

Image Processing

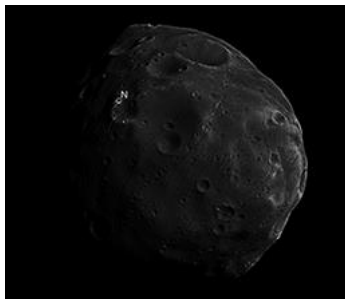
Assignment #3

Dongwann Kang

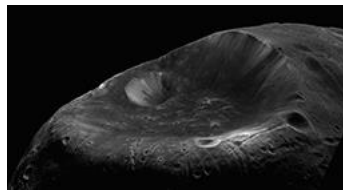
Assistant Professor
Department of Computer Science and Engineering

Histogram equalisation & matching 구현

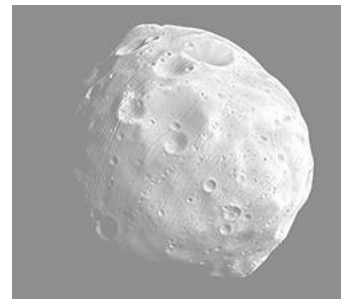
- 입력 이미지
 - Phobos1.png
- 타겟 이미지
 - Phobos2.png
- 출력1 (Histogram equalisation)
 - 입력 이미지에 histogram equalisation을 수행한 결과 이미지
- 출력2 (Histogram matching)
 - 입력 이미지가 타겟 이미지의 히스토그램을 갖도록 histogram matching을 수행한 결과 이미지



입력 이미지



타겟 이미지



histogram equalisation



histogram matching

Histogram equalisation & matching 구현

- 주의사항
 - Histogram equalisation과 histogram matching 각각의 프로그램을 제출
 - Histogram matching의 중간 과정에서 출력 1(histogram equalisation 결과)을 생성할 수 있으므로, 하나의 프로그램에서 출력1과 출력2가 모두 생성되도록 만들 수 있으나... 그렇게 하지 말 것!
 - 하나의 프로그램으로 합치지 말고, “출력1이 생성되는 프로그램1”과 “출력2가 생성되는 프로그램2”를 각각 만들어 제출할 것
 - 두 프로그램간 코드가 중복되어도 전혀 상관없음
 - 공통적인 기능들을 함수로 만들었다면, 두 프로그램이 그 함수를 공유하여도 됨
 - Histogram matching의 최종 결과 이미지를 생성할 때, Equalised image에서 transformation (mapping)을 수행하지 말고, “입력 이미지”에서 transformation (mapping)을 수행할 것
 - 이론 시간에 배웠듯이, histogram matching의 중간 과정에서 equalised image를 생성하는 과정은 생략될 수 있음
 - 이 것이 프로그램을 합치지 말고 각각의 프로그램을 만들라고 했던 이유임
 - 즉, histogram matching 프로그램은 histogram equalisation 알고리즘은 사용하되, 중간 이미지(equalised image)를 생성하지는 말아야 함
 - 반드시 주어진 이미지(Phobos1.png & Phobos2.png) 사용

Histogram equalisation & matching 구현

- 주의사항
 - OpenCV 수업시간에서 배운 것 이외의 OpenCV 함수 사용 금지
 - OpenCV가 Histogram과 관련된 함수들을 제공하나, 그것을 “사용”하는 것이 아니라 “직접 구현”하는 것이 본 과제의 의의임
 - 수업시간에 배운 histogram equalisation & matching 이론을 따라 구현할 것
 - Histogram의 구현
 - 히스토그램은 단순히 배열을 통해 구현 가능 (e.g., `int histogram[256];`)
 - 히스토그램을 화면에 출력할 필요 없음!
 - 따라서 OpenCV가 제공하는 histogram 관련 함수들을 사용해야 할 이유가 전혀 없음
 - 과제 수행에 어려움이 있을 시, 이메일 또는 eclass 통한 문의 환영
- 제출물
 - E-class [과제]에 아래 문서 업로드
 - 1) histogram equalisation 소스코드 (C++)
 - 2) histogram matching 소스코드 (C++)
 - 3) 각각에 대한 간략한 코드 설명과 결과 이미지를 포함한 보고서 (표지없이 3-5 페이지)
- 제출 기한
 - 6월 30일 23시 59분까지