(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl.[6]
H04N 7/13

(45) 공고일자 1997년06월21일
(11) 공고번호 특1997-0010101
(24) 등록일자 1997년06월21일

(21) 출원번호 특1994-0015373
(22) 출원일자 1994년06월30일
(65) 공개번호 특1996-0003420
(43) 공개일자 1996년01월26일

(73) 특허권자 대우전자주식회사 배순훈
서울특별시 중구 남대문로 5가 541번지
(72) 발명자 김상호
서울특별시 강서구 화곡 8동 391-1
(74) 대리인 장성구, 김원준

심사관 : 이종배 (책자공보 제5077호)

(54) 디지틀 전송 화상의 위치 적응적 후처리 방법

요약

내용없음.

대표도

도1

명세서

[발명의 명칭]

디지틀 전송 화상의 위치 적응적 후처리 방법

[도면의 간단한 설명]

제1도는 본 발명이 적용되는 하드웨어의 개략도.

제2도는 본 발명에 따른 후처리 방법에 대한 처리흐름도.

[발명의 상세한 설명]

본 발명은 디지틀 전송 화상의 위치 적응적 후처리에 관한 것으로, 특히 전송 영상의 주관 화질을 높이기에 적합한 디지틀 전송화상의 위치 적응적 후처리 방법에 관한 것이다.

디지틀 전송 화상에서 블럭 경계가 보이는 블럭킹 효과(Blocking effect)를 제거하기 위한 가장 일반적인 방법은 2차원저역 필터(Low Pass Filter; LPF)를 사용하여 후처리를 하는 것이지만, 이 경우 일정량의 블럭킹 현상이 감소하기는 하지만 세밀한 영상을 둔화시키고, 블럭킹 효과의 제거도 충분하지 않다는 문제점이 있었다.

상기 종래 기술에 대한 제반 문제점을 해결하기 위하여 안출된 본 발명은, 블럭 경계가 보이는 블럭킹 효과를 제거하여 전송영상의 주관 화질을 높이는 디지틀 전송 화상의 위치 적응적 후처리 방법을 제공하는데 그 목적이 있다.

상기 목적을 달성하기 위하여 본 발명은, 후처리 방법에 있어서, 입력 영상신호를 저역 패스 필터를 통과시킨 출력값을 두고, 그 영상신호를 이용하여 블럭당 평균 분산값 및 현재 블럭값을 계산하여 설정된 임계치를 이용하여 출력값에서 입력값을 뺀 절대값과 상기 임계치를 비교하는 제1단계; 상기 제1단계 수행후, 임계치가 절대값보다 작으면 입력값에 임계치를 더하거나 뺀 값으로 출력값을 설정하고, 작지 않으면 그대로 상기 저역 패스 필터의 통과 횟수를 증가시키는 제2단계; 및 상기 제2단계 수행 후, 상기 저역 패스 필터를 설정한 최대 횟수만큼 통과시켜 블럭킹 효과를 제거하는 제3단계를 포함하여 이루어지는 것을 특징으로 한다.

이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세히 설명하면 다음과 같다.

제1도를 참조하면, 제1도는 본 발명이 적용되는 하드웨어와 개략도로서, 1은 가변길이 부호화부, 2는 역양자화부, 3은 역이산 코사인 변환부, 4는 가산연산부, 5는 움직임 보상부, 6은 프레임 메모리부, 7은 후처리부 그리고 8은 디스플레이부를 각각 나타낸다.

이와같은 본 발명이 적용되는 하드웨어의 구성은, 입력 비트스트림(Bit stream)을 입력받아 가변길이 부호화하는 가변길이 부호화부(1), 가변길이 부호화부(1)의 신호를 인가받아 역양자화하는 역양자화부(2), 역양자화부(2)의 신호를 인가받아 역이산 코사인 변환하는 역이산 코사인 변환부(3), 역이산 코사인 변환부(3)의 신호와 움직임 보상신호를 가산 연산하여 출력하는 가산 연산부(4), 가산 연산부(4)를 통해 출력된 값을 프레임 단위로 인가받아 저장하는 프레임 메모리부(6), 프레임 메모리부(6)의 출력을 입력받아

움직임 보상후 그 움직임 보상된 신호를 가산 연산부(4)로 인가하는 움직임 보상부(5), 가산 연산부(4)로부터의 출력을 순차로 입력받아 후처리 한 후 디스플레이부(8)로 제공하는 후처리부(7)로 이루어진다.

