 그대로 유지됨에 따라, 메타 데이터 영역의 속성값이 초기화되거나 변경되지 않는 경우에는 해당 정보를 탐색하여 파일을 복구한다.

[0009] 만약, 도 1(c)에 도시된 것처럼, 메타 데이터 영역이 유실되는 경우에는 파일이 저장된 파편들의 위치 정보를 확인할 수 없기 때문에 저장소영역 전체에 대해서 파일의 시작 파편을 탐지한 후, 데이터의 유효 크기를 계산하거나 내부 포맷 구조를 검증하는 방식을 이용하여 파일을 복구한다.

[0010] 하지만 이러한 파일 복구 방법은 일반적으로 삭제된 파일을 구성하는 데이터 파편 전체가 연속적으로 존재해야만 정상적인 파일 상태로 복구가 가능하기 때문에, 만약 데이터 파편이 불연속적으로 단편화된 경우에는 정상상태로의 파일 복구가 어려운 문제점이 발생했다.

[0011] 특히, 동영상 파일의 경우에는 다수의 파편 집합이 불연속적으로 조각나서 저장될 가능성이 높아 파일이 삭제되고 메타 데이터 영역이 유실되는 경우에는 복구가 어렵게 된다. 이처럼 동영상 파일의 생성과 관련하여 최근 스마트폰, 태블릿, CCTV, 블랙박스 등의 디지털 기기가 널리 사용되고, 이러한 상기 디지털 기기에서는 플래시 메모리 기반의 저장매체를 주로 사용하고 있다. 이러한 플래시 메모리의 경우에는, 저장 메커니즘의 특성상 데이터 파편들의 단편화가 매우 많이 발생할 수 있어, 데이터 파편이 불연속적으로 단편화되는 경우의 정상상태로의 파일 복구 방법이 요구되고 있다.

선행기술문헌

특허문헌

[0012] (특허문헌 0001) KR 10-2001-0064801 (전자문서 백업/복구 장치 및 그 방법, 한국전기통신공사) 2001.07.11.

발명의 내용

해결하려는 과제

[0013] 상기와 같은 종래 기술의 문제점을 해결하기 위해, 본 발명은 복원하고자 하는 데이터와 관련있는 데이터 파편을 분류하고, 분류 결과에 기초하여 데이터 파편을 재조립하여 영상 프레임을 추출함으로써, 데이터 파편이 불연속적으로 단편화되는 경우에도 용이하게 복원할 수 있는 데이터 파편 분류를 이용한 데이터 복원 방법을 제공하고자 한다.

과제의 해결 수단

[0014] 위와 같은 과제를 해결하기 위한 본 발명의 한 실시 예에 따른 데이터 파편 분류를 이용한 데이터 복원 방법은 입력부가 복원하고자 하는 복원 대상 데이터를 입력받는 단계; 파편분류부가 저장매체 내 기저장된 데이터 파편 중 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계; 파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 재조립하는 단계; 및 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계;를 포함한다.

[0015] 특히, 파편의 분류를 위한 기준 파편의 크기를 설정하는 것을 더 포함하는 입력부가 대상 데이터를 입력받는 단계를 포함할 수 있다.

[0016] 특히, 상기 저장매체의 종류에 따라 섹터, 클러스터, 블록, 페이지 중 적어도 하나에 해당하는 저장 단위인 기준 파편의 크기를 포함할 수 있다.

[0017] 보다 바람직하게는 상기 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정; 식별된 파편의 유형에 기초하여 상기 복원 대상 데이터와 관련된 파편이 저장매체 내 기저장된 데이터 파편 중에 존재하는지 검색하는 과정; 및 검색된 파편에 기초하여 상기 복원 대상 데이터에 대한 파편 맵을 생성하는 과정; 을 포함하는 파편분류부가 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계를 포함할 수 있다.

[0018] 보다 바람직하게는 상기 파편의 유형을 식별하는 과정을 수행하기 전, 해쉬 알고리즘에 기초하여 상기 복원 대상 데이터 내 중복되는 파편을 제거하는 과정;을 더 포함하는 파편분류부가 상기 복원 대상 데이터와 관련된 파편을 분류하는 단계를 포함할 수 있다.

[0019] 특히, 상기 파편의 유형이 키 프레임(Key-frame), 델타 프레임(delta-frame), 디코딩 정보, 랜덤 중 적어도 하나인 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정을 포함할 수 있다.

[0020] 보다 바람직하게는 상기 파편의 유형이 키 프레임 또는 델타 프레임인 경우, 상기 키 프레임 또는 델타 프레임의 헤더 영역에 저장된 메타 정보에 기초하여 영상 프레임의 시작 부분이 포함된 파편을 분류하는 것을 더 포함하는 복원 대상 데이터의 영상 프레임에 대한 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 식별하는 과정을 포함할 수 있다.

[0021] 보다 바람직하게는 상기 복원 대상 데이터에 대한 파편 맵을 입력받는 과정; 상기 복원 대상 데이터의 헤더 영역에 저장된 메타 정보의 유효성을 검증하는 과정; 상기 복원 대상 데이터의 영상 프레임에 대한 구조를 검증하는 과정; 및 상기 파편의 메타 정보에 대한 유효성 및 영상 프레임의 구조에 대한 검증 결과에 기초하여 영상 프레임 체인을 재조립하는 과정; 을 포함하는 파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 재조립하는 단계를 포함할 수 있다.

[0022] 보다 바람직하게는 상기 영상 프레임 체인에 대한 메타 정보가 존재하는지 검색하는 과정; 상기 영상 프레임 체인에 대한 메타 정보가 존재하는 경우, 상기 메타 정보를 획득하는 과정; 추출하고자 하는 영상 프레임의 이름을 설정하는 과정; 및 획득한 상기 메타 정보에 기초하여 이미지 파일 형태의 영상 프레임을 추출하는 과정; 을 포함하는 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프

프레임을 추출하는 단계를 포함할 수 있다.

[0023] 특히, 상기 이미지 파일을 시계열 순으로 나열하여 동영상 파일 형태의 영상 프레임을 추출하는 과정;을 더 포함하는 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계를 포함할 수 있다.

[0024] 보다 바람직하게는 상기 영상 프레임 체인에 대한 메타 정보가 존재하지 않는 경우, 상기 영상 프레임 체인의 데이터를 디코딩하여 이미지 파일 형태의 영상 프레임을 추출하는 과정; 추출된 상기 이미지 파일에 대하여 이미지 프로세싱 기법을 적용하여 적어도 하나의 이미지 집합을 분류하는 과정;을 더 포함하는 영상프레임추출부가 상기 파편들을 재조립한 결과에 기초하여 상기 복원 대상 데이터에 대한 영상 프레임을 추출하는 단계를 포함할 수 있다.

[0025] 특히, 파일 시스템의 비할당 영역, 미사용 영역, 단일 파일의 비할당 영역, 미사용 영역, 손상된 파일, 로우 덤프(raw dump) 중 적어도 하나인 복원 대상 데이터를 포함할 수 있다.

[0026] 특히, 시간, 장소, GPS 정보, 카메라 번호, 속도, 컨테이너 메타 정보, NAL 헤더 정보, 슬라이스 헤더 정보, 프레임 메타 정보(시간, 장소, GPS 정보, 카메라 번호, 속도 등) 중 적어도 하나를 포함하는 복원 대상 데이터의 메타 정보를 포함할 수 있다.

발명의 효과

[0027] 본 발명의 데이터 파편 분류를 이용한 데이터 복원 방법은 복원하고자 하는 데이터 파편과 관련있는 데이터 파편을 분류하고, 분류된 데이터 파편을 재조립하여 영상 프레임을 추출함으로써, 복원하고자 하는 데이터 파편이 불연속적으로 단편화되더라도 데이터를 용이하게 복원할 수 있는 효과가 있다.

