|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **과목명** | **영상신호처리** | **학기** | **2023 - 1** | **담당교수** | **장주용** |
| **학과** | **전자통신공학과** | **학번** | **2018707072** | **이름** | **이승룡** |
| **과제 2 - gray world assumption** | | | | | |

**1. 과제설명**

white balance with gray world assumption 알고리즘을 “wb\_grayworld.m”에 구현한다.

“hw2.m” 스크립트가 실행되었을 때 아래와 같은 그림이 출력되어야 한다. white balancing을 위한 라이브러리를 활용하지 말아야 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. 이론**

white balance with gray world assumption는 이미지의 평균 색상이 회색이라는 가정을 사용하여 이미지의 색 온도를 보정하는 데 사용되는 기술이다.

gray world assumption은 대부분의 자연 이미지에서 평균 색상이 중성 회색에 가깝도록 색상이 분포된다는 생각을 기반으로 한다. 이 가정은 평균값이 동일하도록 이미지의 색상 채널을 조정하여 색온도를 수정하도록 한다.

white balancing을 위한 gray world assumption을 적용하기 위해 먼저 이미지의 색상 채널을 빨강, 녹색 및 파랑 요소로 분리한다. 그런 다음 이러한 각 구성 요소의 평균 색상 값을 계산하고 평균 회색 값(빨강, 녹색, 파랑 채널 모두의 평균 값)을 해당 구성 요소의 평균 색상 값으로 나누어 각 구성 요소에 대한 scaling factor를 계산한다.

마지막으로 이미지는 각각의 scaling factor로 각 색상 채널을 조정한다. 결과적으로 각 색상 채널의 평균 색상 값이 동일해지고 이미지가 중성 또는 회색으로 나타난다.

gray world assumption을 사용하는 이점은 많은 자연 이미지에 잘 작동하는 간단하고 효과적인 기술이기 때문이다. 그러나 색상이 고르게 분포되지 않거나 조명 조건에서 지배적인 색상 경향이 있는 장면에서는 제대로 작동하지 않을 수 있다.

**3. 소스코드 설명**

|  |
| --- |
| % Calculate the mean values of the red, green, and blue channels  Red\_avg = mean(mean(im(:,:,1)));  Green\_avg = mean(mean(im(:,:,2)));  Blue\_avg = mean(mean(im(:,:,3))); |

red, green, blue 채널의 평균값을 mean() 함수를 사용하여 계산한다.

|  |
| --- |
| % Calculate the average gray value of the image  Gray = (Red\_avg + Green\_avg + Blue\_avg) / 3; |

평균 회색 값(빨강, 녹색, 파랑 채널들의 평균의 평균 값)을 구한다.

|  |
| --- |
| % Compute the scaling factors for each channel  Red\_scale = Gray / Red\_avg;  Green\_scale = Gray / Green\_avg;  Blue\_scale = Gray / Blue\_avg; |

위에서 구한 평균 회색 값에 각 채널의 평균을 나누어 채널 별 scaling factor를 구한다.

각 채널에 대해 별도의 scaling factor를 계산함으로써 각 채널의 강도를 독립적으로 조정하여 이미지에서 보다 균형 잡힌 색상 분포를 얻을 수 있다. 이 접근 방식을 사용하면 단순히 모든 채널을 동일한 비율로 조정하는 것보다 더 좋은 결과를 얻을 수 있다.

|  |
| --- |
| % Apply the scaling factors to the image  im\_out(:,:,1) = im(:,:,1) \* Red\_scale;  im\_out(:,:,2) = im(:,:,2) \* Green\_scale;  im\_out(:,:,3) = im(:,:,3) \* Blue\_scale; |

Scaling factor를 입력 이미지의 각 채널에 곱해서 출력 이미지를 구한다.

**4. 실행결과**

웹사이트이(가) 표시된 사진

자동 생성된 설명

출력 이미지가 입력 이미지에 비해 색상이 더 자연스럽고 눈에 띄었던 주황색이 사라진 것을 확인할 수 있다.

**5. 전체 소스코드**

|  |
| --- |
| function [im\_out] = wb\_grayworld(im)  im\_out = im;  % Calculate the mean values of the red, green, and blue channels  Red\_avg = mean(mean(im(:,:,1)));  Green\_avg = mean(mean(im(:,:,2)));  Blue\_avg = mean(mean(im(:,:,3)));  % Calculate the average gray value of the image  Gray = (Red\_avg + Green\_avg + Blue\_avg) / 3;  % Compute the scaling factors for each channel  Red\_scale = Gray / Red\_avg;  Green\_scale = Gray / Green\_avg;  Blue\_scale = Gray / Blue\_avg;  % Apply the scaling factors to the image  im\_out(:,:,1) = im(:,:,1) \* Red\_scale;  im\_out(:,:,2) = im(:,:,2) \* Green\_scale;  im\_out(:,:,3) = im(:,:,3) \* Blue\_scale; |