



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0141529
(43) 공개일자 2013년12월26일

(51) 국제특허분류(Int. Cl.)
H04N 1/409 (2006.01) G06T 5/00 (2006.01)
(21) 출원번호 10-2013-7012596
(22) 출원일자(국제) 2011년06월08일
심사청구일자 없음
(85) 번역문제출일자 2013년05월15일
(86) 국제출원번호 PCT/EP2011/059473
(87) 국제공개번호 WO 2012/065759
국제공개일자 2012년05월24일
(30) 우선권주장
10306263.4 2010년11월16일
유럽특허청(EPO)(EP)
11305205.4 2011년02월25일
유럽특허청(EPO)(EP)

(71) 출원인
툼슨 라이선싱
프랑스 92130 이씨레폴리노 잔 다르크 뢰 1-5
(72) 발명자
더이스, 올리버
독일 30625 하노버 칼 비헤르트 알레 74 리서치
앤드 이노베이션 도이치 톼슨 오에이치쥐
뮐러, 랄프
독일 30625 하노버 칼 비헤르트 알레 74 리서치
앤드 이노베이션 도이치 톼슨 오에이치쥐
캠프헨켈, 올리버
독일 30625 하노버 칼 비헤르트 알레 74 리서치
앤드 이노베이션 도이치 톼슨 오에이치쥐
(74) 대리인
백만기, 양영준, 전경석

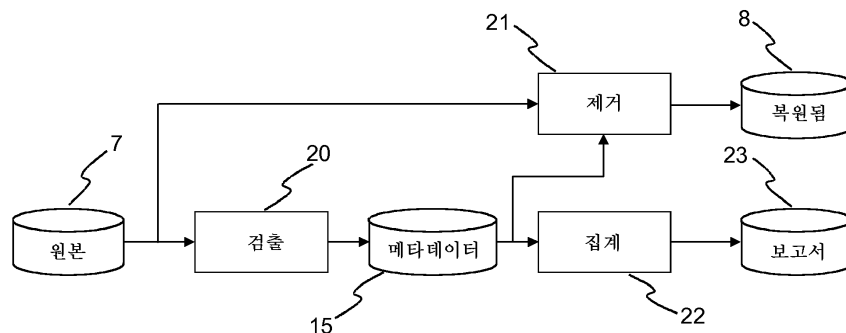
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 필름 자동 복원 방법 및 장치

(57) 요약

예를 들어, 동화상과 같은 이미지(1) 또는 이미지 시퀀스(1)의 자동 복원 방법 및 장치가 개시되고, 이는 이미지(1) 또는 이미지 시퀀스(1)의 내에서 아티팩트(2, 4, 50, 51)의 제거를 가능하게 한다. 복원은 검출 절차(20)와 제거 절차(21)로 나뉜다. 검출 절차(20)에서, 이미지(1) 또는 이미지 시퀀스(1)에서의 아티팩트(2, 4, 50, 51)가 검출되고 검출된 아티팩트(2, 4, 50, 51)에 대한 정보가 메타데이터 데이터베이스(15)에 저장된다. 후속의 제거 절차(21)에서, 메타데이터 데이터베이스(15)에 저장된 정보에 기초하여 하나 이상의 검출된 아티팩트(2, 4, 50)가 이미지(1) 또는 이미지 시퀀스(1)로부터 제거된다.

대표도 - 도9



특허청구의 범위

청구항 1

이미지(1) 또는 이미지 시퀀스(1)에서 아티팩트들(2, 4, 50, 51)을 검출하는 단계(20);

메타데이터 데이터베이스(15)에 검출된 상기 아티팩트들(artifacts; 2, 4, 50, 51)에 대한 정보를 저장하는 단계; 및

상기 메타데이터 데이터베이스(15)에 저장된 정보에 기초하여 하나 이상의 검출된 상기 아티팩트(2, 4, 50)를 상기 이미지(1) 또는 상기 이미지 시퀀스(1)에서 제거하는 단계(21)

를 포함하는 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 2

제1항에 있어서,

상기 아티팩트들(2, 4, 50, 51)은 스크래치와 오염물 객체들(scratch and dirt objects) 중 적어도 하나인, 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 3

제1항 또는 제2항에 있어서,

아티팩트(2, 4, 50, 51)에 대한 상기 정보는 상기 아티팩트(2, 4, 50, 51)의 인덱스, 상기 아티팩트(2, 4, 50, 51)의 유형, 직사각형 경계 상자(rectangular bounding box)의 좌표, 상기 아티팩트(2, 4, 50, 51)의 출현 프레임 인덱스(a frame index of the appearance), 경계 상자 크기의 바이너리 픽셀 마스크(binary pixel mask of bounding box size), 변질된 픽셀(affected pixels)의 수, 검출 가중치, 제거 플래그(removal flag), 제거 토글 플래그(removal toggle flag) 및 제거 정보 중 하나 이상을 포함하는, 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서,

아티팩트(2, 4, 50, 51)에 대한 상기 정보는 이전의 이미지(1)에 있어서 동일한 아티팩트(2, 4, 50, 51)의 인덱스와 이후의 이미지(1)에 있어서 동일한 아티팩트(2, 4, 50, 51)의 인덱스 중 적어도 하나를 더 포함하는, 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 메타데이터 데이터베이스(15)에 상기 검출된 아티팩트들(2, 4, 50, 51)에 대한 정보를 저장한 이후 그리고 하나 이상의 상기 검출된 아티팩트(2, 4, 50)를 상기 이미지(1) 또는 상기 이미지 시퀀스(1)에서 제거(21)하기 이전에 상기 메타데이터 데이터베이스(15)에 저장된 상기 정보로 품질 제어 절차(24)를 수행하는 단계

를 더 포함하는 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 6

제5항에 있어서,

상기 품질 제어 절차(24)는 상기 메타데이터 데이터베이스(15)에 아티팩트들(2, 4, 50, 51)을 추가하는 것 및/또는 상기 메타데이터 데이터베이스(15)로부터 아티팩트들(2, 4, 50, 51)을 제거하는 것을 포함하는, 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 7

제6항에 있어서,

아티팩트(2, 4, 50, 51)가 상기 메타데이터베이스(15)에서의 상기 아티팩트(2, 4, 50, 51)에 대한 상기 정보를 삭제함으로써 또는 상기 제거 토크 플래그의 값을 설정하여 상기 아티팩트(2, 4, 50, 51)를 제거를 위한 선택에서 해제함으로써 상기 메타데이터 데이터베이스(15)로부터 제거되는, 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 8

제1항 내지 제7항 중 어느 한 항에 있어서,

이미지(1) 내에서 검출된 상기 아티팩트들(2, 4, 50, 51)의 전체 영역과 상기 이미지(1) 내에서 제거를 위하여 설정된 상기 아티팩트들(2, 4, 50, 51)의 전체 영역 중 적어도 하나를 결정하는 단계

를 더 포함하는 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 9

제1항 내지 제8항 중 어느 한 항에 있어서,

상기 메타데이터 데이터베이스(15)에 저장된 상기 정보에 기초하여 품질 보고서(23)를 생성하는 단계(22)

를 더 포함하는 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 방법.

청구항 10

이미지(1) 또는 이미지 시퀀스(1)에서 아티팩트들(2, 4, 50, 51)을 검출(20)하는 제1 이미지 프로세서;

검출된 상기 아티팩트들(2, 4, 50, 51)에 대한 정보를 저장하는 메타데이터 데이터베이스(15); 및

상기 메타데이터 데이터베이스(15)에 저장된 상기 정보에 기초하여 하나 이상의 검출된 상기 아티팩트(2, 4, 50)를 상기 이미지(1) 또는 상기 이미지 시퀀스(1)에서 제거(21)하는 제2 이미지 프로세서

를 포함하는 이미지(1) 또는 이미지 시퀀스(1)를 복원하는 장치.

명세서

기술분야

[0001] 본 발명은, 예를 들어 동화상 필름(motion picture films)과 같은 이미지 또는 이미지 시퀀스(image sequences)의 자동 복원 방법 및 장치에 관한 것으로, 더욱 상세하게는 자동 복원 시 이미지 내의 스크래치(scratch) 및 오염물(dirt)을 제거하는 방법 및 장치에 관한 것이다.

배경기술

[0002] 동화상 필름은 우리 문화 유산의 일부이다. 불행하게도, 그것은 스크래치, 먼지, 오염물, 얼룩, 마모(abrasion)와 같이 바람직하지 않은 객체(objects)에 의해 종종 변질된다. 이러한 것들은 보통 필름 장면(film footage)을 현상(developing), 처리(handling), 저장(storing), 스크리닝(screening) 또는 스캐닝(scanning)하는 기술적인 절차에서 비롯된다. 몇몇 드문 경우에 필름에 기록(capturing)할 때, 예를 들어 렌즈 내의 털(fluff) 또는 스캐너 유리 상의 먼지와 같은 정적 객체가 이미 생겼을 수 있다. 이하, 이러한 모든 결점은 '스크래치 및 오염물(scratch and dirt)'로 간단히 언급될 것이다.

