

Mean-Shift Algorithm을 이용한 Image inpainting에 관한 연구

공재웅*, 정재진*, 황의성*, 김태형*, 김두영*
*동아대학교 전자공학과

A Study on Image inpainting using Mean-Shift Algorithm

Jae-Woong Gong*, Jae-Jin Jung*, Eui-Sung Hwang*, Tae-Hyoung Kim*,
Doo-Yung Kim*

*Dept. of Electronics Engineering, Dong-A University

E-Mail : netgjw@nate.com

요 약

오늘날 컴퓨터의 발달과 인터넷의 확산으로 멀티미디어 콘텐츠의 보급이 급격히 확대되고 있으며, 이들 콘텐츠에는 원거리 화상회의, 감시시스템, 주문형 비디오(VOD), 주문형 뉴스(NOD), 디지털 편집 시스템 등 동영상도 포함되어 있다. 이처럼 동영상은 정보교환과 정보표현의 매개체로서 중요한 역할을 한다. 그러나 이와 같은 동영상은 노이즈나 전송과정 중 발생하는 문제 등으로 인해 항상 좋은 품질을 보장되지 않는다. 이런 훼손된 영상을 원영상으로 복원하거나 사용자가 제거 혹은 복원하고자 하는 영역을 지정 처리함으로써 다양한 정보를 획득할 수 있다. 일반적으로 pc에서 사용되어지는 대부분의 동영상은 15fps~30fps이다. 대부분의 동영상 편집 기술은 각각의 frame을 추출하여 수동적으로 처리하므로 비용과 시간이 많이 든다. 이런 단점을 해결하기 위해 여러 방법이 기존에 시도되고 있다. 제거 혹은 복원하고자 하는 영역을 전 frame에서 처리하기 위해 움직임 검출 및 추적기법이 사용되며, 제거 혹은 복원하기 위해 median filtering, image inpainting 처리 방법들이 있다. 본 연구에서는 사용자의 해 미리 정의된 바운딩 박스내의 객체를 추적하여 객체의 중심값을 찾는 mean-shift algorithm을 이용하여 움직이는 객체를 추적하였고 image inpainting algorithm을 이용하여 훼손된 영역을 복원하거나 제거하고자 하는 객체를 제거하였다.

I. 서론

오늘날 컴퓨터의 발달과 인터넷의 확산으로 멀티미디어 콘텐츠의 보급이 급격히 확대되고 있으며, 이들 콘텐츠에는 원거리 화상회의, 감시시스템, 주문형 비디오(VOD), 주문형 뉴스(NOD), 디지털 편집 시스템 등 동영상도 포함되어 있다. 이처럼 동영상은 정보교환과 정보표현의 매개체로서 중요한 역할을 한다. 그러나 이와 같은 동영상은 노이즈나 전송과정 중 발생하는 문제 등으로 인해 항상 좋은 품질을 보장되지 않는다. 또한 사용자가 원하지 않은 정지된 객체, 움직이는 객체를 제거하거나 복원시키고자 할 때도 있다. 이런 훼손된 영상을 원영상으로 복원하거나 사용자가 제거 혹은 복원하고자 하는 영역을 지정 처리함으로써 다양한 정보를 획득할 수 있다.

일반적으로 pc에서 사용되어지는 대부분의 동영상은

15fps~30fps 이다. 대부분의 동영상 편집 기술은 각각의 frame을 추출하여 수동적으로 처리하므로 비용과 시간이 많이 든다.

이런 단점을 해결하기 위해 여러방법이 기존에 시도되고 있다. 제거 혹은 복원하고자 하는 영역을 전 frame에서 처리하기 위해 움직임 검출 및 추적기법이 사용되며, 제거 혹은 복원하기 위해 median filtering, image inpainting 처리 방법들이 있다.[1]

움직임 검출 및 추적기법을 살펴보면 neural network을 이용한 기법과 histogram을 이용한 기법이 많이 사용되고 있다. S.Birchfield는 일정한 영역을 탐색하면서 컬러의 히스토그램과 객체의 그라디언트(gradient)를 이용하여 객체 추적을 하였다. 그러나 일정영역 탐색에 많은 시간을 소비하며 초당 프레임 비율이 낮을 경우 부적절하

다.