이와 같이 이루어진 본 발명을 제2도를 참조하여 보면, 먼저 제2도와 같이 저역 패스 필터의 입력영상신호(in[i][j])가 저역 패스 필터를 통과한 횟수(iter)가 '0'이면(21) 저역 패스 필터는 아래 표 1과 같은 2차원 필터를 통과시키는 것, 혹은 표 2와 같은 1차원 필터를 가로, 세로 방향으로 한번씩 적용하는 등의 방법을 사용하여 출력값(out[i][j])를 출력한다(22).

그리고, 블럭당 평균 분산값 계산 및 저장은 한 프레임의 주기로 작동하며, 한 프레임내의 블럭들의 분산을 구하여 그 분산의 평균값을 구하고 그 값을 저장한다(23).

이어, 현재 블럭 분산값 계산 및 저장은 하나의 블럭당 계산하며, 현재 처리중인 화소가 속한 블럭의 분산값을 계산하여 저장한다(24).

그리고, 분산값 비교 및 임계치(Thr) 판단은 전체 블럭들의 평균 분산값과 현재 블럭의 분산값을 비교하여 현재 블럭 분산값이 전체 평균 분산보다 크면, 작은값(예를들면, Thr=4)을 할당하고, 현재 블럭 분산값이 전체 평균 보다 작으면 큰 값(예를들면, Thr=6)을 할당한다(25).

이와 같이 하는 이유는 현재 분산이 크다는 것은 현재 처리할 화소가 윤곽 부분에 있다는 것을 뜻하기 때문에 윤곽을 좀더 보존하기 위하여 임계치를 작은 값으로 셋팅하는 것이다.

또한, 윤곽부(도시하지 않음)에서는 블럭킹 효과가 잘 보이지 않기 때문에 임계치를 크게 하여 블럭킹 효과를 충분히 제거할 필요가 없기도 하다.

상기에 반해, 현재 블럭 분산이 작다는 것은 현재 처리중인 화소가 속한 블럭이 평탄한 부분의 블럭이라는 뜻이므로 임계치를 크게하여 충분히 블럭킹 효과를 제거해 주어야 한다.

또한, 이 블럭내에는 윤곽부가 없을 확률이 크기 때문에 임계치를 키워도 윤곽이 흐려지는 문제점은 발생하지 않는다.

이와같은 임계값은 본 발명의 후처리 필터의 최초 입력과 최후 출력과의 가능한 범위내의 최대 차이로서, 이 제한이 있으므로 해서 반복적인 저역 패스 필터를 통과시켜도 윤곽이 흐려지는 것을 방지할 수 있는 것이다.

그리고, 저역 패스 필터의 출력값에서 입력값을 뺀 값의 절대치가 임계치 보다 큰가를 조사하여(26) 클경우 출력값이 입력값보다 크면 입력값에 임계치를 더하여 출력값을 지정하고, 출력값이 입력값보다 크지 않으면 입력값에서 임계치를 뺀 값을 출력값으로 지정한다(27 내지 29).

상기 판단(26) 결과 크지 않을 경우 저역 패스 필터의 통과 횟수에 1을 더하여(30) 통과 횟수와 설정된 최대값을 넘지 않도록 저역 패스 필터를 통과시키고 종료한다(31).

여기서, 최대값(MAX)는 최대 통과 횟수로 약 3 내지 5회 정도면 블럭킹 효과가 충분히 제거되나, 최대값을 크게 설정하면 블럭킹 효과는 충분히 제거되지만 후처리에 걸리는 시간이 많이 소요되고, 너무 적게 설정하면 블럭킹 효과의 제거가 불충분하다.