도면의 간단한 설명

[0028] 도 1은 종래기술에 따른 디지털 기기에서 저장매체 내 데이터를 기록하는 방법을 나타낸 도면이다.
 도 2는 본 발명의 일 실시 예에 따른 데이터 파편 분류를 이용한 데이터 복원 방법을 나타낸 순서도이다.
 도 3은 본 발명의 복원 대상 데이터와 관련된 파편 분류 단계의 세부 과정을 나타낸 순서도이다.
 도 4는 연속적 및 비연속적으로 저장된 파편 분류의 예를 나타낸 도면이다.
 도 5는 MP4 컨테이너 파일에 포함된 H.264 영상프레임 구조의 예를 나타낸 도면이다.
 도 6은 데이터 파편 분류 결과를 나타낸 도면이다.
 도 7은 영상 프레임 체인 재조립 단계의 세부 과정을 나타낸 순서도이다.
 도 8은 단일 슬라이스 및 다중 슬라이스를 사용하는 영상 프레임 체인 구조를 나타낸 도면이다.
 도 9는 영상 프레임 체인의 재조립을 위한 검증 체계를 개략적으로 나타낸 도면이다.
 도 10은 영상 프레임 체인의 재조립을 위한 검증 체계 중 메타 정보 검증과정의 알고리즘을 나타낸 도면이다.
 도 11은 영상 프레임 체인의 재조립을 위한 검증 체계 중 영상 데이터 구조 검증을 나타낸 도면이다.
 도 12는 단일 슬라이스 및 다중 슬라이스를 사용하는 영상 프레임 체인 재조립 과정을 나타낸 도면이다.
 도 13은 영상프레임추출부의 파편들을 재조립한 결과에 기초하여 복원 대상 데이터에 대한 영상 프레임을 추출하는 과정의 세부과정을 나타낸 순서도이다.
 도 14는 영상 프레임과 관련된 메타 정보 관리 유형을 나타낸 도면이다.
 도 15는 이미지 파일 내 메타 정보를 저장하는 유형을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0029] 이하, 본 발명을 바람직한 실시 예와 첨부한 도면을 참고로 하여 본 발명이 속하는 기술 분야에서 통상의 지식

을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 여기에서 설명하는 실시 예에 한정되는 것은 아니다.

[0030] 이하, 도 2를 참조하여 본 발명에 따른 데이터 파편 분류를 이용한 데이터 복원 방법에 대하여 자세히 살펴보도록 한다.

[0031] 도 2는 본 발명의 일 실시 예에 따른 데이터 파편 분류를 이용한 데이터 복원 방법을 나타낸 순서도이다.

[0032] 도 2에 도시된 바와 같이, 본 발명의 데이터 파편 분류를 이용한 데이터 복원 방법은 입출력장치, 프로세서 및 저장매체 등의 하드웨어 수단을 통해 구현되는데, 먼저, 입력부가 입출력장치를 통해 복원하고자 하는 복원 대상 데이터를 입력받는다(S110). 이때, 상기 입력부가 입력받는 상기 복원 대상 데이터는 파일 시스템의 비할당 영역, 미사용 영역, 단일 파일의 비할당 영역, 미사용 영역, 손상된 파일, 로우 덤프(raw dump) 중 적어도 하나이며, 임의의 바이트 스트림 형태로 이루어질 수 있다.

[0033] 특히, 상기 파일 시스템에서는 데이터를 저장매체에 기록하기 위해 특정 크기의 단위를 사용하여 저장하는데, 이때, 사용되는 특정 크기의 단위를 데이터 파편(fragment)이라고 할 수 있다.

[0034] 또한, 상기 입력부가 데이터 파편의 분류를 위해 기준 파편의 크기 또한 설정할 수 있는데, 이때, 설정되는 상기 기준 파편의 크기는 저장매체 또는 파일 시스템의 종류에 따라 섹터(sector), 클러스터(cluster), 블록(block), 페이지(page) 중 적어도 하나에 해당하는 것으로서, 최소 저장 단위를 나타낸다. 이처럼, 데이터 파편의 크기를 최소 저장 단위로 설정함으로써, 데이터가 조각난 채로 저장되더라도 단편화가 발생한 경계를 신속하고 용이하게 파악할 수 있도록 한다.

[0035] 이에 더하여, 미리 확보된 영상 프레임의 디코딩 정보가 존재하는 경우에는 상기 디코딩 정보를 기본값으로 입력하여 복원 대상 데이터의 스트림에서 디코딩 정보를 검색하지 못하는 경우에도 데이터의 복원이 가능하도록 한다.

[0036] 이어서, 프로세서 내에서 구동되는 파편분류부가 저장매체 내 기저장된 데이터 파편 중 앞서 입력받은 복원 대상 데이터와 관련된 파편을 분류한다(S120).

[0037] 특히, 이하 파편의 유형을 식별하는 과정을 수행하기에 앞서, 해쉬 알고리즘에 기초하여 상기 복원 대상 데이터 내 중복되는 파편을 제거하는 과정을 수행할 수 있으며, 이러한 복원 대상 데이터 내 중복되는 파편을 먼저 제거하는 과정은 생략될 수도 있다.

[0038] 도 3은 본 발명의 복원 대상 데이터와 관련된 파편 분류 단계의 세부 과정을 나타낸 순서도이다.

[0039] 도 3에 도시된 바와 같이, 파편분류부가 상기 복원 대상 데이터의 영상 프레임에 대해 바이트 단위의 검색을 통해 분류하고자 하는 파편의 유형을 먼저 식별한다(S121). 이때, 식별되는 파편의 유형은 키 프레임(Key-frame), 델타 프레임(delta-frame), 디코딩 정보, 랜덤 중 적어도 하나일 수 있다.

[0040] 먼저, 상기 키 프레임은 이미지 전체를 저장하는 프레임으로서, 디코딩을 위해 다른 프레임이 요구되지 않으며, 비디오 코덱에 따라 I-frame(Intra-coded frame) 또는 IDR(Instantaneous Decoding Refresh) 프레임으로 표현될 수 있다.

[0041] 또한, 델타 프레임은 상기 키 프레임과 달리, 디코딩을 위해 자신의 이전 또는 이후에 프레임이 요구되며, 비디오 코덱에 따라 P-frame(Predicted frame), B-frame(Bi-predictive frame) 또는 non-IDR 프레임으로 나타낼 수 있다. 특히, 상기 키 프레임 또는 델타 프레임의 헤더(header)영역에는 각 프레임의 메타정보가 저장되는데, 이와 같이 저장된 각 프레임의 메타정보를 이용하여 영상 프레임의 시작 부분이 포함된 파편을 분류할 수 있다. 예를 들면, H.264 비디오 코덱의 경우에는 헤더 영역에 NAL(Network Abstraction Layer) 헤더와 Slice 헤더가 프레임의 메타정보로서 포함된다.

[0042] 이어서, 디코딩 정보는 압축 저장된 영상 프레임의 데이터를 압축 해제하여 가시적인 정보를 획득하기 위해 필수적으로 요구되는 설정 정보를 나타낸다. 이에 따라, 어떤 파편 내 디코딩 정보가 포함되어 있다면, 해당 파편을 디코딩 정보가 포함된 파편으로 분류하고, 영상의 디코딩에 활용하도록 한다. 이러한 디코딩 정보는 비디오 코덱에 따라 저장되는 형태가 다양한데, 예를 들어, H.264 비디오 코덱의 경우에는 SPS(Sequence parameter set)와 PPS(Picture parameter set) 정보로 저장된다.

[0043] 넷째, 랜덤 파편은 특정 시그니처(signature)값이 존재하지 않는 데이터 스트림이 높은 난수성(randomness)을 갖는 경우에 해당된다. 일반적으로 비디오 코덱에서는 영상 프레임을 압축 저장하므로, 데이터 스트림의 난수성

이 높은데, 키 프레임 또는 델타 프레임의 크기가 큰 경우에 있어서, 1개 이상의 랜덤 파편에 영상 데이터의 중간 영역이 저장될 수 있다.

[0044] 이와 같이 식별된 파편의 유형에 기초하여 입력받은 상기 복원 대상 데이터와 관련된 데이터 파편이 저장매체 내 기저장된 데이터 파편 중에 존재하는지 검색한다(S122).

[0045] 이하에서는 MP4 컨테이너 파일에 포함된 H.264 영상 데이터의 파편이 연속적이거나, 또는 불연속적으로 저장된 경우를 예로 들어 자세히 살펴보고도록 한다.

[0046] 도 4(a)는 연속적으로 저장된 데이터 파편 분류의 예를 나타낸 도면이다.

[0047] 도 4(a)에 도시된 바와 같이, 비디오 데이터를 저장하는 방식에 따라 일부 차이가 존재하지만, 일반적으로 디코딩 정보는 별도로 관리되며 영상 프레임이 연속적으로 존재하는 형태로 저장된다. 이와 같이 저장된 영상 프레임에 대하여 데이터 파편의 유형을 식별해 본 결과, 1번 파편에서 H.264의 디코딩 정보가 발견되었고, 2번 파편의 중간 지점에서 키 프레임의 시작 부분이 발견되었다. 또한 3번 파편은 알려진 시그니처값이 발견되지 않았고 데이터 스트림의 난수성이 높아 랜덤으로 분류되었다. 또한 4번 파편에서 델타 프레임의 시작 부분이 발견되었고, 5번 파편은 난수성이 높아 랜덤으로 분류되었다.