훼손된 영상을 복원하기 위해 Median Filtering와 같은 기존의 처리 방법들은 주변 화소(pixel)를 평활화(smoothing) 처리를 하기 때문에 noise 처리에는 좋으나 원영상의 중요한 에지 성분까지도 평활화 처리를 함으로서 에지 부분의 공간적 이동을 초래할 수 있다. 이러한 문제점을 해결하기 위하여 image inpainting 방법이 제안되고 있으며, inpainting 기법에는 편미분 방정식(PDE)을 이용한 방법, 텍스처 병합 기반의 방법들이 있다.[4]

본 연구에서는 사용자에게 의해 미리 정의된 바운딩 박스 내의 객체를 추적하여 객체의 중심값을 찾는 mean-shift algorithm을 이용하여 움직이는 객체를 추적하였고 image inpainting algorithm을 이용하여 훼손된 영역을 복원하거나 제거하고자 하는 객체를 제거하였다.

II. 배경이론

2.1 Mean-Shift Algorithm

본 연구에서 객체의 추적은 mean-shift algorithm을 이용한다. Fukunaga에 의해 제안된 mean-shift algorithm은 일반화된 클러스터링 알고리즘으로 객체의 segmentation, 필터링, 최적화 문제의 해결 등에 유용하게 사용되었다. mean-shift algorithm은 일련의 확률분포의 최고치를 최대경사법을 기반으로한 반복적인 갱신에 의해 추정할 수 있다. 기존의mean-shift의 단점은 작은 영역에 큰 영역이 겹쳐지면 큰 영역을 찾는다. 또한 큰영역을 설정할 경우 여러 자기 잡음이 포함되어 진다. 마지막으로 mean-shift algorithm은 움직이는 객체를 추적 하지 못한다. 이러한 단점을 해결하기 위해 mean-shift algorithm을 개선하였다.

개선한 mean-shift algorithm은 다음과 같다.[5]

- (1) 최초 바운딩 박스 형태의 관심 영역 설정
- (2) 관심 영역의 중심점과 사이즈로 조금 확장 시킴
- (3) row image를 hsv image로 변환
- (4) hsv image에서 관심영역의 histogram 생성
- (5) histogram back-projection으로부터 2D 컬러 확률 분포 image 생성
- (6) 생성된 컬러 확률 분포를 가지고 Mean-Shift Algorithm 적용
- (7) 선택된 영역에서의 중심점을 찾는다.

zero moment를 찾는다.

$$M_{00} = \sum_x \sum_y I(x,y)$$

x,y를 위한 first moment를 찾는다

$$M_{10} = \sum_x \sum_y xI(x,y),$$

$$M_{01} = \sum_x \sum_y yI(x,y)$$

선택 영역의 평균을 찾는다.

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$$

$I(x,y)$: position(x,y)의 화소값

(8) 중심점으로부터 새로운 영역을 설정한다.

(9) 중심점이 변하지 않거나 Threshold값 이하 또는 정해진 반복 횟수까지 (7),(8)단계를 반복한다.

(10) 계산된 x,y,z,θ 값을 저장한다.

Second moment

$$M_{20} = \sum_x \sum_y x^2 I(x,y),$$

$$M_{02} = \sum_x \sum_y y^2 I(x,y)$$

주축의 방향

$\theta =$

$$\arctan \left\{ \frac{2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left(\frac{M_{20}}{M_{00}} - x_c^2 \right) - \left(\frac{M_{02}}{M_{00}} - y_c^2 \right)} \right\}$$

$$a = \frac{M_{20}}{M_{00}} - x_c^2, b = 2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right),$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2$$

장축의 길이와 단축의 길이

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}$$

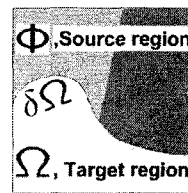
$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}$$

(11) 검출된 영역을 중심과 크기로 위치한다.

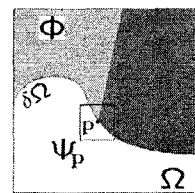
(12) 다음 frame으로 2단계부터 반복한다.

2.2 텍스처 병합 기반의 Image inpainting

텍스처 병합 기반의 Image inpainting 방법은 제거될 영역의 경계에서 우선순위에 따라 일정 크기의 표본을 선택하여 제거될 영역을 제외한 영역에서 표본을 비교하여 오차가 가장 적은 표본을 찾아 제거될 영역에 채우는 방법이다. 텍스처 병합 기반의 Image inpainting 과정은 다음과 같다.



(a)



(b)

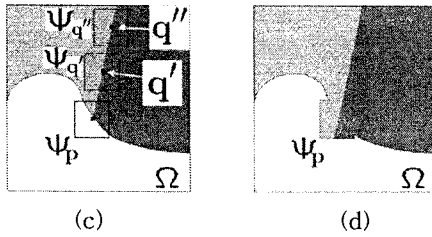


그림 1. inpainting processing

우선 그림 1에서 다음과 같이 정의한다.