[0003] 스크래치 및 오염물은 필름 감상에 있어서 그 품질과 즐거움을 떨어뜨린다. 이는 대부분의 경우 영상 정보의 상당한 양이 손실되어서 그런 것은 아니다. 실제로는 그와 정반대로서, 스크래치 및 오염물은 전부 관련 없긴 하지만 필요한 양 이상의 정보를 제공한다. 인간의 뇌는 매일의 시각적 경험에 기초하여 없어진 관련 정보를 대체함에 있어서 매우 효율적이다. 하지만 이 세상에서 물질적인 것들이 보통 어떻게 보이고 행동하는지에 대한 경험은 변질된 장면(affected footage)을 볼 때 끊임없이 교란된다. 예를 들어, 스캐닝 또는 스크리닝 시 필름 상에 놓인 오염물이 한 프레임에서 나타났다가 다른 프레임에서는 사라진다. 이것은 보통 부드럽게 나타나고, 움직이고, 사라지는 객체에 대한 인간의 경험을 부정하고, 뇌는 상기 비정상적인 시각적 자극을 예외적으로 다루게 된다. 그러므로, 이러한 방법으로 관련된 정보와 관련없는 정보를 구분하는 것은 시청자에게 스트레

스를 주고 피곤하게 만든다. 스크래치 및 오염물의 제거는 시청자들을 편안하게 유지시키고 콘텐츠에 계속 집중시키기 위해 요구되며, 그러므로 이는 임의의 복원 절차에서 중요한 부분이 된다.

[0004] 가급적이면, 복원은 스캐닝 이후에 디지털 방식으로 수행된다. 비록 스캐닝 이전에 오염물과 잠재적인 스크래치를 줄이기 위해 화학적 세척 절차로 재료를 투입하는 습식 스캐닝 기술(wet scanning techniques)이 있긴 하지만, 이는 비용이 매우 비싸지도록 하는 많은 기술적인 문제들로 인해 널리 사용되지 않는다. 또한, 이미 많은 장면이 스캐닝되어 테이프 상에 디지털로 보관되어 있다.

[0005] 비록 상기 과제의 다양한 측면에서 예술가들을 돕는 소프트웨어가 시중에 나와 있긴 하지만, 각각의 스크래치 및 오염물 객체를 찾아서 제거함으로써 디지털화된 필름을 수동 복원하는 것은 명백히 시간 소모가 큰 일이다. 특히, 많은 양의 스크래치나 오염물이 포함된 오래된 콘텐츠의 수동 복원은 재정적으로 불가능할 수 있다. 이는 헤아릴 수 없는 상업적 가치를 갖는 장면을 포함하는 많은 고기록(archives)의 경우에도 마찬가지이다.

[0006] 스크래치 및 오염물을 검출하고 제거하려고 시도하는 알고리즘을 이용한 자동 복원 소프트웨어의 응용은 단지 수동 절차에 대한 실행 가능한 대안이 될 뿐이다. 현재, 거의 자동으로 스크래치 및 오염물의 검출 및 제거를 수행하는 다수의 소프트웨어 및 하드웨어 제품을 시장에서 구할 수 있다. 일반적으로, 간혹 각각의 장면에 대하여 개별적으로 검출 및 제거를 미세 조정하기 위해 특정 파라미터의 수동 조정이 요구된다. 처리과정 이후에, 복원된 출력 또는 그 부분들은 상이한 파라미터를 이용하여 복원을 재실행할 지에 관한 선택에 따라 사용되거나 거부되어야 한다. 이것은 불만족스러울 수 있는데, 그러한 조정은 시간이 걸리고 특별히 적응되지 않은 결정적인 장면에서 품질이 충분하지 않을 수 있기 때문이다. 스크래치 및 오염물의 검출은 현재 완전히 풀리지 않은 중요한 문제이다. 미검출 또는 오검출될 특정 비율의 객체는 여전히 존재한다.

발명의 내용

[0007] 본 발명의 목적은 효율적인 방법으로 필름 자동 복원 알고리즘의 결과를 제어하고 향상 가능하게 하는 이미지 복원을 위한 해결책을 제공하는 것이다.

[0008] 본 발명의 일 측면에 따르면, 상기 목적은

[0009] - 이미지 또는 이미지 시퀀스에서 아티팩트(artifacts)를 검출하는 단계;

[0010] - 메타데이터 데이터베이스(metadata database)에 상기 검출된 아티팩트에 대한 정보를 저장하는 단계; 및

[0011] - 상기 메타데이터 데이터베이스에 저장된 정보에 기초하여 상기 이미지 또는 상기 이미지 시퀀스로부터 하나 이상의 상기 검출된 아티팩트를 제거하는 단계를 포함하는 이미지 또는 이미지 시퀀스를 복원하는 방법에 의해 달성된다.

[0012] 마찬가지로, 이미지 또는 이미지 시퀀스를 복원하는 장치는

[0013] - 상기 이미지 또는 상기 이미지 시퀀스에서 아티팩트를 검출하는 제1 이미지 프로세서;

[0014] - 상기 검출된 아티팩트에 대한 정보를 저장하는 메타데이터 데이터 베이스; 및

[0015] - 상기 메타데이터 데이터베이스에 저장된 정보에 기초하여 상기 검출된 하나 이상의 아티팩트를 상기 이미지 또는 상기 이미지 시퀀스에서 제거하는 제2 이미지 프로세서를 포함한다.

[0016] 자동 메타데이터 구동형 워크플로우(automatic metadata driven workflow)를 사용하여 상기 복원 절차를 스크래치 및 오염물 객체와 같은 객체의 검출과 제거로 구분하는 본 발명에 따른 상기 해결책은 다수의 이점을 갖는다. 아티팩트를 제거하기 위한 최신의 워크플로우와 대조적으로, 이는 다양한 콘텐츠에 대하여 파라미터를 개별적으로 조정할 필요가 없다. 동시에, 상기 복원 결과는 완전히 재현가능하고 반복으로 인한 불필요한 처리과정의 수고로움이 없어도 된다. 더 나아가, 상기 객체 검출 절차 중에 생성된 상기 메타데이터는 상기 메타데이터 데이터 베이스에 저장된 정보에 기초하여 종합적인 품질 보고서를 생성할 수 있도록 한다. 그러므로, 본 발명에 따른 상기 해결책은 많은 필름 고문서들의 복원에 적절하다. 상기 검출 절차 중, 각각의 아티팩트에 대한 입력(entries)이 있다면 메타데이터 데이터베이스는 자동으로 생성된다.

[0017] 메타데이터 베이스에 검출된 아티팩트에 대한 정보를 저장한 이후 및 하나 이상의 상기 검출된 아티팩트를 이미지 또는 이미지 시퀀스에서 제거하기 이전에 상기 메타데이터 데이터베이스에 저장된 정보 상에서 품질 제어 절차가 수행되는 것이 편리하다.

[0018] 파라미터 수동 조정이 필요없는 스크래치 및 오염물 자동 복원 알고리즘을 사용하는 경우, 기타 다른 이미지 영

역이 완전히 본래 그대로 남아있는 것에 비하여 객체의 수는 상당히 줄어들게 된다. 비록 완벽하지는 않지만, 이 알고리즘은 많은 경우에 사용자 상호작용(user interaction) 없이도 충분한 결과를 달성할 수 있도록 한다. 하지만, 상기 메타데이터 데이터베이스가 상기 검출 절차 중 자동으로 생성되므로, 완벽하지 않은 알고리즘 성능으로 인해 상기 메타데이터는 부정확하거나 불완전할 수 있다. 그러므로, 완벽한 복원이 요구되는 결정적인 상황에서는, 검출 이후 및 제거 이전에 각각의 객체에 대한 상기 메타데이터의 그래픽 검토(graphical review)를 통해 결과를 개선할 수 있다. 이는 메타데이터 구동형 3 단계의 복원 워크플로우에 의해 달성된다.

[0019] 검출 절차 및 제거 절차로의 복원 절차 분리는 품질 검토 시 메타데이터 데이터베이스 내에서 입력, 예를 들어 객체를 수정하고, 추가하고, 제거하는 것을 가능하게 한다. 상기 품질 검토는 오로지 상기 메타데이터 베이스 상에서만 일어나고 개별적인 프레임 또는 프레임 시퀀스에 기초한다. 아티팩트는 상기 메타데이터 데이터베이스에서 상기 아티팩트에 대한 정보를 삭제함으로써 또는 제거 토글 플래그(removal toggle flag)의 값을 설정하여 제거할 아티팩트를 선택 해제함으로써 상기 메타데이터 데이터베이스에서 알맞게 제거된다.