[0048] 이와 달리, MP4 컨테이너 파일에 포함된 H.264 영상 데이터의 파편이 불연속적으로 조각난 채로 저장된 경우를 살펴보고도록 한다.

[0049] 도 4(b)에 도시된 바와 같이, 데이터 파편의 유형을 식별해 본 결과, 1번 파편에서 델타 프레임의 시작 부분이 발견되었고, 2번 파편의 중간 지점에서 키 프레임의 시작 부분이 발견되었다. 또한 3번 파편에서 H.264의 디코딩 정보가 발견되었으며, 4번과 5번 파편은 알려진 시그니처값이 발견되지 않았고 데이터 스트림의 난수성이 높아 랜덤으로 분류되었다.

[0050] 특히, 영상 프레임의 헤더 정보를 이용하여 키 프레임과 델타 프레임의 시작 파편을 분류할 수 있다. 이때, 영상 프레임의 데이터가 저장되는 컨테이너 파일(MP4, AVI, MKV 등) 또는 비디오 코덱(H.264, MPEG-4 visual 등)에 따라 이용 가능한 헤더 정보의 종류에 차이가 존재하지만, 이러한 헤더 정보는 영상 프레임의 시작 부분을 효과적으로 탐지하기 위해 사용할 수 있으며, 뿐만 아니라, 단편화된 데이터 파편의 재조립 과정에서도 유용하게 활용될 수 있다.

[0051] 예를 들어, H.264 비디오 코덱으로 생성된 영상 데이터의 경우에는 헤더 영역에 NAL (Network Abstraction Layer) 헤더와 슬라이스 (Slice) 헤더가 포함된다.

[0052] 이때, 상기 NAL은 H.264 코덱에서 데이터를 저장하기 위해 사용하는 단위로서, 디코딩 정보와 각 프레임(또는 픽처)들이 독립적인 NAL 구조로 존재한다.

[0053] 표 1은 다양한 형태의 NAL을 나타낸 표이다.

표 1

NAL 타입	설명	NAL 헤더 값 (8비트)
1	Coded slice of a non-IDR picture	델타-프레임 0x01, 0x21, 0x41, 0x61
5	Coded slice of an IDR picture	키-프레임 0x25, 0x45, 0x65
6	Supplemental enhancement information	부가 정보 0x06
7	Sequence parameter set (SPS)	디코딩 정보 0x27, 0x47, 0x67
8	Picture parameter set(PPS)	디코딩 정보 0x28, 0x48, 0x68

[0055] 상기 표 1에 기재된 바와 같이, 키-프레임, 델타-프레임 그리고 디코딩 정보(SPS, PPS)가 기본적으로 사용된다.

[0056] 또한 슬라이스(Slice)는 하나의 NAL 구조에 저장되는 영상 데이터의 저장 단위를 나타내며, 비디오 코덱에서 픽처(Picture, 가시적인 이미지)를 구성하는 요소이다. 일반적으로 픽처와 슬라이스가 1:1로 구성되어 하나의 NAL 구조에 하나의 픽처가 포함된다. 하지만, 만약 하나의 픽처가 다수의 슬라이스로 구성되는 경우에는 하나의 NAL 구조에 저장되는 슬라이스에 픽처의 일부분만이 포함된다.

[0057] 표 2는 H.264 비디오 코덱에서 사용하는 슬라이스 타입의 값을 나타낸다.

표 2

슬라이스 타입	설명
0, 1, 3, 5, 6, 8	델타-슬라이스
2, 4, 7, 9	키-슬라이스

[0059] 상기 표 2에 기재된 바와 같이, H.264 비디오 코덱에서 사용하는 슬라이스 타입의 값은 키 슬라이스 또는 델타 슬라이스로 나누어진다. 이때, 사용되는 슬라이스 타입은 NAL 구조에 포함되는 형태이므로, 정상적인 영상 데이터라면 NAL 타입과 슬라이스 타입이 서로 일치해야 하며, 이러한 특성을 통해 데이터의 파편 분류 시, 정확성을 높일 수 있다.

[0060] 도 5는 MP4 컨테이너 파일에 포함된 H.264 영상프레임의 구조를 나타낸 도면이다.

[0061] 도 5에 도시된 바와 같이, H.264 데이터가 MP4 컨테이너 파일에 포함되는 경우에는 각 NAL 구조의 앞부분에 해당 NAL의 프레임 길이 정보가 4 바이트 길이로 존재함에 따라, 분류 및 재조립 단계에서 추가적인 메타정보로서 활용할 수 있다.

[0062] 특히, 하나의 NAL 구조가 갖는 길이와 관련하여, 데이터와 인코딩 설정에 따라 다양하지만, 현재 널리 사용되는 해상도인 320p, 480p, 720p, 1080p 등에서는 대부분 200 킬로바이트(KB) 이내이기 때문에 값의 검증 범위를 줄임으로써 분류의 정확도를 높일 수 있다.

[0063] 4 바이트의 길이 정보 이후로는 곧바로 NAL 헤더와 슬라이스 헤더가 위치한다. 표준에 규정된 문법에 따라 헤더 영역을 해석하여 각 필드의 값을 획득하고 검증한다. NAL 구조 내의 각 필드는 비트(bit) 단위로 구분되는데, 도 5에 기재된 u(count)는 count 만큼의 비트를 읽는 것을 나타내고, ue(v)는 Exp-Golomb code를 의미하며, 정해진 문법에 따라서 가변 길이의 비트를 읽는다. 이때, count는 비트 1이 나오기 전까지의 비트 0의 개수를 나타내고, 상기 count 개수만큼 비트 0을 제거하며, 상기 ue(v)는 하기의 수학적 식 1과 같이 정의될 수 있다.

[0064] [수학적 식 1]

[0065]
$$ue(v) = u(count+1) - 1$$

[0066] 슬라이스 헤더에 존재하는 매크로 블록의 시작 위치(first_mb_in_slice) 값은 픽처와 슬라이스의 관계(1:1 관계이면 0)에 의해 결정되고, 슬라이스 타입(slice_type)은 표 2를 통해 의미를 확인할 수 있다. 또한, 디코딩 정보를 통해 획득한 값(log2_max_frame_num_minus4)으로 계산되는 프레임 번호(frame_num) 필드는 현재 프레임의 번호를 나타내는데, 키 프레임은 0번이며 다음 키 프레임 이전까지 존재하는 델타 프레임들에 대해서는 순차적으로 증가하는 번호가 부여된다.

[0067] 이때, 앞서 설명한 각 필드 값이 모두 유효한 범위를 가져야하며, 슬라이스는 NAL 구조에 포함되는 형태이므로 NAL 타입이 키 프레임이면 슬라이스 타입도 키 슬라이스여야 검증을 통과할 수 있다.

[0068] 도 5(a)는 MP4 컨테이너 파일 내부에 연속적으로 저장된 H.264 영상프레임을 나타낸 도면으로서, 파편 분류 단계에서 활용되는 주요 정보를 나타낸다. 파편 분류 시, 도 5(a)와 같은 입력 파편에 대하여 바이트 단위 검색을 수행할 때 [프레임 길이][NAL 헤더][슬라이스 헤더] 영역이 정상적으로 탐지되면, 이를 영상프레임 파편으로 분류한다.

[0069] 한편 H.264 영상 데이터가 MP4 컨테이너 파일에 포함되지 않고 H.264 로우(raw) 저장 형식으로 존재하는 경우에는 각각의 NAL 구조 앞부분에 길이 정보가 아닌 syncword(0x00000001)가 위치한다. 따라서 로우 저장 형식인 경우에도 syncword와 NAL/슬라이스 헤더의 문법을 검증함으로써 효과적으로 탐지가 가능하다.

[0070] 뿐만 아니라, NAL 타입(nal_unit_type), 슬라이스 타입(slice_type), 프레임 번호(frame_num), 매크로블록의 시작 위치(first_mb_in_slice) 등의 필드는 이후의 재조립 단계에서도 활용된다.

[0071] 특히, H.264의 디코딩 정보는 SPS와 PPS NAL 구조에 저장된다. 도 5(b)는 MP4 컨테이너 파일에 포함된 H.264 디코딩 정보 영역을 상세하게 나타낸 도면으로서, SPS와 PPS NAL의 저장 구조를 확인할 수 있다. 도 5(b)에서 확인할 수 있는 것처럼 MP4 컨테이너에서 H.264 디코딩 정보를 관리하는 avcC 항목의 문법 검증과 함께 SPS 이후에 연속적으로 PPS가 존재하는 구조를 검색함으로써 매우 효과적으로 발견할 수 있다.