Ω : 제거할 영역(Target Region)

$\delta\Omega$: Ω 의 경계

Φ : 제거할 영역의 외부(Source Region)

p : $\delta\Omega$ 위의 픽셀

q : Φ 위의 픽셀

Ψ_p : p 를 중심의 일정크기 표본

Ψ_q : q 를 중심의 일정크기 표본

그림 1의 (b)와 같이 제거할 영역의 경계에서 일정 크기의 표본을 우선순위에 따라 선택하고 그림 1의 (c)와 같이 제거할 외부영역 표본과 비교하여 최소 오차를 갖는 표본을 검색하고 제거할 영역만을 채운다.

2.2.1 표본의 우선순위 결정

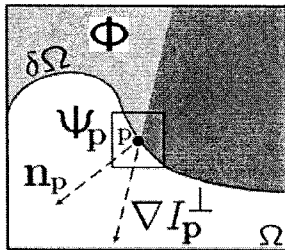


그림 2. inpainting priority

그림 2와 같이 $\delta\Omega$ 위의 점 p 에서 Ψ_p 가 주어졌을 때 우선 순위 $P(p)$ 는 다음과 같은 두 함수의 곱으로 표현된다.

$$P(p) = C(p)D(p)$$

여기서 $C(p)$ 는 표본 Ψ_p 안에서 Φ 영역에 속하는 픽셀들의 포함비율이며 $D(p)$ 는 p 에서의 등조선(isophote)의 강도를 의미한다.

$$C(p) = \frac{\sum C(q)}{|\Psi_p|} \quad q \in \Psi_p \cap \bar{\Omega}$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}$$

그림 2에서와 같이 ∇I_p^\perp 는 등조선의 방향, n_p 는 점 p 에서 $\delta\Omega$ 에 대하여 법선벡터이며 α 는 정규화 파라미터 (Grey-level 영상에서 α 는 255)이다.

2.2.2 표본 검색

$\delta\Omega$ 위의 모든 점 p 에서 우선순위가 결정되면 우선순위가 가장 높은 Ψ_p 을 찾아서 영역 Φ 에서 Ψ_p 와 가장 비슷한 Ψ_q 을 검색한다.

$$\Psi_q = \arg \min d(\Psi_p, \Psi_q) \quad \Psi_q \in \Phi \quad (5)$$

여기서 두 표본 Ψ_a, Ψ_b 의 거리 $d(\Psi_a, \Psi_b)$ 는 이미 채워져 있는 픽셀들의 SSD(sum of squared differences)로 정의한다. 식 (4)에서 얻어진 Ψ_q 은 Ψ_p 에 대응하는 픽셀로 채운다.(단 Ψ_p 에서 Ω 에 속하는 픽셀만 해당 된다.)

픽셀에 값을 채운 후 2.2.1과 2.2.2과정을 반복하여 영역 Ω 는 줄어들게 한다.[3]

III. 실험 결과

모든 실험은 Intel Celeron CPU 2.40Ghz, 128MB memory, window XP 환경에서 실험하였다.

제안한 mean-shift algorithm은 PC window 환경에서 Visual C++를 이용하여 구성되었다. 구성된 사용자 인터페이스에서는 초기 객체를 설정하기 위한 도구를 제공하고, 추적과정에서 매 프레임의 화면을 보여준다.

본 연구는 동영상에서 훼손된 영역이나 제거하고자 하는 영역을 자동적으로 제거 복원하는 것에 목적을 두고 있다. 이러한 목적을 이루기 위해 가장 중요한 사항은 전 frame 영역에서 처리하기 위해 움직임 검출 및 추적기법과 정확한 제거 복원을 위한 algorithm이다.

위와 같은 문제점을 해결하는 수단으로 움직임 검출 및 추적기법은 mean-shift algorithm을 사용하였고 제거 복원을 위한 algorithm은 텍스처 병합 기반의 image inpainting 방법으로 처리하였다.

제안된 모델의 성능과 타당성을 확인하기 위한 실험을 위해 다음과 같은 순서와 내용으로 실험 결과를 비교 고찰하였다.

- 1) 움직임 검출 및 추적 실험을 위하여 본 연구에서 사용하는 mean-shift algorithm 사용하여 오차비율에 대하여 실험하였다.
- 2) 움직임 검출 및 추적이 끝난 영상에서 제거 혹은 복원하고자 하는 영역을 지정 처리하기 위해 image inpainting algorithm을 사용하였다.