[0020] 아티팩트에 대한 정보는 상기 아티팩트에 관한 하나 이상의 인덱스, 상기 아티팩트의 유형, 직사각형인 경계 상자(bounding box)의 좌표, 상기 아티팩트가 출현하는 프레임 인덱스, 경계 상자 크기에 대한 바이너리 픽셀 마스크(binary pixel mask), 변질된 픽셀 수, 검출 가중치(detection weight), 제거 플래그(removal flag), 제거 토글 플래그 및 제거 정보를 포함하는 것이 편리하다. 이러한 정보는 종합적인 품질 제어 절차를 가능하게 한다. 그 외에, 상기 정보는 검출된 결함을 그래픽 사용자 인터페이스(graphical user interface)에서 디스플레이 하기에 적절하다.

[0021] 가급적이면, 아티팩트에 대한 상기 정보는 이전 이미지(previous image)에서의 동일한 아티팩트의 인덱스와 이후 이미지(subsequent image)에서의 동일한 아티팩트의 인덱스 중 적어도 하나를 포함한다. 스크래치 또는 정적 오염물과 같은 객체는 보통 다수의 프레임에 걸쳐 퍼져있다. 그러므로, 상기 프레임 이전 및 이후에서의 동일한 객체의 인덱스를 저장하는 것은 유용하다.

[0022] 품질 검토를 위해, 사람이 직접 (그래픽으로) 스크래치 및 오염물 객체에 주석을 붙이고, 검출된 스크래치 및 오염물 객체를 개별적으로 제거하며, 스크래치 및 오염물 객체를 추가할 수 있도록 하는 그래픽 사용자 인터페이스가 알맞게 제공된다. 가급적이면, 상기 그래픽 사용자 인터페이스는 더 나아가 전문적인 그래픽 편집기를 사용하여 검출된 아티팩트를 수동으로 수정할 수 있도록 한다.

[0023] 이미지 내에서 검출된 아티팩트들의 전체 영역과 이미지 내에서 제거되는 아티팩트 세트들의 전체 영역 중 적어도 하나가 알맞게 결정된다. 이 값들은 통계의 목적뿐만 아니라, 상기 메타데이터의 그래픽 검토에도 도움이 된다.

도면의 간단한 설명

[0024] 더 나은 이해를 위해 본 발명은 도면들을 참고로 하여 이하의 설명에서 더 구체적으로 설명될 것이다. 본 발명은 이러한 예시적인 실시예로 제한되지 않으며 구체적인 특징들 또한 첨부된 청구항에서 정의된 바와 같은 본 발명의 범위를 벗어나지 않고서 편의상 조합될 수 있고 그리고/또는 수정될 수 있음을 이해할 것이다. 도면들에서,

- 도 1은 스크래치 및 오염물 검출의 문제점을 도시하고,
- 도 2는 두 개의 스크래치를 갖는 프레임을 묘사하며,
- 도 3은 스크래치의 제거 이후 도 2의 프레임을 도시하고,
- 도 4는 오염물 객체 및 움직이는 객체를 갖는 프레임 시퀀스를 묘사하며,
- 도 5는 상기 오염물 객체의 제거 이후 도 4의 프레임 시퀀스를 도시하고,
- 도 6은 필름 복원에 대한 단일 단계 자동 절차를 도시하며,
- 도 7은 운영자 보조 프레임(operator-assisted frame) 기반 복원 절차를 도시하고,
- 도 8은 운영자 보조 시퀀스 기반 복원 절차를 도시하며,
- 도 9는 본 발명에 따른 분리된 복원 절차를 묘사하고,
- 도 10은 필름 복원에 대한 더 정제된 3 단계 워크플로우를 도시하며,

도 11은 스크래치 및 오염물의 다단계 검출 전략을 개략적으로 도시하고,

도 12는 객체와 비객체(non-object)의 확률 밀도 함수(probability density functions) 및 그것들의 검출 문턱값과 제거 문턱값과의 관계를 예시적으로 도시하며,

도 13은 스크래치 또는 오염물 객체의 예시적인 경계 상자를 묘사하고,

도 14는 메타데이터베이스의 예시적인 그래픽 표현을 도시하며,

도 16은 어떠한 복원도 이루어지지 않은 원본 이미지를 묘사하며,

도 17은 검출 절차 뒤에 메타데이터 주석을 포함한 도 16의 이미지를 도시하고,

도 18은 제거 절차의 적용 이후 출력 이미지를 묘사한다.

발명을 실시하기 위한 구체적인 내용

- [0025] 사용자에게 의한 파라미터 조정 없이 디지털화된 필름 장면 상의 스크래치 및 오염물 객체를 완전히 자동 검출하는 것은 노이즈, 미세입자(grain), 플리커(flicker), 지터(jitter), 모션(motion) 및 콘텐츠 그 자체와 같이 다수의 변화하는 요인들로 인해 도전적인 과제이다. 상기 검출 알고리즘의 설계는 일반적으로 검출 문턱값 주위의 가능성을 구분함으로써 오검출률 및 미검출률을 줄이는 데에 주안을 둔다. 이는 도 1에 개략적으로 도시되어 있다. 현실에 존재하는 알고리즘에 대하여, 객체의 확률 밀도 함수는 보통 기타 다른 프레임 콘텐츠의 확률 밀도 함수와 중첩된다. 이것은 상기 객체의 오검출률 및 미검출률이 항상 0보다 클 것이라는 것을 의미한다. 그 이유는 이하에서 설명될 것이다.
- [0026] 도 2 및 도 3은 스크래치의 검출 및 제거 문제를 도시한다. 도 2에서, 두 개의 스크래치(2, 2')를 갖는 프레임(1)이 도시된다. 도 3은 상기 스크래치의 제거 이후 동일한 프레임(1)을 도시한다.
- [0027] 스크래치는 처리장치(processors), 영상기(projectors) 또는 스캐너를 통해 전송되는 동안 필름의 표면에 계속하여 손상을 주는 작은 입자들에 기인한다. 이것들은 각각의 프레임에 걸쳐 돌아다니는 경향이 있다. 만약 돌아다니는 속도가 느리다면, 이것들은 거의 동일한 위치에서 다수의 프레임에 수직에 가깝게 나타난다. 하지만, 만약 돌아다니는 속도가 빠르다면, 이것들은 상당히 큰 수평 요소로 나타나고 심지어 일 프레임에서 나타났다가 사라질 수도 있다. 수직 스크래치에 따른 문제점은 이것이 도 2에서 빌딩(3)으로 나타난 것과 같은 수직적인 이미지 특징과 혼동되기 쉬울 수 있다는 것이다. 이것은 종종 빌딩, 다리 및 울타리와 같이 인공적인 객체들의 경우에 해당한다.
- [0028] 거의 동일한 위치에서 다수의 프레임에 나타나는 스크래치는 대체로 상기 스크래치 밑의 콘텐츠에 대한 시간 정보(temporal information)를 크게 손상시킨다. 그러므로, 제거는 스크래치의 주변 영역의 공간 정보를 이용하여 여야 한다.
- [0029] 도 4 및 도 5는 오염물의 검출 및 제거에 관한 문제점을 도시한다. 도 4에서, 오염물 객체(4) 및 움직이는 객체(5)를 갖는 프레임 시퀀스(1)가 도시된다. 도 5는 오염물 객체(4)의 제거 이후 동일한 프레임 시퀀스(1)를 도시한다.
- [0030] 만약 정적이기 않다면, 필름 표면 상에 놓인 오염물은 일 프레임에서 나타나고 그 다음 프레임에서 사라진다. 로컬 이웃(local neighborhood)을 살펴보면, 움직이는 객체에 대하여도 마찬가지이다. 도 4에서의 프레임 시퀀스(1)는 프레임(n)에서의 이웃 윈도우(neighborhood window)(6) 내에서 왼쪽으로부터 오른쪽으로 움직이고 있는 움직이는 객체(5), 즉 공을 도시한다. 동일한 윈도우 위치에서의 프레임(n+1 및 n-1)에는 움직이는 객체(5)가 발견되지 않는다. 따라서, 주요 문제점은 모션으로부터 오염물을 구별하는 것이다. 이를 위해 모션이 참작되어야 한다.
- [0031] 또 다른 문제는 오염물과 동일한 시간 특성을 갖는 불규칙한 모션, 모션 블러(motion blur), 폐색(occlusion), 노출(uncovering) 및 기타 다른 효과들에 의해 야기되는 비정상적인 모션을 통해 발생한다.
- [0032] 오염물 객체(4)는 많은 경우에 프레임마다 현저하게 변경되지 않는 콘텐츠 밀부분을 복제함으로써 쉽게 제거될 수 있다. (비정상적인) 모션의 경우에 상황이 더 어려워지는데, 이는 오염물 객체(4) 밑의 원본 콘텐츠의 위치가 현저하게 변경되 때문이다.
- [0033] 도 6에서, 원본 콘텐츠(7)로부터 복원된 콘텐츠(8)를 생성하는 단일 단계 자동 복원 절차(10)가 묘사된다. 일 단 스크래치 및 오염물 자동 복원 알고리즘이 장면의 임의의 광경(scene) 및 유형에서 동등하게 잘 동작하면 그

러한 단일 단계의 자율적인 복원 절차(unsupervised restoration process)(10)는 많은 경우에 충분할 수 있다. 하지만, 아직 이용가능한 완벽한 알고리즘은 없다. 주요 이슈는 다양한 유형의 장면에 대한 알고리즘 품질을 제어하는 것이다.