- [0072] 또한, SPS NAL은 1바이트의 NAL 헤더와 디코딩 정보로 구성되며, 특히 영상 프레임의 번호를 확인하기 위한 값인 $\log_2\text{max_frame_num_minus4}$ 필드가 존재한다. 또한 PPS NAL은 1바이트의 NAL 헤더와 디코딩 정보로 구성된다.
- [0073] 추가적으로 동일한 인코딩 설정으로 생성된 영상 데이터는 동일한 디코딩 정보(H.264의 경우 SPS, PPS)를 갖기 때문에 전체 분류 대상 데이터 파편에서 디코딩 정보가 발견되지 않더라도 사용자가 정의하여 디코딩에 활용할 수도 있다.
- [0074] 다시 도 3으로 돌아가서, 검색된 파편에 기초하여 상기 복원 대상 데이터에 대한 파편 맵을 생성한다(S123).
- [0075] 도 6은 저장매체 내 저장된 데이터의 파편 분류 결과를 나타낸 도면으로서, 도 6(a)는 저장매체 내 기저장된 데이터 파편 분류 결과를 나타내고, 도 6(b)는 H.264 영상 데이터 관련된 파편 분류 결과를 나타낸다. 상술한 4개의 파편 유형 즉, 키 프레임, 델타 프레임, 디코딩 정보, 랜덤과 같이 특별한 시그니처 정보가 발견되지 않은 Unknown 파편이 이후의 재조립 과정에서 핵심적으로 활용되며, 다양한 유형의 파편을 식별할수록 효율성과 정확성을 높일 수 있다.
- [0076] 다시 도 2로 돌아가서, 프로세서에서 구동되는 파편재조립부가 분류한 파편 중 불연속적으로 조각난 파편들을 재조립하여 영상 프레임 체인을 생성한다(S130).
- [0077] 도 7은 영상 프레임 체인 재조립 단계의 세부 과정을 나타낸 순서도이다.
- [0078] 도 7에 도시된 바와 같이, 파편재조립부가 파편분류부로부터 상기 복원 대상 데이터에 대해 생성한 파편 맵을 입력받는다(S131).
- [0079] 이후, 상기 복원 대상 데이터의 헤더 영역에 저장된 메타 정보의 유효성을 검증한다(S132). 이때, 상기 복원 대상 데이터의 메타 정보는 컨테이너 메타 정보, NAL 헤더 정보, 슬라이스 헤더 정보, 프레임 메타 정보 중 적어도 하나를 포함할 수 있다. 이때, 상기 프레임 메타 정보는 시간, 장소, GPS 정보, 카메라 번호, 속도 중 적어도 하나를 포함할 수 있다.
- [0080] 이어서, 상기 복원 대상 데이터의 영상 프레임에 대한 구조를 검증한다(S133).
- [0081] 상기 파편의 메타 정보에 대한 유효성 및 영상 프레임의 구조에 대한 검증 결과에 기초하여 영상 프레임 체인을 생성한다(S134). 이때, 생성되는 영상프레임 체인(chain)은 하기의 표 3과 같이 정의할 수 있다.

표 3

[0082]	<pre> 프레임 체인(1) = 키-프레임 [0] 델타-프레임 [1] 델타-프레임 [2] ... 델타-프레임 [N-1] 프레임 체인 (2) = 키-프레임 [0] 델타-프레임 [1] 델타-프레임 [2] ... 델타-프레임 [N-1] ... 프레임 체인 (M) = 키-프레임 [0] 델타-프레임 [1] 델타-프레임 [2] ... 델타-프레임 [N-1] </pre>
	<pre> * 가정 1: 영상 데이터의 평균 프레임 속도 = N 프레임/sec * 가정 2: 픽처: 슬라이스 = 1:1 (슬라이스 헤더의 매크로블록 시작 위치 값이 항상 0) * 기호 의미: (체인 번호), [프레임 번호] </pre>

- [0083] 영상 프레임 체인의 재조립하는 단계는 앞서 상술한 분류 단계를 통해 획득한 각각의 키 프레임에 대하여 가능한 모든 파편을 재조립해서 영상 프레임 체인을 구성한다. 이때, 보다 효율적으로 재조립을 수행하기 위해, 먼저 전체 파편 맵(도 6(a))에서 영상프레임과 관련이 있는 파편들만으로 별도의 파편 맵(도 6(b))을 구성하는 과정을 수행한다.
- [0084] 이하에서는 각각의 키 프레임에 대한 영상프레임 체인 재조립 과정을 설명하기에 앞서 단일 및 다중 슬라이스를 사용하는 영상프레임 체인의 개념에 대하여 간략히 설명하도록 한다.
- [0085] 도 8(a)은 단일 슬라이스를 사용하는 영상 프레임 체인 구조를 나타낸 도면이다.
- [0086] 도 8(a)에 도시된 바와 같이, 픽처와 슬라이스가 1:1의 관계이기 때문에 프레임과 슬라이스도 1:1로 대응이 되

는 형태를 갖는 것을 알 수 있다. 이처럼, 각각의 픽처가 하나의 프레임(=슬라이스)에 저장됨에 따라, 매크로 블록 시작 위치 값(first_mb_in_slice, 줄여서 first_mb로 표시)이 항상 0이 된다. 그리고 프레임 번호(frame_num)는 키-프레임 (0)으로부터 다음 키 프레임이 나오기 전까지의 델타 프레임에 대해서 순차적으로 증가하는 값(1이상)을 갖는다. 도 8(a)에서는 영상 데이터의 평균 프레임 속도가 초당 30프레임이기 때문에 델타 프레임이 평균적으로 29번까지의 번호를 부여받게 된다.

[0087] 이어서, 다중 슬라이스를 사용하는 영상 프레임 체인 구조를 살펴보도록 한다.

[0088] 도 8(b)는 다중 슬라이스를 사용하는 영상 프레임 체인 구조를 나타낸 도면이다.

[0089] 도 8(b)에 도시된 바와 같이, 픽처와 슬라이스가 1:N의 관계이므로, 프레임과 슬라이스 또한 1:N의 형태를 갖는 것을 알 수 있다. 이에 따라, 각 픽처가 여러 개의 슬라이스에 나뉘어져 저장되면서 각 슬라이스의 매크로 블록 시작 위치 값(first_mb)이 하나의 프레임 내에서 순차적으로 증가한다. 그리고 프레임 번호(frame_num)는 (0)으로부터 다음 키 프레임이 나오기 전까지의 델타 프레임에 대해서 순차적으로 증가하는 값(1이상)을 갖는데, 하나의 프레임이 여러 개의 슬라이스로 구성되기 때문에 동일한 프레임에 속한 슬라이스들은 모두 같은 프레임 번호를 부여받게 되는 특징이 있다.

[0090] 단일 슬라이스 또는 다중 슬라이스 환경에 따라서 영상 프레임 체인의 재조립 수행 과정의 복잡성이 일부 달라질 수 있지만, 상술한 주요 특징들을 활용하여 파편의 유효성을 검증할 수 있다.

[0091] 이러한 영상프레임 체인 재조립은 각각의 키 프레임을 기준으로 수행된다.

[0092] 하나의 H.264 키 프레임을 기준으로 영상 프레임 체인의 재조립을 위한 검증 체계는 도 9를 통해 개략적으로 확인할 수 있다.

[0093] 영상 프레임 체인의 재조립을 위한 검증 단계는 먼저, 메타 정보 검증을 수행하고, 이어서 영상 데이터 구조 검증을 수행한다.

[0094] 도 10은 영상 프레임 체인의 재조립을 위한 검증 체계 중 메타 정보 검증과정의 알고리즘을 나타낸 도면이다.

[0095] 도 10에 도시된 바와 같이, 먼저 본 검증과정에서는 컨테이너 메타정보, NAL 헤더 정보, 슬라이스 헤더 정보와 같은 다양한 메타정보의 유효성을 검증한다.

[0096] 이러한 첫 번째 검증과정은 메타 정보를 이용하기 때문에 각 영상프레임의 시작 부분이 존재하는 파편에 대해서만 수행하였다면, 이어지는 영상 데이터의 구조 검증과정은 모든 후보 데이터 파편에 대해서 영상 데이터 구조를 검증한다.

[0097] 도 11은 영상 프레임 체인의 재조립을 위한 검증 체계 중 영상 데이터 구조 검증을 나타낸 도면이다.