따라서, 상기와 같은 본 발명은 현재 처리중인 화소가 속한 블럭이 윤곽부인지 아닌지를 판단하여 다른 임계치를 가지로 후처리를 함으로써, 윤곽을 좀더 보존하고 블럭킹 효과가 눈에 잘 띄는 평탄부에서는 좀더 블럭킹을 강하게 제거할 수 있는 효과가 있다.

[표 1]

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

표 2

1/4	1/2	1/4

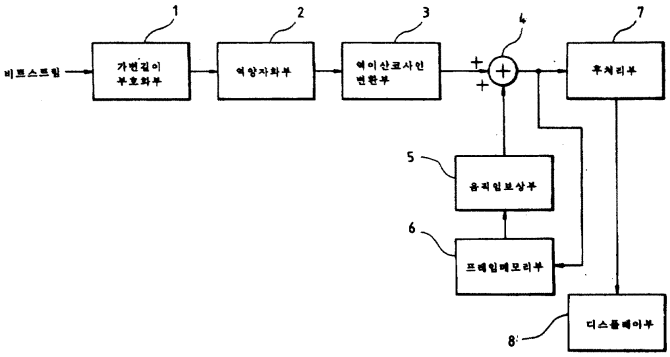
(57) 청구의 범위

청구항 1

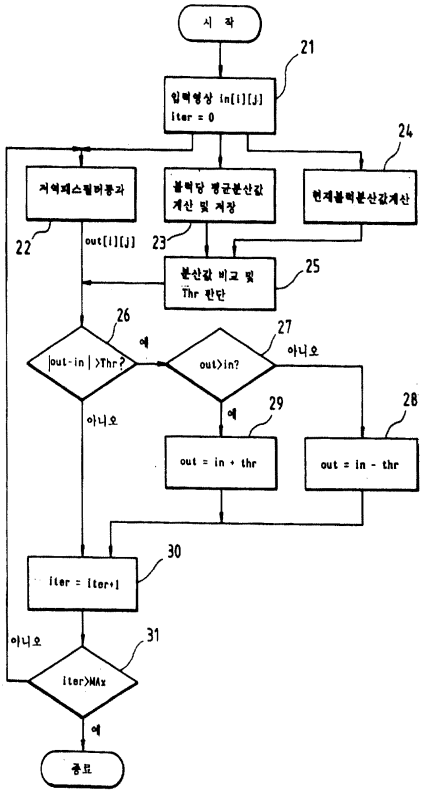
디지탈 전송 화상의 후처리에 있어서, 입력 영상신호를 저역 패스 필터를 통과시킨 출력값을 두고, 그 영상신호를 이용하여 블럭당 평균 분산값 및 현재 블럭값을 계산하고 설정된 임계치를 이용하여 출력값에서 입력값을 뺀 절대값과 상기 임계치를 비교하는 제1단계(21 내지 26); 상기 제1단계(21 내지 26) 수행 후, 임계치가 절대값 보다 작으면 입력값에 임계치를 더하거나 뺀 값으로 출력값을 설정하고 작지 않으면 그대로 상기 저역 패스 필터의 통과 횟수를 증가시키는 제2단계(27 내지 30); 상기 제2단계(27 내지 30) 수행 후, 상기 저역 패스 필터를 설정한 최대 회수만큼 통과시켜 블럭킹 효과를 제거하는 제3단계(31)를 포함하여 이루어지는 디지틀 전송 화상의 위치 적응적 후처리 방법.

도면

도면1


비트스트림
가변길이 부호화부
역양자화부
역이산코사인 변환부
+
+
+
후처리부
움직임보상부
프레임메모리부
디스플레이부
1
2
3
4
5
6
7
8

도면2


시 작
입력영상 in[i][j]
iter = 0
지역피스필터링과
블록당 평균분산값 계산 및 저장
현재블록분산값계산
out[i][j]
분산값 비교 및 Thr 판단
|out-in| >Thr?
예
out>in?
아니오
아니오
예
out = in + thr
out = in - thr
iter = iter+1
아니오
iter>MAx
예
종료
21
22
23
24
25
26
27
28
29
30
31