[0034] 이 문제에 대한 공통 해결책이 도 7에 도시되어 있는데, 도 7에서는 운영자(9)가 복원 루프에 자리잡고 상기 결과를 프레임마다 검토(11)하고 복원 알고리즘 파라미터를 개별적으로 조정(12)한다. 완전히 상호작용하는 워크플로우인 이 반복적인 단일 단계는 시간 및 운영자의 기술에 의존하여 스크래치 및 오염물의 완벽한 제거를 가능하게 할 수도 있다. 복원 알고리즘을 제공하고 운영자가 복원된 콘텐츠(8)와 원본 콘텐츠(7)를 비교하는 데에 도움이 되는 정교한 소프트웨어 도구가 이용가능하다.

[0035] 작업 시간을 줄이기 위하여, 운영자 보조 시퀀스 기반의 복원 절차를 제시하는 도 8에 도시된 바와 같이 상기 루프는 보통 2 단계로 분리된다. 상기 복원 절차는 프레임 시퀀스, 예를 들어 장면 컷(scene cuts) 상에서 반복적으로 동작한다. 특정 파라미터 세트에 운영자(9)에 의해 복원이 초기화되고, 중간 결과(14)가 저장된다. 품질 제어 단계(13)에서, 중간 결과(14)가 수용되거나 거부되고, 조정된 파라미터 또는 경계 지어진 처리 영역을 이용해 처리과정이 재시작된다. 중간 결과(14)와 더불어, 노이즈 레벨, 미세입자 레벨, 모션 벡터 필드(motion vector fields) 및 장면 컷 위치와 같은 상기 복원 단계 중 수집된 일반적인 정보를 갖는 메타데이터 세트(15)가 저장될 수 있다. 위치 및 크기와 같은 구체적인 결점 정보가 추가적으로 저장될 수 있고 운영자에게 복원 가이드를 제공하도록 표시될 수 있다. 메타데이터의 생성은 스크리닝에 관한 다소 정성적인 절차가 정량적인 측정이 가능한 품질 절차로 변경되도록 할 수 있다. 예를 들어, 객체 수 및 프레임당 누적 면적(픽셀 총계)은 복원된 출력(8)에 덧붙여 메타데이터(15)로 저장된다. 처리과정 이후에, 완전한 메타데이터 세트의 그래픽 표현은 프레임당 얼마만큼의 픽셀들이 대체되었는지에 대한 정보를 제공한다. 전과 마찬가지로, 생성된 메타데이터(15)는 단지 결과를 수용 또는 거부하기 위한 결정을 하는 데에 도움이 된다.

[0036] 본 발명에 따르면, 최종 출력을 생성하기 이전에 복원 절차의 품질을 확인하기 위해, 도 9에 개략적으로 도시된 바와 같이 상기 복원 절차는 검출 절차(20)와 제거 절차(21)로 분리된다. 메타데이터(15)는 상기 두 절차(20, 21) 간 정보 전송을 제공한다. 검출 절차(20)에서, 상기 검출 알고리즘은 자동으로 동작하여 그 다음의 제거 절차(21) 중 사용되는 메타데이터 정보(15)를 생성한다. 상기 검출 결과는 사람 또는 기계에 의해 확인될 수 있는 종합적인 품질 보고서(23)로 집계(22)된다.

[0037] 도 10은 운영자(9)에 의해 실행되는 품질 제어(24)가 스크래치 및 오염물 객체의 검출(20)과 제거(21) 사이에 삽입된 더 정제된 3 단계 워크플로우를 묘사한다. 품질 제어(24) 중 메타데이터 데이터베이스(15)는, 예를 들어 제거가 실행되기 이전에 스크래치 또는 오염물 객체에 대한 설명을 수정하거나 추가함으로써 상호작용하며 편집된다.

[0038] 이하, 상기 3 단계 워크플로우의 상세한 설명이 주어질 것이다. 물론, 도 9의 상기 2 단계 워크플로우에서는 품질 제어에 관련된 단계를 수행하지 않는다.

[0039] 특히, 디지털화된 필름 장면 상에서 파라미터의 수동 조정이 없는 스크래치 및 오염물 객체의 자동 검출(20)은 노이즈, 미세입자, 플리커, 지터, (비정상적인) 모션과 같이 다수의 변화하는 요인으로 인해 복잡한 작업이다. 다양한 요인들에 대처하기 위해 다단계 검출 전략이 스크래치 및 오염물에 적용된다. 이 전략은 도 11에 개략적으로 도시되었다. 제1 단계(30)에서, 전체 노이즈 레벨이 추정된다. 이어서 스푸리어스 픽셀(spurious pixels)의 검출(31)이 수행된다. 상기 스푸리어스 픽셀들을 클러스터링(clustering)함으로써, 잠재적인 객체가 식별(32)된다. 식별 이후에 가중치를 주는 함수가 상기 잠재적인 객체에 적용(33)된다. 이어서 상기 객체들이, 예를 들어 가중치 임계화(weight thresholding)에 의해 검출(34)된다. 이는 검출 문턱값보다 더 큰 가중치를 갖는 이러한 잠재적인 객체들만이 메타데이터 데이터베이스(15)에 저장된다는 것을 의미한다. 상기 검출 문턱값은 미탐(false negative) 결과를 줄이기 위해 비교적 낮게 설정된다. 마지막으로, 상기 객체는 검증(35)되고, 제거를 위해 플래깅(36)된다. 상기 제거 플래그 속성은 오탐(false positive)이 제거되는 것을 방지하도록 검출 문턱값보다 더 높게 설정된 소정의 제거 문턱값 가중치에 따라 플래깅 단계(36)에서 자동으로 설정된다. 또한, 상기 제거 플래그는 검증 단계(35)에서 실행되는 객체 검증에 따라 결정된다. 스푸리어스 픽셀의 검출(31), 잠재적인 객체의 식별(32) 및 가중치를 주는 함수의 적용(33)에 사용되는 상기 알고리즘의 설계는 객체 및 비객체 확률 밀도 함수의 형태에 영향을 끼친다. 상기 설계는 오검출을 줄이기 위해 확률 밀도 함수들의 중첩 영역을 줄이는 데에 주안을 둔다. 도 12는 예시적인 객체 및 비객체 확률 밀도 함수 및 검출 문턱값과 제거 문턱값에 대한 그들의 관계를 도시한다.

[0040] 스크래치 및 오염물 객체는 하나의 프레임 내에서 유한한 영역을 덮는 특정 유형의 객체로 간주된다. 이는 노

이즈, 플리커 등과 같이 완전한 프레임에 영향을 주는 아티팩트와 대조된다. 각각의 스크래치 및 오염물 객체에 대하여, 아래의 속성 세트는 가급적이면 메타데이터 데이터베이스(15)에 기술(記述)적인 정보로 저장된다.

[0041] - 유형(스크래치 또는 오염물)

[0042] - 직사각형 경계 상자의 좌표

[0043] - 출현하는 프레임 인덱스(frame index appearance)

[0044] - 경계 상자 크기의 바이너리 픽셀 마스크

[0045] - 변질된 픽셀의 수

[0046] - 검출 가중치

[0047] - 제거 플래그(자동 설정)

[0048] - 제거 토글 플래그(사용자에 의한 설정)

[0049] - 제거 정보

[0050] 이하, 스크래치 및 오염물 객체에 대한 메타데이터의 생성을 수직, 수평 및 시간의 인덱스들을 갖는 3 차원 필드인 디지털화된 필름 자료 $y(h, v, t)$ 에 대하여 설명한다. 장면의 유형 $y(h, v, t)$ 에 따라 흑백 필름의 스칼라(scalar) 또는 컬러 필름의 3종의 RGB나 YUV 값일 수 있다.