[0098] 도 11에 도시된 바와 같이, 이러한 영상 데이터 구조 검증은 압축 영상을 디코딩(decoding)하는 과정에서의 문법 체크를 통해 수행된다. 만약 유효한 데이터 파편이라면 디코더에 입력된 모든 바이트 스트림이 처리되지만, 유효하지 않는 데이터 파편이라면 중간에 오류가 발생한다.

[0099] 예를 들어, 도 9에서는 1번 파편에 존재하는 키 프레임의 시작 부분에 대해 검증을 수행한 결과, 상술한 메타 정보 검증 및 영상 데이터의 구조 검증을 통해 영상 프레임 체인의 첫 번째 파편으로 판별되었다. 또한 2, 3번 파편은 영상 데이터의 중간 부분으로 영상 데이터의 구조 검증만을 수행하여 이에 대한 유효성을 확인하였다. 한편 4번 파편에는 델타 프레임의 시작 부분이 저장되어있지만 첫 번째 메타정보 검증을 통과하지 못하여 영상 프레임 체인에 포함되지 못하였으며, 5번 파편이 유효성이 검증되어, 체인에 포함되었다. 이러한 메타 정보 검증과 영상 데이터 구조 검증의 2중 검증 체계를 통해 효과적인 검증과 영상 프레임의 재조립이 수행될 수 있다.

[0100] 도 12(a)는 단일 슬라이스를 사용하는 영상 프레임 체인 재조립 과정을 나타낸 도면이다.

[0101] 도 12(a)에 도시된 바와 같이, 데이터 파편 분류의 결과에 기초하여 키 프레임의 시작 부분인 파편 2를 기준으로 영상 프레임 체인의 재조립을 시도한다. 파편 2에 대한 메타정보 검증 결과, 프레임 번호와 매크로 블록 시작 위치가 0이었으며 전체 프레임의 길이가 30a인 것을 확인할 수 있는데, 현재 파편에 포함된 프레임의 길이가 9a이기 때문에 나머지 21a에 대한 재조립이 요구된다. 이때 재조립이 요구되는 21a는 1개 파편(16a) 이상의 길이기 때문에 다음 이어지는 파편은 랜덤 파편이 되어야 한다. 먼저 파편 3에 대해서 영상 데이터 구조 검증을 시도하였지만 중간 과정에서 오류가 발생되었으며, 다음으로 파편 4는 전체 16a 길이의 데이터에 대해서 정상 디코딩이 성공하여 검증을 통과하였다. 또한 파편 4 이후에도 키 프레임의 끝 부분 5a 길이가 재조립

이 필요한 상황이기 때문에 다음 후보 파편들에 대한 검증을 시도하여 델타 프레임 시작 위치가 5a 이후이면서 프레임 번호가 1이고 매크로블록 시작 위치가 0인 파편 1을 발견하였다. 또한, 추가적으로 파편 1에 포함된 프레임 번호가 1인 델타 프레임의 길이가 27a인데 현재 파편에 포함된 프레임의 길이가 11a이기 때문에 나머지 16a에 대한 재조립이 요구된다. 이때 재조립이 요구되는 16a는 1개 파편 이상의 길이기 때문에 다음 이어지는 파편은 랜덤 파편이 되어야 하고, 파편 3의 전체 데이터에 대해서 정상 디코딩이 성공하여 검증을 통과하였다. 이와 같이 상술한 방법을 통해 각각의 키 프레임을 기준으로 모든 델타 프레임을 재조립하여 영상 프레임 체인을 구성할 수 있다.

[0102] 도 12(b)는 다중 슬라이스를 사용하는 영상 프레임 체인 재조립 과정을 나타낸 도면이다.

[0103] 도 12(b)에 도시된 바와 같이, 프레임과 슬라이스가 1:2의 관계이기 때문에 동일한 프레임 번호이면서 매크로블록 시작 위치가 0 이상인 슬라이스가 존재할 수 있다. 이외에는 도 12(a)와 동일한 구조이기 때문에 자세한 설명은 생략한다.

[0104] 다시 도 2로 돌아가서, 프로세서 내부에서 구동되는 영상프레임추출부가 파편들을 재조립한 결과에 기초하여 복원 대상 데이터에 대한 영상 프레임을 추출한다(S140).

[0105] 도 13은 영상프레임추출부의 파편들을 재조립한 결과에 기초하여 복원 대상 데이터에 대한 영상 프레임을 추출하는 과정의 세부과정을 나타낸 순서도이다.

[0106] 도 13에 도시된 바와 같이, 먼저 상기 영상 프레임 체인에 대한 메타 정보가 존재하는지 검색한다(S141).

[0107] 만약, 상기 영상 프레임 체인에 대한 메타 정보가 존재하는 경우, 상기 메타 정보를 획득한다(S142).

[0108] 이어서, 추출하고자 하는 영상 프레임의 이름을 설정한다(S143).

[0109] 앞서 획득한 상기 메타 정보에 기초하여 이미지 파일 형태의 영상 프레임을 추출한다(S144).

[0110] 이에 더하여 추출한 이미지 파일을 시계열 순으로 나열하여 동영상 파일을 생성한다(S145).

[0111] 만약, 상기 과정 S141에서 상기 영상 프레임 체인에 대한 메타 정보가 존재하지 않는 경우, 상기 영상 프레임 체인의 데이터를 디코딩하여 이미지 파일 형태의 영상 프레임을 추출한다(S146).

[0112] 이후, 추출된 상기 이미지 파일에 대하여 이미지 프로세싱 기법을 적용하여 내용이 유사한 적어도 하나의 이미지 집합을 분류한다(S147).

[0113] 이러한 영상 프레임 추출 과정을 통해 단일 파편 또는 재조립된 영상 프레임 체인으로부터 의미있는 정보를 획득하여 편리하고 직관적인 확인이 가능하다.

[0114] 상술한 바와 같이, 영상프레임 파편이 성공적으로 재조립되면 각 영상 프레임의 내용을 가시적으로 확인할 수 있다. 하지만, 대상 데이터 스트림 내에 수천 내지 수만 개의 영상 프레임이 존재한다면, 이를 일일이 확인하는데 매우 오랜 시간이 소요될 수 있는데, 이하에서는 각각의 영상 프레임을 효과적으로 추출하는 방법에 대하여 자세히 살펴보도록 한다.

[0115] 영상 프레임을 추출하는 과정에 있어서, 가장 먼저 단일 파편 또는 재조립된 영상 프레임 체인과 관련이 있는 메타정보의 유무를 확인한다. 상기 영상 프레임과 관련된 메타정보의 예로는 카메라 번호, 촬영 시간, 촬영 장소, 기타 부가정보 등이 있으며, 영상데이터 생성 장치별로 이와 같은 메타정보를 관리하는 방식이 다양하게 존재할 수 있다.

[0116] 도 14는 영상 프레임과 관련된 메타 정보 관리 유형을 나타낸 도면이다.

[0117] 영상 프레임의 메타정보를 관리하는 방식은 크게 두 가지로 나누어지는데, 도 14(a)에 도시된 바와 같이, 저장 구조를 해석하여 메타정보를 획득할 수 있으며, 각 영상 프레임의 메타정보를 별도로 관리하는 경우도 존재하지만, MP4와 같은 컨테이너 파일에 영상 데이터가 포함되는 경우에는 메타정보를 따로 저장하지 않는 것이 일반적이다. 하지만 이를 대신하여, CCTV, 자동차 블랙박스 등과 같은 임베디드 동영상 촬영 시스템에서는 감시, 기록의 목적으로 메타 정보를 텍스트 형태로 이미지 파일 자체에 포함하기도 한다. 도 14(b)는 이미지 파일 내에 메타 정보가 포함되는 유형의 예를 나타낸 것으로서, 광학 문자 인식(OCR, Optical Character Recognition) 기술을 활용하여 이미지 파일 내의 메타정보를 추출할 수 있으며, 촬영 시간, GPS 정보, 카메라 번호, 속도 등이 정보가 주로 저장된다.

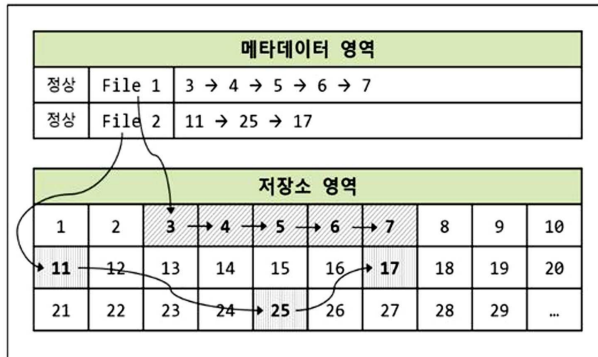
[0118] 만약, 영상 프레임에 대한 메타정보가 존재하는 경우에는 상기 메타 정보를 이용하여 해당 영상 프레임의 이름

을 결정한다. 촬영 시간, 카메라 번호 등의 형태로 영상 프레임의 이름을 결정한 후에 가시적인 이미지 파일로 추출함으로써, 영상 프레임들이 단편화되어 불연속적으로 존재하더라도 시간적으로 연속적으로 내용을 확인할 수 있도록 한다.

