불필요한 이미지 정보 제거와 마스크를 이용한 빈 공간의 색상 보간에 관한 연구

박현국, 이강만
동국대학교 멀티미디어공학과
e-mail: dltlrlwkd@dongguk.edu, gangman@mme.dongguk.edu

A Study on the Removal of Unnecessary Image Information and a Mask to Interpolate the Color of the Empty Space

Hyunkook Park, Gangman Yi
Dept of Multimedia Engineering, Dongguk University, Seoul, 04620, Korea

요 약

사진 촬영이나 영상에서의 단일이미지를 저장할 때, 결과이미지에 사용자가 중점을 둔 이미지 정보를 훼손하는 불필요한 장애물이 존재하는 경우가 있다. 본 논문에서는 훼손된 이미지의 정보를 복원하기 위해 불필요한 부분을 표시하여 제거하고, 그 결과 빈 이미지 공간을 근접한 색상 정보를 이용해서 보간하여 손상된 이미지를 최대한 복원할 수 있도록 구현했다.

1. 서론

사진을 촬영하고 영상에서의 단일이미지를 저장하는 경우, 원하는 이미지의 정보를 가리거나, 방해하는 장애물이 존재할 수 있다. 울타리나 쇠창살로 배경이 가려지는 경우와 비, 눈으로 이미지의 정보가 훼손되는 경우를 예로 들 수 있다. 이러한 장애물은 사진을 촬영하는 각도 또는 방향을 바꾸는 것으로 해결되지 않는다. 따라서 위의 문제를 해결하고자 본 논문에서는 이미지 내의 방해되는 장애물을 제거하고 색상 보간을 수행하여 원하는 이미지를 얻는 방법에 대해 소개하고자 한다.

먼저 사용자가 직접 지우고자 하는 부분을 표시하여 제거와 보간이 이루어질 공간의 정확성을 높인다. 그 이후 사용자가 불필요하다고 판단하여 표시한 이미지 정보를 제거하며, 비어있는 공간을 주변 정보를 이용하여 색상을 채우는 기능을 수행한다. 이를 통해서 사진을 촬영할 때, 이미지에 장애물의 반영이 가능한 방식을 제시한다.

2. 시스템 흐름

본 논문의 연구는 C++와 Open Source Computer Vision Library (OpenCV)를 사용하며 요구되는 개발 환경은 다음과 같다<표 1>.

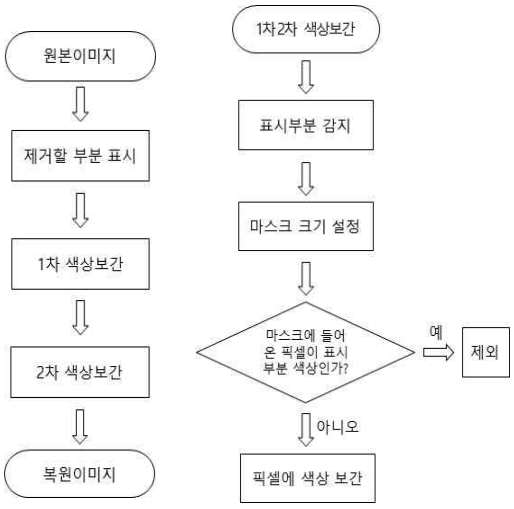
"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었음"(2016-0-00017)

<표 1> 개발 환경

구 분	사 양
운영시스템	Windows 10 Education / 64bit
통합개발환경	Visual Studio 2017
개발언어	C++
라이브러리	Open Source Computer Vision Library (OpenCV)

시스템의 흐름은 먼저 촬영된 이미지에서 불필요한 부분을 표시한다. 그 다음 1차 색상 보간을 통해 불필요한 부분의 중심부가 지워지고, 주변 색으로 지워진 공간을 보간한다. 2차 색상 보간은 1차 보간 후, 다 지워지지 않은 불필요한 부분을 전부 지우고, 지워진 공간을 근접한 색으로 보간한다. 즉 1차 보간을 통해서 불필요한 장애물의 중심부가 제거되어 색이 보간되고, 2차 보간으로 중심부 외의 경계선 부분까지 전체 색상 복원이 완료된다. 실험의 최종결과물로 불필요한 장애물이 제거되고, 제거된 부분의 색상이 복원된 이미지가 나온다. 이를 도식화한 시스템의 흐름도는 (그림 2)이며, 실험에 사용한 원본이미지는 (그림 1)과 같다. 본 논문의 연구에서 초점을 두어 복원하려는 이미지 정보는 마우스 거치대로 가려진 (그림 1)의 분홍색의 보온병이며, 보온병을 가리는 마우스 거치대를 제거하고자 하는 방해물로 설정했다.

(그림 1) 원본이미지



(그림 2) 시스템 흐름도

2.1 표시

먼저 촬영된 이미지에서 지우고자 하는 부분을 그림판 또는 photoshop을 이용해서 표시(mark)를 한다. 본 연구에서는 보온병을 가리는 마우스 거치대를 방해물로 설정해서 보라색(R:255 G:0 B:92)으로 표시하였고, 보라색으로 가려진 보온병의 부분에 복원이 진행된다. 물병의 색과 배경에 대해 복원할 수 있는지 확인하고자 이미지를 사용하였다. 원본 이미지에서 물병만 가리는 일부분만 표시한 이미지(그림 3), 물병을 가리는 장애물과 배경까지 가리는 전체 부분을 표시한 이미지(그림 4).