[0051] 유일한 메타데이터 세트 $meta(t, n)$ 은 T 개의 프레임들 중 프레임 번호 t 내에서 검출된 N 개의 객체들 중 각각의 객체 n 에 대하여 정의된다. 객체는 프레임 행렬 $y(:, :, t)$ 의 특정한 유한 영역을 덮는 것으로 정의된다. 메타데이터 세트들은, 예를 들어 $metaScratch(t, n)$, $metaDirt(t, n)$ 과 같이 객체 유형으로 나누어질 수 있다. 이하, 하나의 또는 두 개의 유형을 표시하기 위해 $metaX(t, n)$ 를 사용한다. 이는 스크래치 및 오염물 객체가 동일한 번호 n 을 가질 수 있다는 것을 의미한다. 편리하게 이하의 객체 속성들이 저장된다.

[0052] a) 객체를 식별하고 구별하기 위해, 반드시 유일한 인덱스(index)가 설정되어야 한다. 프레임의 왼쪽 상단에서 시작하여 오른쪽 하단까지 상기 객체 위치에 따라 번호를 붙일 수 있다.

[0053] $metaX(t, n).index = n \{1, \dots, N\}$

[0054] b) 모든 객체 유형에 걸쳐 연속적으로 객체에 번호를 붙이는 경우에, 유형 속성의 차이에 의한 구별이 요구된다.

[0055] $metaX(t, n).type = \{'scratch'\} \{'dirt'\} \{'static\ dirt'\} \dots\}$

[0056] c) 직사각형 경계 상자의 폭과 높이는 상기 객체의 최대 수평 및 수직 범위 $xWidth(n)$ 및 $yWidth(n)$ 에 의해 정의된다. 왼쪽 상단 모퉁이 $xPos(n)$ 및 $yPos(n)$ 를 포함하여 상기 상자에 의해 덮인 영역의 수평 및 수직 위치의 정의가 아래와 같이 명확하게 구체화된다.

[0057] $b(t, n) = y(yPos(n):yPos(n)+yWidth(n)-1, xPos(n):xPos(n)+xWidth(n)-1, t)$

[0058] 예시적인 경계 상자가 도 13에 도시되어 있다. 그러므로, 상기 속성 상자는 4 개의 벡터 요소로 정의된다.

[0059] $metaX(t, n).box = [xPos(n) \ yPos(n) \ xWidth(n) \ yWidth(n)]$

[0060] d) 프레임 행렬 $y(:, :, t)$ 내에서 상기 객체에 의해 덮인 픽셀들에 관한 유일한 설명을 위하여 다양한 방법이 이용될 수 있다. '1'이 그 위치에 존재하는 객체를 나타내는 것이라고 할 때, 검출 이후에 유형 X 인 각각의 객체 n 은 바이너리 마스크 $maskX(1:yWidth(n), 1:xWidth(n), n)$ 에 의해 정의된다고 가정한다. 예를 들어, 도 13의 경계 상자에 도시된 상기 객체에 대한 바이너리 마스크는 이하의 표에 의해 설명될 것이다.

0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	0	0	1	0
0	0	1	1	1	1	1	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

[0061]

[0062] 표 1

[0063] 제1 방법에 따르면, 상기 바이너리 마스크의 정보는 아래와 같이 상기 바이너리 마스크 스스로를 저장함으로써 저장된다.

[0064]

$metaX(t,n).pixel = maskX(:, :, n)$

[0065]

다른 방법에 따르면, 아래와 같이 인덱스들의 목록이 '1'과 동일한 마스크($mask$)에 대하여 저장된다.

[0066]

$metaX(t,n).pixel = find(maskX(:, :, n))$

[0067]

e) 부울 플래그(Boolean flag) $meta.remove$ 는 상기 객체가 후속의 제거 단계에서 복원될 것인지 아닌지의 여부를 나타낸다. 상기 값은 자동 검출의 결정이 분명하지 않고 검토자의 승인을 필요로 하는 때마다 거짓(false)으로 설정될 수 있다.

[0068]

$metaX(t,n).remove = \{true | false\}$

[0069]

f) 스크래치 또는 정적 오염물과 같은 객체들은 보통 다수의 프레임에 걸쳐 퍼져있다. 그러므로, 상기 프레임 이전에($indexForward$) 및 이후에($indexBackward$) 동일한 객체의 인덱스를 저장하도록 제안된다. 영(null)은 응답이 없음을 나타내기 위해 사용된다.

[0070]

$metaX(t,n).indexForward = \{1, \dots, N_{forward} | 0\}$

[0071]

$metaX(t,n).indexBackward = \{1, \dots, N_{backward} | 0\}$

[0072]

g) 검토 및 통계의 목적으로 아래와 같이 모든 마스크에서의 '1'의 합계를 생성함으로써 검출되는 모든 객체들 또는 특정 유형 X 의 객체들의 영역을 저장하도록 제안된다.

[0073]

$metaX(t).areaDetected = sum(maskX(:, :, :))$

[0074]

h) 마찬가지로, 검토 및 통계의 목적으로 프레임 t 내에서 검출되고 'true'와 동일한 $metaX.remove(t,n)$ 에 대한 모든 마스크에서 '1'의 합계를 생성함으로써 제거될 모든 객체들 또는 특정 유형 X 의 객체들의 영역을 저장하도록 제안된다.

[0075]

$metaX(t).areaRemoved = sum(maskX(:, :, metaX(t,n).remove))$

[0076]

i) 편리하게 $metaX(t,n)$ 은 아래의 필드를 포함하고

[0077]

$metaX(t,n).replacement,$

[0078]

이는 제거를 어떻게 실행하는지, 예를 들어 만약 $metaX(t,n).remove = true$ 라면 $metaX(t,n).pixel$ 로 주어진 객체 픽셀들이 어떤 방법으로 변경되거나 대체되어야 하는지에 관한 정보를 제공한다. $metaX(t,n).remove$ 가 품질 제어 단계(24) 중 변경될 수 있으므로, 가급적이면 $metaX(t,n)$ 의 상태를 무시하는 것인 $metaX(t,n).replacement$ 가 항상 정의된다. 상기 필드는 제거 단계(21) 중 적합한 대체 패턴을 찾기 위한 처리가 요구되지 않는 이점을 갖는다. 이는 비록 검출에 시간이 더 오래 걸리긴 하지만, 제거는 계산 상 덜 부담이 된다는 것을 의미한다. 또한, 메타데이터의 양이 증가한다.

[0079]

제1 가능성에 따르면, 주위 상자(surrounding box) $b(n)$ 을 구비한 객체 n 에 대한 $metaX(t,n).replacement$ 는 상기 객체에 의해 덮이는 $r(n)$ 의 값만을 직접 저장함으로써 아래와 같이 얻어진다.

[0080]

$metaX(t,n).replacement = r(metaX(t,n).pixel)$

[0081]

대안적으로, $b(n)$ 과 $r(n)$ 간 차이가 아래와 같이 저장된다.