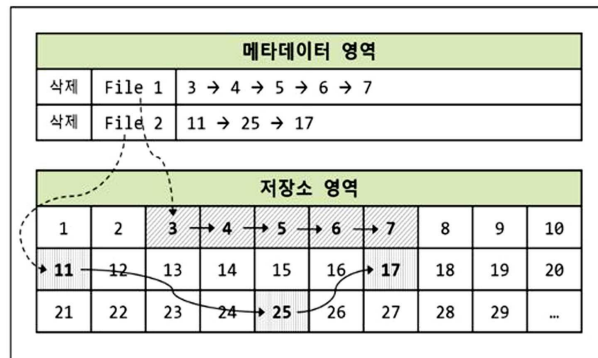
- [0119] 또한 영상 프레임의 이름 설정을 통해 시간적으로 연속적인 이미지를 획득할 수 있을 뿐만 아니라, 여러 장의 이미지를 합쳐서 하나의 재생 가능한 동영상 파일 형태로 재구성함으로써 내용 확인의 편리성과 직관성을 높일 수도 있다.
- [0120] 한편 영상프레임에 대한 메타정보가 존재하지 않는 경우에는 이름을 결정할 기준이 없기 때문에 이름 설정 없이 가시적인 이미지 파일로의 추출만을 수행한다. 추출된 여러 이미지 파일들을 육안으로 확인할 수도 있지만, 이미지 처리(image processing) 기법을 활용하여 내용이 유사한 이미지 집합으로 분류하면 보다 효율성을 높일 수 있다.
- [0121] 아울러 본 발명에서는 H.264 비디오 코덱으로 생성된 영상 데이터를 위주로 설명하였지만, 통상의 지식을 가진 자라면 MPEG-4 visual, HEVC 등 다양한 코덱으로 생성된 영상 데이터에 대해서도 본 발명에서 제안하는 분류, 재조립, 그리고 추출 단계의 기술적 사상을 벗어나지 않는 범위 내에서 적용이 가능할 것이다.
- [0122] 또한, 이러한 데이터 파편 분류를 이용한 데이터 복원 방법은 컴퓨터로 실행하기 위한 프로그램이 기록된 컴퓨터 판독가능 기록매체에 저장될 수 있다. 이때, 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, DVD±ROM, DVD-RAM, 자기 테이프, 플로피 디스크, 하드 디스크(hard disk), 광데이터 저장장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 장치에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- [0123] 본 발명의 데이터 파편 분류를 이용한 데이터 복원 방법은 복원하고자 하는 데이터 파편과 관련있는 데이터 파편을 분류하고, 분류된 데이터 파편을 재조립하여 영상 프레임을 추출함으로써, 복원하고자 하는 데이터 파편이 불연속적으로 단편화되더라도 데이터를 용이하게 복원할 수 있는 효과가 있다.
- [0124] 상기에서는 본 발명의 바람직한 실시 예에 대하여 설명하였지만, 본 발명은 이에 한정되는 것이 아니고 본 발명의 기술 사상 범위 내에서 여러 가지로 변형하여 실시하는 것이 가능하고 이 또한 첨부된 특허청구범위에 속하는 것은 당연하다.

도면

도면1



(a)

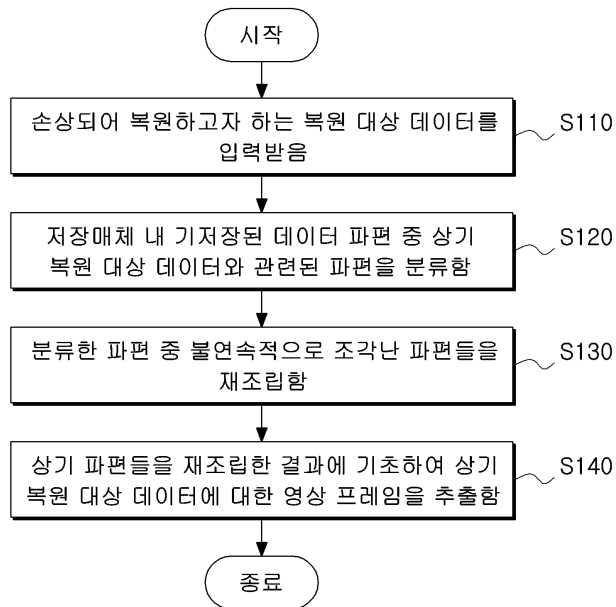


(b)

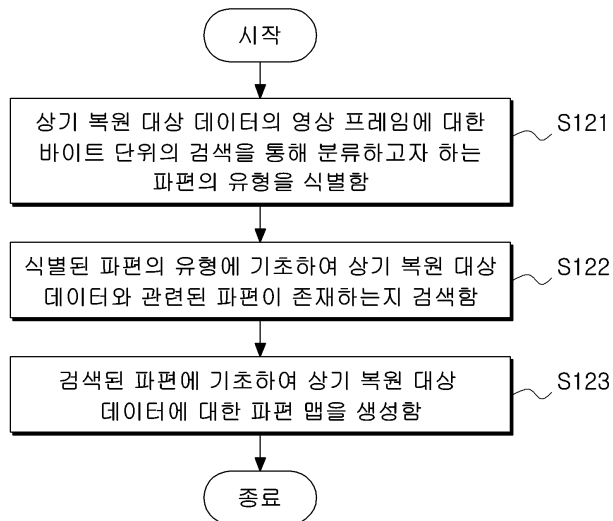


(c)

도면2



도면3



도면4

번호	1	2	3	4	5
파편					
설명	- 디코딩 정보 (SPS, PPS)	- 키-프레임 시작 (NAL, Slice 헤더)	- 랜덤 스트림	- 델타-프레임 시작 (NAL, Slice 헤더)	- 랜덤 스트림
분류 결과	디코딩 정보	키-프레임	랜덤 (random)	델타-프레임	랜덤 (random)

(a)

번호	1	2	3	4	5
파편					
설명	- 델타-프레임 시작 (NAL, Slice 헤더)	- 키-프레임 시작 (NAL, Slice 헤더)	- 디코딩 정보 (SPS, PPS)	- 랜덤 스트림	- 랜덤 스트림
분류 결과	델타-프레임	키-프레임	디코딩 정보	랜덤 (random)	랜덤 (random)

(b)

도면5

프레임 길이	NAL 헤더	슬라이스 (Slice) 헤더	인코딩된 영상 데이터 (슬라이스 데이터 == 매크로블록 집합)
4 바이트	1 바이트	가변 길이	가변 길이
00 00 58 CC	65	88 82 11 9C	9C ... ← 0x58CC (22732) → ...
00 00 10 B9	41	9A 10 29 9C	58 ... ← 0x10B9 (4281) → ...
00 00 15 A2	41	9A 20 29 9C	58 ... ← 0x15A2 (5538) → ...
⋮ ⋮ ⋮ ⋮	⋮	⋮ ⋮ ⋮ ⋮	⋮ ⋮ ⋮ ⋮
00 00 47 91	41	88 F8 29 9C	5F ... ← 0x4791 (18321) → ...
00 00 46 5A	65	88 81 02 99	99 ... ← 0x465A (18010) → ...
00 00 11 58	41	9A 10 29 9C	58 ... ← 0x1158 (4440) → ...