3.1 움직임 검출 및 추적 실험

표1은 25fps의 avi 동영상에서 제안된 방법을 이용한 추적의 개략적인 정확성을 보여주고 있다. 표에서 객체비율이란 처음 frame에서 마운딩 박스의 객체를 포함하는 매크로 블록과 객체 이외의 매크로 블록의 수를 표현하며

이들의 비율은 변화지 않거나 오차비율이 +값을 가져야 정확하다고 간주한 것이다. 여기서 사용되어진 매크로 블록은 5*5이다. 실제로 히스토그램값을 가지고 처리하기 때문에 이 비율은 추적이 성공적이라도 변할 수 있다. 이러한 점을 감안할 때 본 연구에서 제안된 객체추적은 성공적으로 시행되고 있다고 판단된다.



(a)Original (b)Median (c)Inpainting

그림 5. 640*480 Pixel image

IV. 결론

본 연구에서는 동영상에서 훼손된 영역이나 제거하고자 하는 영역을 자동적으로 제거 복원 시키는 방법으로 개선된 mean-shift algorithm을 이용한 image inpainting 방식을 제안한다. 기존의 동영상 편집 기술은 각각의 frame을 수동적으로 처리하여 많은 비용과 시간이 소모된다. 본 연구에서는 제거, 복원을 원하는 객체를 첫 frame에서 선택하여 mean-shift algorithm을 사용하고 나머지 frame에서 선택한 영역을 자동적으로 추출하여 제거 또는 복원을 한다. 제안한 방법을 여러 동영상에 적용한 결과 다음과 같은 결론을 얻었다.

1. mean-shift algorithm으로 각각 frame내의 선택 대상 객체를 보다 효율적으로 추출된다.
2. 영상내의 객체를 제거 혹은 복원하기 위해 텍스처 병합 기반의 image inpainting을 수행하였다.

위와 같은 특징의 제안 모델은 동영상 편집에서 시간과 비용을 절감할 수 있을 것이다. 아울러 다양한 영상 처리에 많은 발전을 가져올 것으로 기대된다.

향후 연구 방향으로서는 제안한 알고리즘의 수행 속도를 개선하여 실시간 동영상에 대해서도 image inpainting이 가능하도록 지속적인 연구가 수행되어야 할 것으로 사료된다.

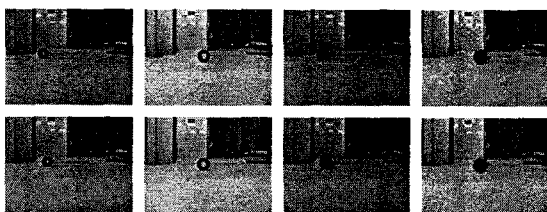
참고 문헌

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester. "Image Inpainting." Proceedings of SIGGRAPH 2000, New Orleans, USA, July 2000.
 [2] T. Chan and J. Shen, "Mathematical models for local deterministic inpaintings," UCLA CAM TR 00-11, March 2000.
 [3] T. Chan and J. Shen, "Local inpainting models and TV inpainting," SIAM J. Appl. Math. 62:3, pp. 1019-1043, 2001.
 [4] A. Zorzo Barcelos, A. Batista, M. Martins and Antonio Calos Nogueira "Level Lines Continuation based Digital Inpainting", SIBGRAP'04
 [5] 장득열, 채경덕, 조석수, 이경봉 "재료역학", 오토테크 (2005)

		1 frame	2 frame	3 frame	4 frame	평균 오차
움직이는 객체	객체 비율	14/14	14/14	16/14	15/14	
	오차 비율		0 (0%)	+0.7 (+7%)	+0.3 (+7%)	7%
정지한 객체	객체 비율	14/14	14/14	15/14	14/14	
	오차 비율		0 (0%)	+0.3 (+7%)	0 (0%)	3.5%

표 1. Results of mean-shift algorithm

그림 4는 표1의 테스트의 비디오 스트림으로 움직이는 객체와 정지한 객체를 추적하는 장면이다. 그림 4의 (a)그림과 (b)그림은 원영상이고 (c)그림과 (d)그림은 결과 영상이다.



(a)움직이는 객체 (b)정지한 객체 (c)움직이는 객체 (d)정지한 객체

그림 4. Result of mean-shift algorithm

3.2 Image inpainting 처리 실험

기존의 미디언 필터, image inpainting 두 방식에 대해 그림5와 같은 개별 frame을 이용하여 실험하였다. 우선 그림 5(a)는 640*480인 원 영상이며, (b)는 원 영상에 대하여 미디언 필터링을 100회 시킨 결과이며 (c)는 image inpainting을 수행한 결과이다. 두 필터링 방식의 결과를 시각적으로 확연히 차이가 나는 것을 알 수 있다.