- [0082] $metaX(t,n).replacement = r(metaX(t,n).pixel) - b(metaX(t,n).pixel)$
- [0083] 또 다른 대안에 따르면, 만약 $r(n)$ 이 현재의 또는 이전이나 이후의 기타 다른 프레임에서의 이미지 영역으로부터 복제함으로써 발견된다면, 3 요소의 벡터가 수평 오프셋 $yOff(n)$, 수직 오프셋 $xOff(n)$ 및 시간 오프셋 $tOff(n)$ 을 이용하여 저장된다.
- [0084] $metaX(t,n).replacement = [yOff(n) \ xOff(n) \ tOff(n)]$
- [0085] 품질 제어 단계(24)를 검토하는 것은 '훌륭한(excellent)' 복원 결과를 위한 메타데이터 세트(15)를 얻기 위해 이미 '충분히 좋은(good enough)' 결과를 내는 메타 데이터 세트(15)를 편집할 수 있도록 한다. 물론, 실제 결과는 운영자(9)에 의해 소요되는 시간에 따라 결정된다. 그 외에, 품질 제어 단계(24)는 또한 데이터 세트를 수동으로 수정하지 않고서 알고리즘 품질을 제어하는 데에 유용할 수 있다. 품질 제어는 메타데이터를 확인하고 수정함으로써 실행된다. 이는 측정가능한 수단을 통해 더욱 객관적으로 그리고 상호작용으로 나란히 프레임을 비교하는 것보다 훨씬 빠르게 품질을 제어할 수 있도록 한다. 메타데이터가 승인된 이후, 원본 파일에 대한 품질 보고서가 가급적이면 복원 형태에 대한 변경 기록과 함께 생성된다.
- [0086] 가급적이면 그래픽 사용자 인터페이스를 이용한 응용 소프트웨어 내에서 메타데이터 데이터베이스(15)의 그래픽 표현을 사용한다. 예시적인 그래픽 표현이 도 14 및 도 15에 도시되어 있다. 편리하게 상기 그래픽 사용자 인터페이스는 집계된 메타데이터의 시각화를 제공한다. 오검출을 나타내는 이상치(outliers)는 이 방법으로 쉽게 발견될 수 있다. 일반적으로 수적으로 적은 해당 프레임들은 보다 자세한 조사를 필요로 한다. 이러한 목적을 위해, 프레임 내에서 검출된 스크래치 및 오염물 객체들(50, 51)은 경계 상자 좌표 속성을 사용하여 주석이 붙여진다. 상기 제거 플래그 속성에 따라, 가급적이면 다양한 방식들이 상기 경계 상자를 위해 사용된다. 잘못 검출된 객체들은 이 방법으로 빠르게 수집될 수 있다.
- [0087] 만약 상기 설명한 바와 같이 잘못된 검출이 식별되었다면, 메타데이터의 품질을 개선하는 것이 필요하다. 해당 메타데이터는 수정되거나 확장된다. 오탐된 스크래치 및 오염물 객체는 메타데이터 데이터베이스(15)에서 제거되거나 상기 토크 플래그 속성을 설정함으로써 제거에 대해 선택 해제된다. (미탐으로) 전혀 검출되지 않았거나 일부 검출되지 않은 스크래치 및 오염물 객체는 사람이 직접 표시하거나 메타데이터 데이터베이스(15)에 추가시킬 수 있다.
- [0088] 이제 도 14 및 도 15에 도시된 그래픽 사용자 인터페이스를 더 구체적으로 참조하여, 제1 패널(41)에서 각각의 객체(50, 51)는 $metaX(t,n).box$ 에 저장된 크기 및 좌표에 따라 현재 프레임 t 의 콘텐츠의 위에 직사각형 상자를 그림으로써 주석이 붙여진다. 제거될 예정인 객체(50), 예를 들어 $metaX(t,n).remove = true$ 인 객체는 상이한 색상 사용함으로써 그리고/또는 상자의 선에 대시를 표시함(dashing)으로써 제거되지 않을 예정인 객체(51), 예를 들어 $metaX(t,n).remove = false$ 인 객체로부터 편리하게 구별된다.
- [0089] 제2 패널(42)에서, 상기 벡터 $metaX(t).areaDetected$ 및/또는 $metaX(t).areaRemoved$ 가 모든 프레임에 대하여 시각화된다. 프레임 수 I 가 보통 디스플레이의 수평 해상도보다 훨씬 크기 때문에, 전체 데이터를 제2 패널(42)로 피팅하기 위해 데이터 축소(data reduction)가 수행된다. 요소 $metaX(t).areaDetected$ 는 항상 $metaX(t).areaRemoved$ 와 동일하거나 그보다 크기 때문에, 두 개의 막대 플롯(bar plots)은 $metaX(t).areaDetected$ 의 앞쪽에 $metaX(t).areaRemoved$ 를 갖는 동일한 패널(42) 내에 도시된다.
- [0090] $metaX(t).areaDetected$ 의 앞쪽에 $metaX(t).areaRemoved$ 를 갖는 추가적인 두 개의 막대 플롯이 제3 패널(43)에 도시된다. 제2 패널(42)과 대조적으로, $metaX(t).areaDetected$ 의 앞쪽에서의 $metaX(t).areaRemoved$ 의 각각의 요소는 한 개의 픽셀보다 큰 폭을 갖는 분리된 막대에 의해 표현된다. 프레임 수 I 가 보통 디스플레이의 수평 해상도보다 훨씬 크기 때문에, 상기 플롯의 일부분만이 보인다. 이러한 일부분은 현재의 프레임 t 에 해당하는 막대가 항상 상기 플롯의 중앙에 있도록 하는 방법으로 선택된다.
- [0091] 잘못 검출된 객체들이 제거되는 것을 방지하기 위해, $metaX(t,n).removed$ 의 객체 상태와 상기 주석 상자의 색상 및/또는 대시 표시를 적절히 전환(toggle)하도록 해당 상자를 클릭하거나 터치한다. $metaX(t,n).removed$ 의 상태를 전환할 때, 제2 패널(42) 및 제3 패널(43)에 도시된 플롯뿐만 아니라 $metaX(t).areaRemoved$ 의 누적된 영역이 알맞게 즉시 업데이트된다.
- [0092] 가급적이면 제2 패널(42) 및 제3 패널(43)에서 상기 플롯들의 특정 위치를 클릭이나 터치할 때마다 상기 그래픽 사용자 인터페이스는 해당 프레임으로 전환한다. 제3 패널(43)에서의 상기 메타데이터 막대 플롯 간격뿐만 아니라 제1 패널(41)에 도시된 상기 콘텐츠 및 주석이 즉시 업데이트된다.

- [0093] 제4 패널(44)은 제1 패널(41)에서 스크래치만, 오염물만 또는 모두를 디스플레이 되도록 할 것인지 여부를 선택할 수 있도록 한다. 그 외에, 제5 패널(45)은 제1 패널(41)에서 원본 콘텐츠 대신 제거 절차(21)의 결과인 출력을 미리 볼 수 있도록 한다. 이러한 목적을 위해, 제4 패널에서의 스크래치 및/또는 오염물에 대한 확인상자가 활성화되고 제5 패널(45)에서의 해당 선택을 클릭함으로써 화면이 제거된(필터링된) 것으로 변경되거나 또는 단축키를 누를 때마다 제거 절차(21)가 현재의 프레임 t 에 적용된다. 대안적으로 또는 추가적으로, 전용 삭제/적용 버튼(도시되지 않음)이 눌러질 때 제거 절차(21)가 현재 프레임 t 에 적용된다.
- [0094] 또한, 그래픽 사용자 인터페이스는 아래의 인덱스
- [0095] $metaX(t, n).index = n + 1$
- [0096] 및 아래의 유형
- [0097] $metaX(t, n+1).type = \{ 'scratch' | 'dirt' | 'static dirt' | \dots \}$
- [0098] 을 갖는 추가 객체 $metaX(t, n+1)$ 을 추가할 수 있도록 한다.
- [0099] 이러한 객체는 마우스 자유곡선 도구(mouse freehand tool) 또는 그와 유사한 것을 사용하여 제1 패널(41)에서의 위치 $[xPos \ yPos]$ 에서 폭 $[xWidth \ yWidth]$ 를 갖는 아래의 영역
- [0100] $metaX(t, n+1).box = [xPos \ yPos \ xWidth \ yWidth]$
- [0101] 을 아래와 같이 표시함으로써 플래깅된다.
- [0102] $metaX(t, n+1).remove = true$
- [0103] 그렇게 함으로써 상기 d)에 이미 설명된 바와 같이 $metaX(t, n+1).pixel$ 을 설정하기 위해 $maskX(:, :, n+1)$ 가 정의된다. 또한, 세트는 f)에 따른
- [0104] $metaX(t, n+1).indexForward = \{1, \dots, N_{forward} | 0\}$
- [0105] $metaX(t, n+1).indexBackward = \{1, \dots, N_{backward} | 0\}$,
- [0106] g)에 따른
- [0107] $metaX(t).areaDetected = metaX(t).areaDetected + \sum(maskX(:, :, n+1))$,
- [0108] h)에 따른
- [0109] $metaX(t).areaRemoved = metaX(t).areaRemoved + \sum(maskX(:, :, n+1))$,
- [0110] 및 i)에 따른
- [0111] $metaX(t, n).replacement$
- [0112] 와 같은 입력들(entries)이다.
- [0113] 마지막으로, 스크래치 및 오염물 객체 제거(21)가 복원된 출력(8)을 생성하도록 메타데이터 데이터베이스(15)를 사용하여 자동으로 실행된다. 그러므로, 결과는 완전히 예측가능하며, 원본 파일(7)은 객체가 검출되고 메타데이터가 제거를 승인한 위치에서만 수정된다. 더 구체적으로, 필름 복원은 각각의 프레임 t 에 메타데이터 $metaX(n)$ 을 적용함으로써 수행된다. 만약 $metaX(t, n).remove = true$ 라면, $metaX(t, n).pixel$ 에 의해 정의된 객체 n 의 픽셀들이 수정된다. 상기 프레임 t 내의 기타 다른 모든 픽셀들은 그대로 남아있다.
- [0114] 이하의 절차는 각각의 객체마다 수행된다. 또한, " "가 표시된 변수들은 "그 자리에서", 예를 들어 추가 메모리를 할당하지 않고 처리될 수 있으며 예시를 위해 아래와 같이 도입된다.
- [0115] a. 프레임 t 에 대하여, 입력 행렬 $y(t)$ 로 복원된 출력 행렬 $y'(t)$ 를 설정한다.
- [0116] $y'(t) = y(t)$
- [0117] b. $metaX(t, n).box$ 의 경우, c)에 설명된 바와 같이 경계 상자 행렬 $b(t, n)$ 을 찾는다.
- [0118] c. 상기 경계 상자 행렬 $b(t, n)$ 으로 $b'(t, n)$ 을 설정한다.
- [0119] $b'(t, n) = b(t, n)$

d. $metaX(t,n).type$ 에 따라, $b'(t,n)$ 에서의 $metaX(t,n).pixel$ 로 주어진 픽셀을 수정하는 방법을 선택한다.

i. $b'(t,n,metaX(t,n).pixel)=metaX(t,n).replacement$

ii. $b'(t,n,metaX(t,n).pixel)=b'(t,n,metaX(t,n).pixel)+metaX(t,n).replacement$

iii. $metaX(t,n).replacement$ 로부터 $yOff(n)$, $xOff(n)$ 및 $tOff(n)$ 을 추출함으로써 임시 경계 상자 행렬 b'' 를 아래와 같이 설정한다.