NAL 헤더	설명	슬라이스 (Slice) 헤더	설명
0x65 (0110 0101)	타입 = 5 (00101) 카-프레임	1 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 ue(1) = 0, ue(0001000) = 7, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 0	first_mb 0 slice_type 7 카-슬라이스 frame_num 0
0x41 (0100 0001)	타입 = 1 (00001) 델타-프레임	1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 ue(1) = 0, ue(00110) = 5, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 1	first_mb 0 slice_type 5 델타-슬라이스 frame_num 1
0x41 (0100 0001)	타입 = 1 (00001) 델타-프레임	1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 ue(1) = 0, ue(00110) = 5, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 2	first_mb 0 slice_type 5 델타-슬라이스 frame_num 2
⋮	⋮	⋮	⋮
0x41 (0100 0001)	타입 = 1 (00001) 델타-프레임	1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 ue(1) = 0, ue(0001000) = 7, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 30	first_mb 0 slice_type 7 델타-슬라이스 frame_num 30
0x65 (0110 0101)	타입 = 5 (00101) 카-프레임	1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 ue(1) = 0, ue(0001000) = 7, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 0	first_mb 0 slice_type 7 카-슬라이스 frame_num 0
0x41 (0100 0001)	타입 = 1 (00001) 델타-프레임	1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 ue(1) = 0, ue(00110) = 5, ue(1) = 0, u(log2_max_frame_num_minus4+4) = 1	first_mb 0 slice_type 5 델타-슬라이스 frame_num 1

(a)

0xAABCCDD	'avcC'	...	Sequence parameter set (SPS)	...	Picture parameter set (PPS)
4 바이트	4 바이트	...	가변 길이	...	가변 길이
'avcC' 항목 길이	항목 이름	- SPS 정보 (개수, 길이)	- NAL헤더 (1바이트) + 디코딩 정보 - 프레임 번호 확인을 위한 값 획득 (log2_max_frame_num_minus4)	- PPS 정보 (개수, 길이)	- NAL헤더 (1바이트) + 디코딩 정보

(b)

도면6

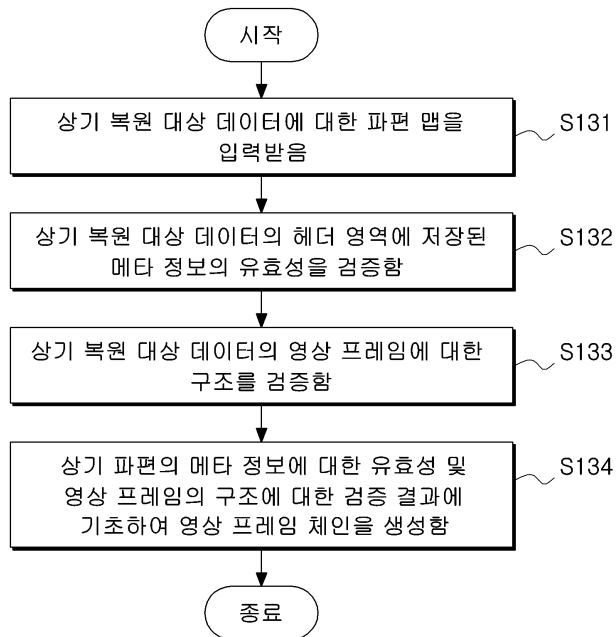
F1	sqlite
F2	h264 (카-프레임)
F3	random
F4	sqlite
F5	text (UTF-8)
F6	random
F7	h264 (델타-프레임)
F8	h264 (디코딩 정보)
F9	xml
F10	h264 (델타-프레임)
F11	random
⋮	⋮

(a)

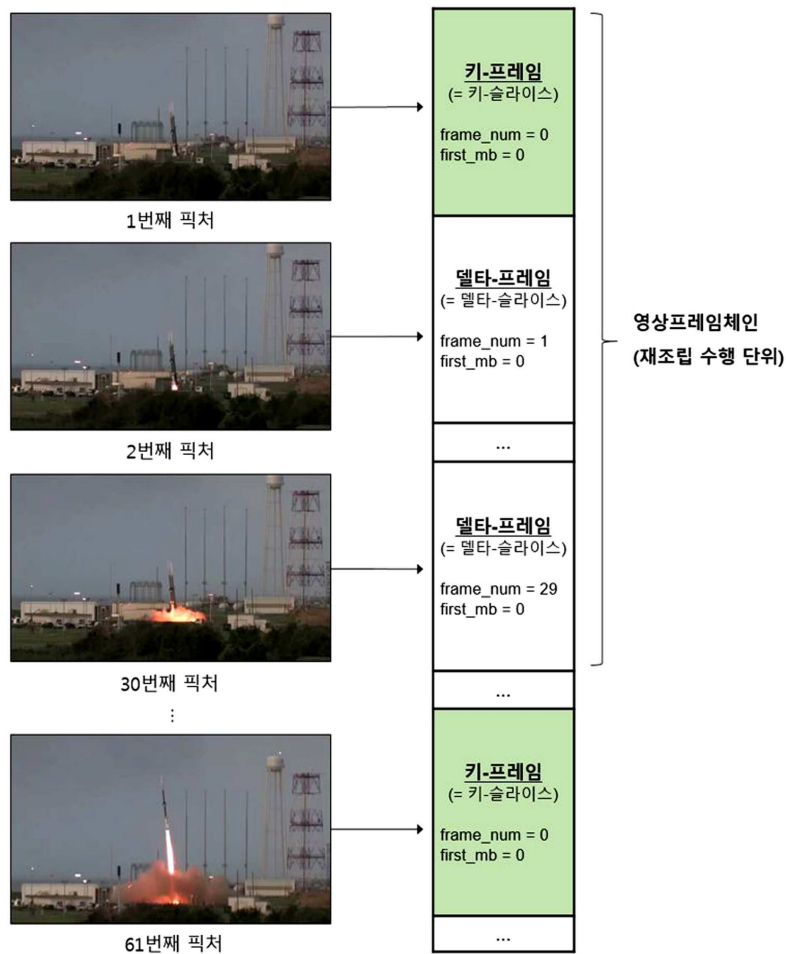
F1	sqlite
F2	h264 (카-프레임)
F3	random
F4	sqlite
F5	text (UTF-8)
F6	random
F7	h264 (델타-프레임)
F8	h264 (디코딩 정보)
F9	xml
F10	h264 (델타-프레임)
F11	random
⋮	⋮

(b)