(그림 3) 원본이미지에서 지우려는 부분 표시(일부분)

그림 4) 원본이미지에 지우려는 부분 표시(전체)

2.2 1차 색상 복원

입력된 이미지는 보라색으로 지우려는 부분(mark area)이 표시되어 있고 일차적으로 표시된 중심부를 위주로 색상 복원을 진행한다. 보라색으로 표시된 이미지를 기준으로 모든 픽셀(pixel)에 접근하여, 새로운 출력 이미지 공간을 생성하고, 생성된 빈 이미지에 픽셀을 옮기는 방식으로 구현했다.

2.2.1 표시부분 감지

전체 픽셀(pixel)을 처리할 필요가 없이, 표시부분(mark area)을 감지한다. 표시부분은 (그림 2)와 (그림 3)에서 보라색으로 채색된 부분을 의미한다. 표시가 되지 않은 부분은 기존의 이미지의 색을 그대로 가져온다. 이후 모든 작업은 표시부분에서만 진행된다.

2.2.2 마스크 크기 결정

이미지에 대한 처리는 표시부분(mark area)에서 픽셀단위로 이루어진다. 마스크(mask)는 하나의 픽셀을 기준으로 근방의 픽셀들의 범위를 의미한다. 마스크의 크기(mask size)는 결정된 마스크의 범위이다. 마스크를 이용하면, 각 픽셀마다 보간(interpolation)할 근접한 색상들의 범위를 결정할 수 있다. 본 논문에서는 복원이 이루어질 물병의 형태를 생각하며, 마스크의 크기를 설정했다.

2.2.3 마스크 색상 제어

복원될 픽셀에 표시부분(mark area)의 색상인 보라색이 포함되면 결과이미지의 복원이 제대로 이루어지지 않는다. 따라서 마스크에는 보간에 사용할 물병의 색만 포함되며, 표시부분의 색상은 제외한다.

2.2.4 표시부분 색상 보간

마스크를 통해 불필요한 색상을 제외하고, 근접한 색상으로 픽셀(pixel)을 보간한다. 표시부분(mark area)의 중심부 위주로 진행된 1차 색상 복원 작업의 결과는 (그림 5) (그림 6)으로 확인할 수 있다.

(그림 5) 1차 복원 후 결과이미지(일부분)

(그림 6) 1차 복원 후 결과이미지(전체)

2.3 2차 색상 복원

원본이미지에서 불필요한 부분을 표시(mark)하면, 표시부분(mark area)의 정중앙은 보라색의 RGB값(R:255 G:0 B:92)을 그대로 유지한다. 하지만 색을 칠하면서 물병과 색이 혼합되며 표시부분의 경계선은 RGB값이 변하게 된다. 이 결과는 (그림 5) (그림 6)에서 경계선은 색이 복원되지 않은 것을 확인할 수 있다. 이를 해결하기 위해 이차적으로 경계선을 중심으로 색상 복원이 이루어져야 한다.

2.3.1 경계부분 감지

경계부분(boundary area)은 표시부분(mark area)이 1차 색상 복원 후에 경계선에만 남아있는 보라색 부분을 의미한다. 2차 색상 복원에서는 이 경계부분의 색상만 처리한다. 그 외에 다른 부분은 1차 복원을 완료한 후의 색상을 그대로 가져온다.

2.3.2 마스크 크기 결정

1차 색상 복원이 완료되면, 경계부분(boundary area) 외에 모든 픽셀에 색상 보간이 완료된 것이다. 따라서 이미 경계부분 근방에는 보라색보다 복원된 물병의 색이 더 많기 때문에 마스크의 크기를 크게 하지 않아도 된다.

2.3.3 마스크 색상 제어

복원될 픽셀에 경계부분(boundary area)의 색상인 보라색이 포함되면, 결과이미지의 복원이 제대로 이루어지지 않는다. 따라서 보간될 색을 가진 마스크에는 경계부분의 색상은 제외시킨다.

2.3.4 경계부분 색상 보간

마스크를 통해 불필요한 색상을 제외하고, 근접한 색상으로 경계부분(boundary area)의 픽셀을 보간한다. 즉 (그림 5) (그림 6)에서 경계부분의 색인 보라색을 근접한 물병의 색으로 채우는 것이다. 최종 작업 결과물은 (그림 7) (그림 8)로 확인된다. (그림 7)을 통해 구현된 프로그램이 물병만 가리는 장애물을 제거하고 물병의 색을 복원하는 것을 확인할 수 있다. (그림 8)은 물병의 색과 지워진 배경도 복원하는 것을 확인할 수 있다.

(그림 7) 2차 복원 완료 이미지(일부분)

(그림 8) 2차 복원 완료 이미지(전체)

3. 최종 결과

본 논문에서 구현한 프로그램은 불필요한 부분이 표시된(mark) 이미지를 입력물로 받는다. 1차 색상 복원으로 표시부분(mark area)의 중심부가 근방의 색으로 채워지고, 처리되지 못한 경계부분(boundary area)은 2차 색상 복원으로 색이 보간된다.

4. 결론

현 구현 단계는 단일이미지에 대한 후처리로 이루어지지만, 더 나아가 실시간으로 이미지 처리가 이루어지는 시뮬레이터와 복잡한 구조로 이루어진 장애물도 제거하며 이미지를 복원할 수 있을 것이라 기대한다.

참고문헌

[1] Open Source Computer Vision(OpenCV) https://docs.opencv.org/3.4.0/index.html

[2] Xue T, Rubinstein M, Liu C, Freeman WT (2015) A computational approach for obstruction-free photography.