$b''=y(yPos(n)+yOff(n):yPos(n)+yOff(n)+yWidth(n)-1,xPos(n)+xOff(n):xPos(n)+xOff(n)+xWidth(n)-1,i+tOff(n))$

그리고 아래와 같이 설정한다.

$b'(t,n,metaX(t,n).pixel)=b''(metaX(t,n).pixel)$

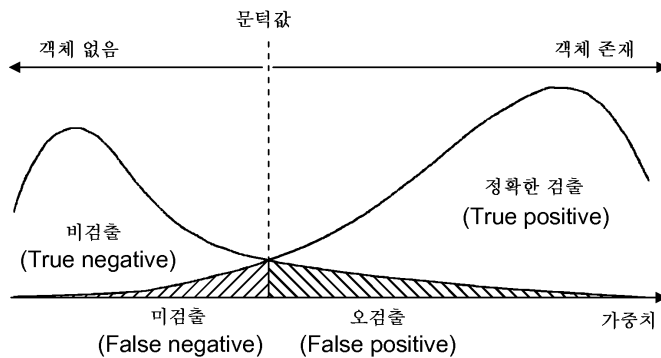
e. 아래와 같이 복원된 것으로 $y(n)$ '에서의 객체 i 를 재정의(override)한다.

$y'(yPos(n):yPos(n)+yWidth(n)-1,xPos(n):xPos(n)+xWidth(n)-1,i)=b'(t,n)$

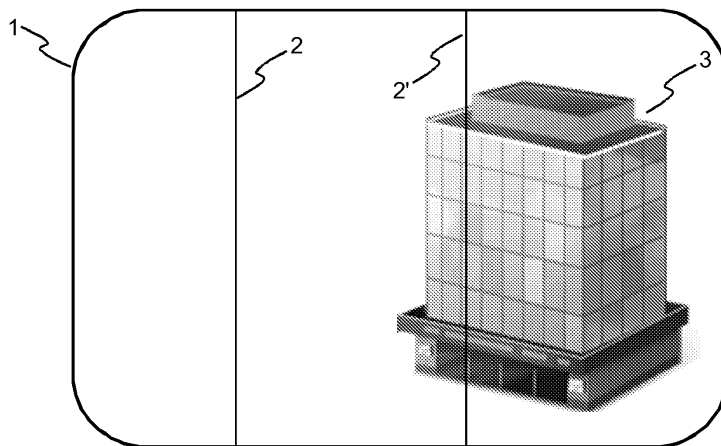
도 16 내지 18은 복원 워크플로우의 예시적인 결과를 도시한다. 도 16에서 어떠한 복원도 이루어지지 않은 원본이 도시된다. 도 17은 메타데이터 주석을 포함하는, 예를 들어 검출 절차(20)가 수행된 이후의 동일한 이미지를 묘사한다. 도 18은 제거 절차(21)의 적용 이후의 출력 이미지를 도시한다.