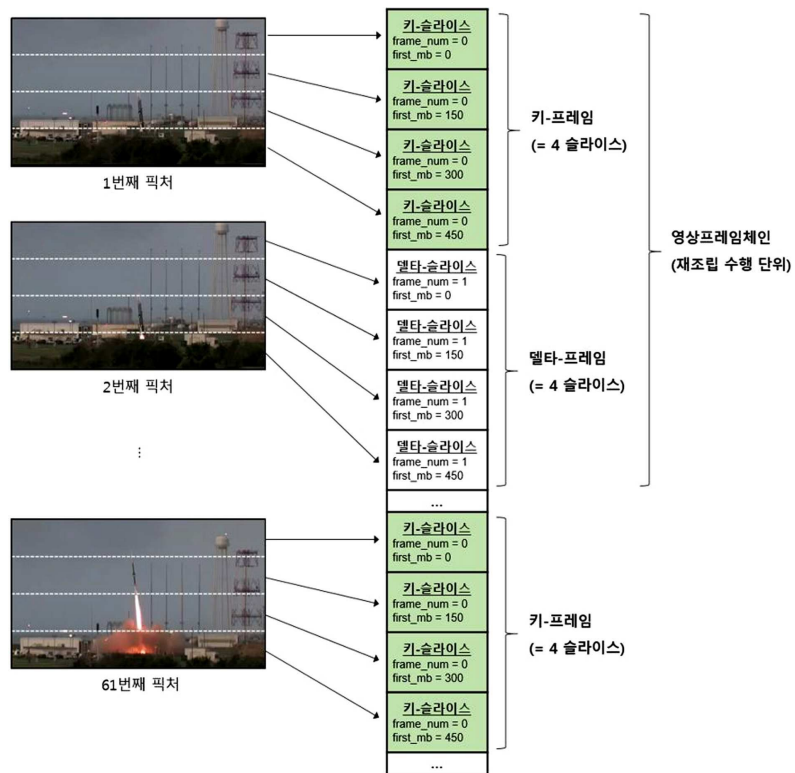
도면7



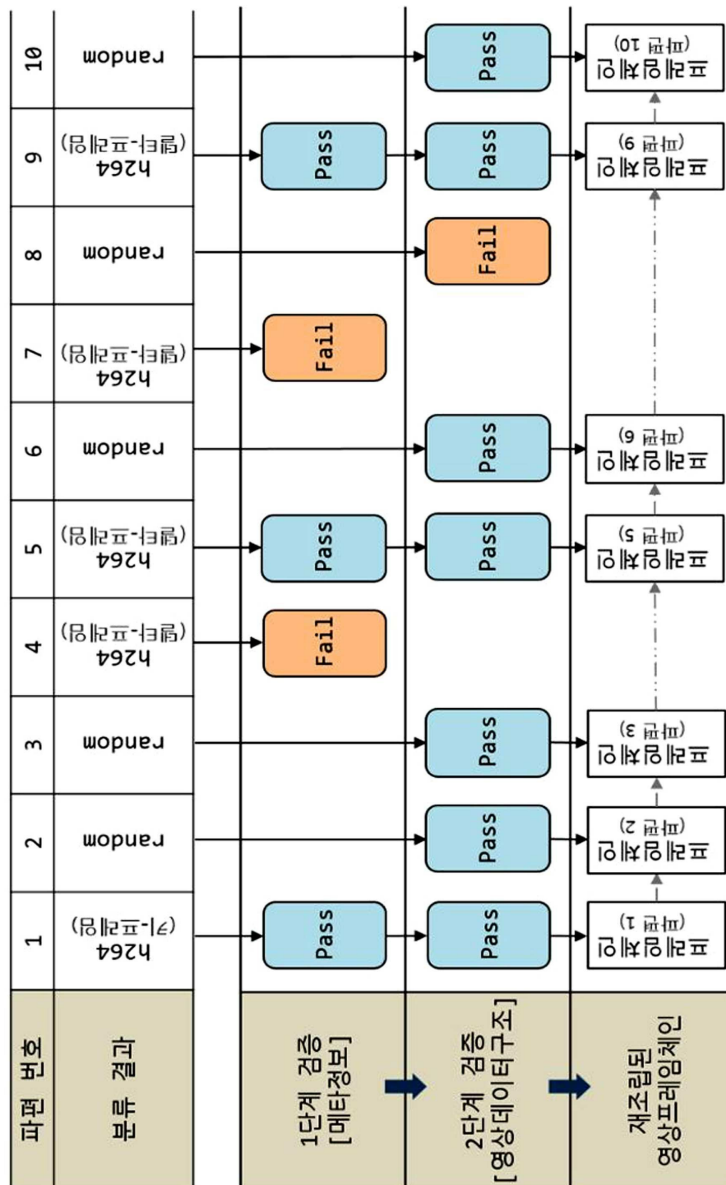
도면8a



도면8b



도면9



도면10

[1단계 검증] 영상프레임 메타정보 유효성 검증

- (1) 헤더 정보 기본 검증
 - NAL 헤더 : [표 2] 범위 참조
 - 슬라이스 헤더
 - NAL 타입과 슬라이스 타입은 동일해야 함 (키 또는 델타)
 - 프레임 번호가 0 이면, 키-프레임이어야 함
 - 프레임 번호가 1 이상이면, 델타-프레임이어야 함
- (2) 프레임 번호(frame_num)와 매크로블록 시작 위치(first_mb)의 연속성 검증
 - 연속성 검증 알고리즘 기본 개념
 - 단일 슬라이스 환경 (1픽처 = 1프레임 = 1슬라이스)
 - first_mb가 항상 0
 - 다중 슬라이스 환경 (1픽처 = 1프레임 = N슬라이스)
 - first_mb 값 사용 (0 이상의 값을 가질 수 있음)
 - 검증의 정확성을 높이기 위해 단일 슬라이스의 최소 매크로블록 개수를 *threshold* 값으로 설정 (선택적으로 적용)
 - 평균 프레임 속도가 초당 N프레임 일 때, 프레임 번호는 0 ~ (N-1) 범위에 속함
 - 연속성 검증 알고리즘 상세


```

if (first_mb == 0)
{
    // 단일/다중 슬라이스 환경 공통
    if (현재프레임번호 ≠ 이전프레임번호 + 1))      → invalid
}
else if (first_mb > 0)
{
    // 다중 슬라이스 환경 (1픽처 = 1프레임 = N슬라이스)
    if (현재프레임번호 ≠ 이전프레임번호)          → invalid
    if (이전 first_mb ≥ 현재 first_mb)              → invalid
    if ((현재 first_mb - 이전 first_mb) < threshold) → invalid
}
                    
```

도면11

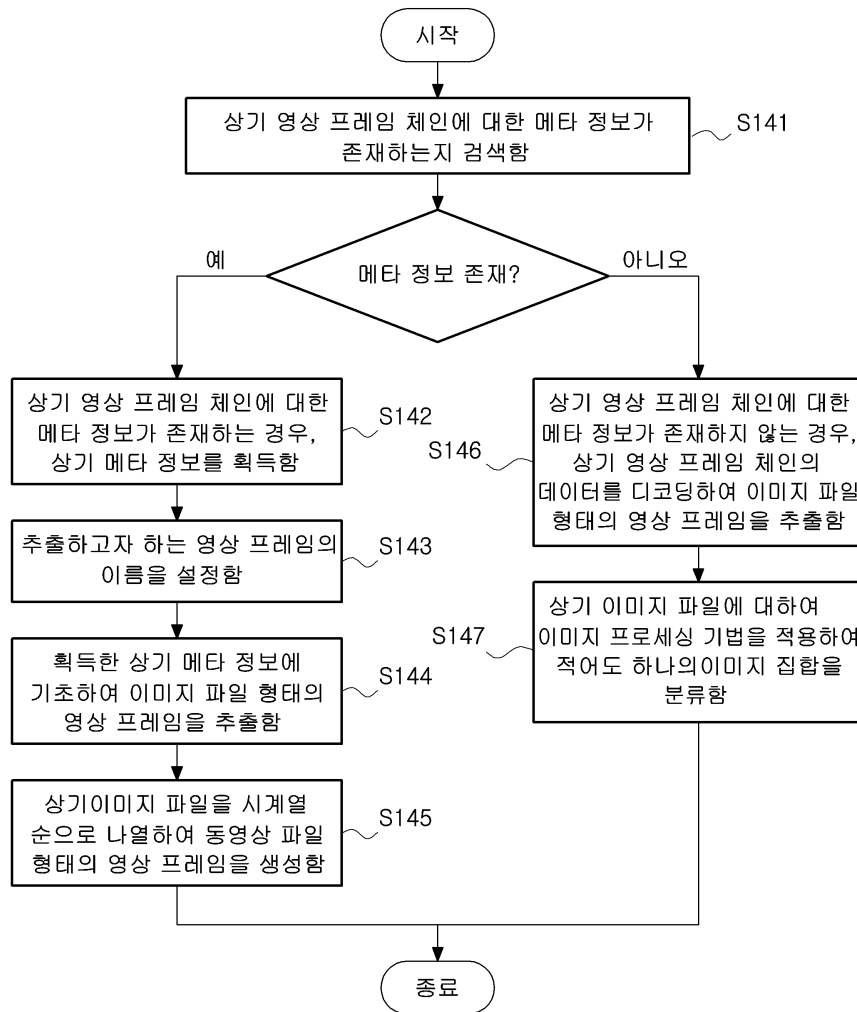
[2단계 검증] 영상 데이터 구조 검증

- 메타정보를 기반으로 모든 후보 파편들에 대해서 수행
- 문법 체크를 위해서는 정상적인 디코딩 정보가 요구됨
 - 획득 방법 1 : 영상프레임 분류 결과 (디코딩 정보 포함 파편)
 - 획득 방법 2 : 사용자 정의 입력
- 영상 데이터 검증 알고리즘 상세

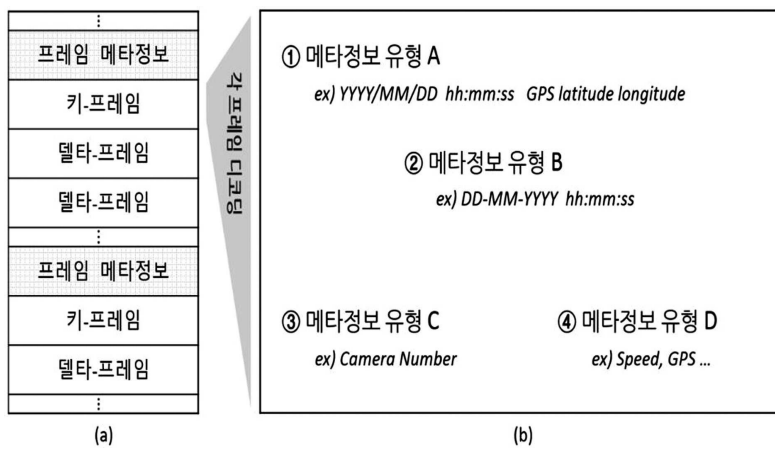

```

reft_bytes = validate_data_syntax ( 입력 데이터 스트림 )
if (reft_bytes > 0)
{
    // 디코딩 과정에서 오류가 발생한 경우
    invalid
}
else
{
    // 데이터 스트림이 정상적인 문법으로 구성된 경우
    valid
}
            
```


도면13



도면14



도면15