도면

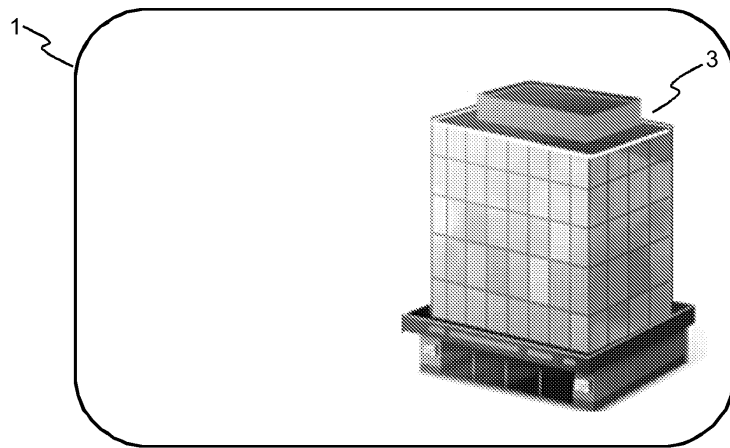
도면1



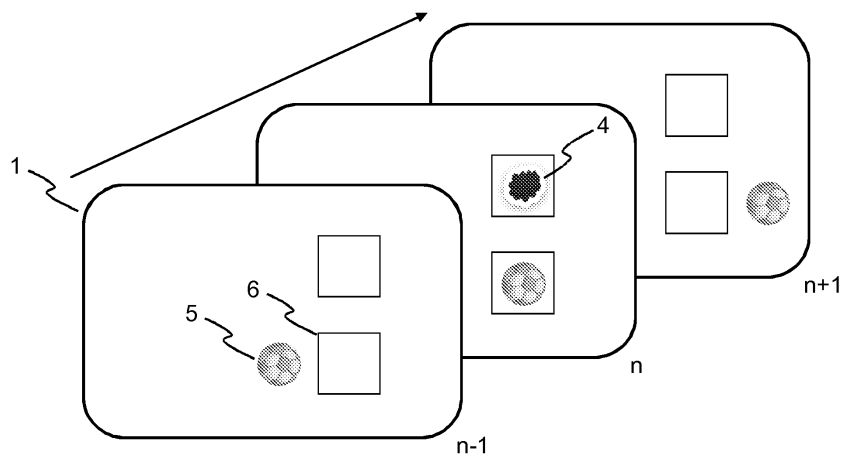
도면2



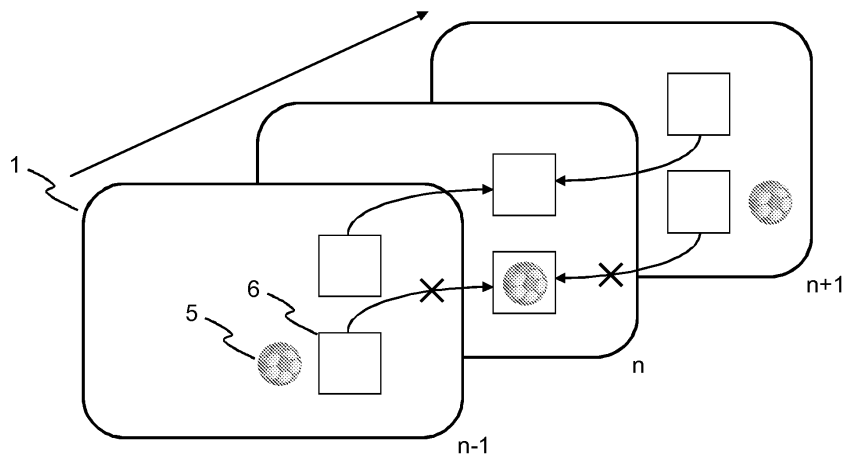
도면3



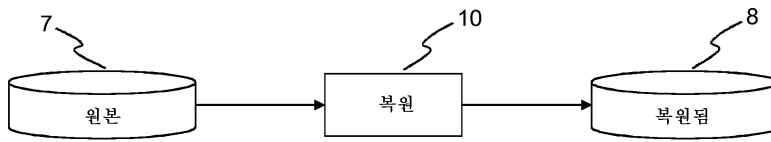
도면4



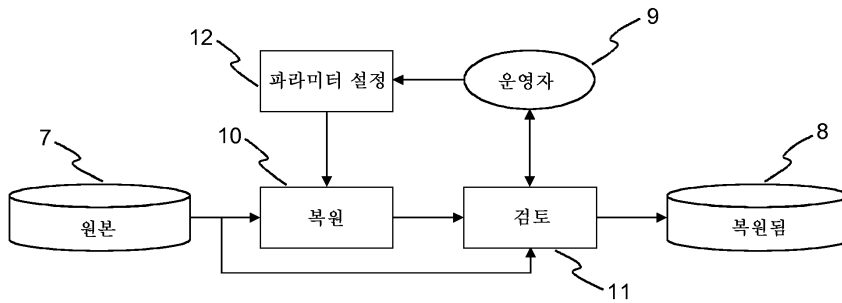
도면5



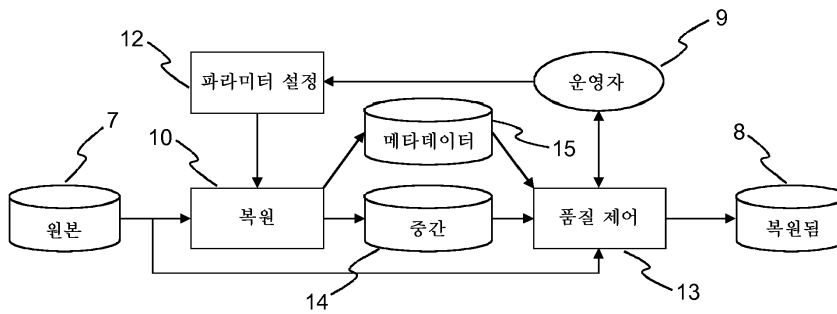
도면6



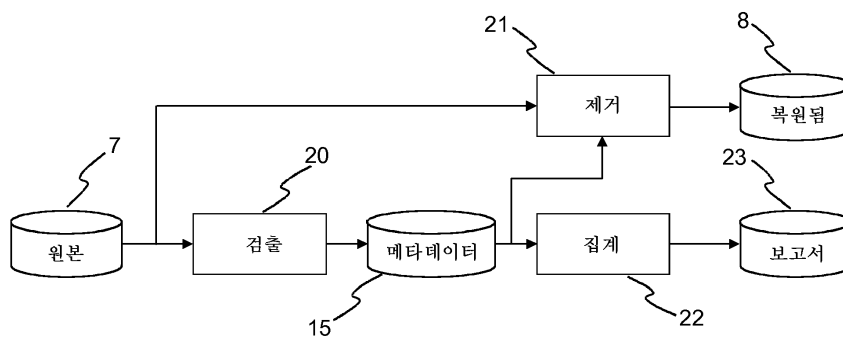
도면7



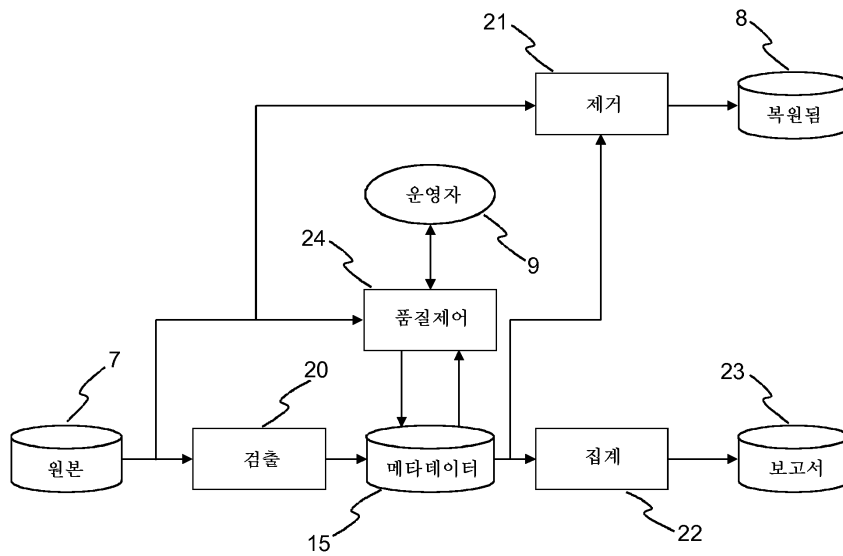
도면8



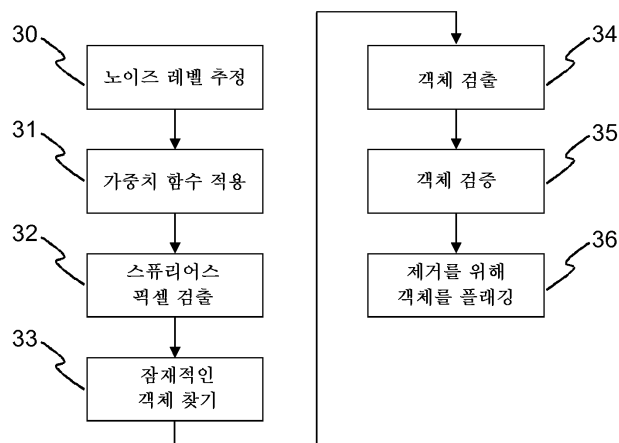
도면9



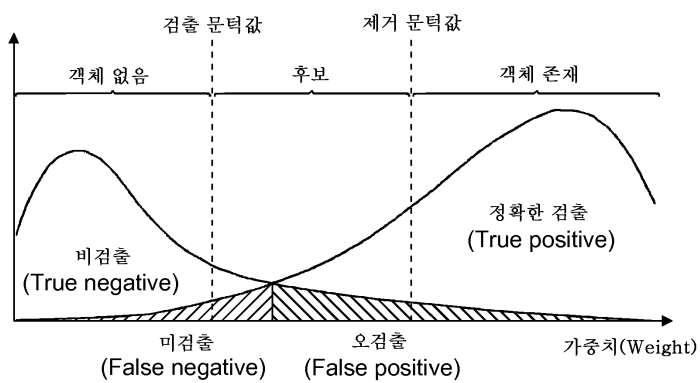
도면10



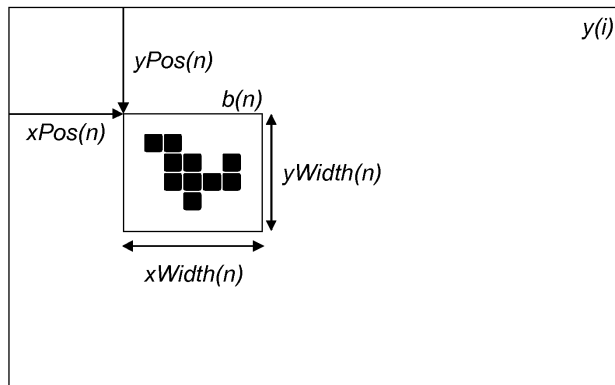
도면11



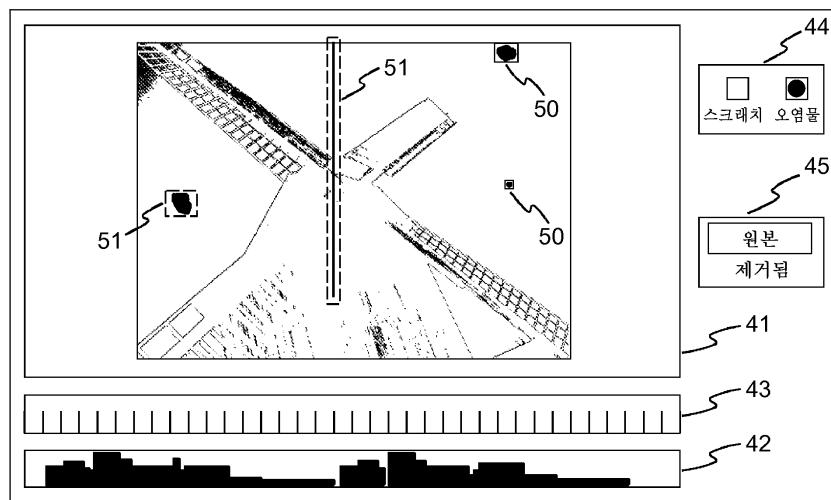
도면12



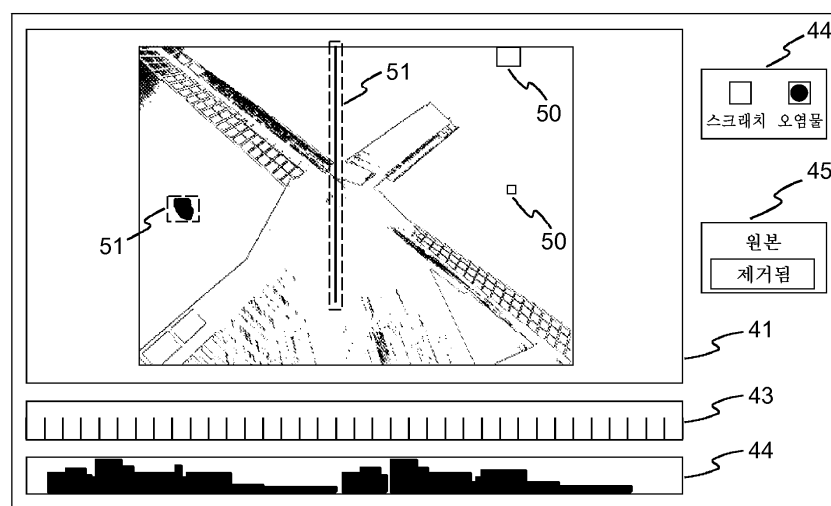
도면13



도면14



도면15



도면16



도면17



도면18

