

Luminosity and contrast normalization in retinal images

Marco Foracchia , Enrico Grisan, Alfredo Ruggeri

Medical Image Analysis, 2004

논문을 구현하고 나서 데이터를 해석해야 하는데, 이때 생긴 의문이 있어 정리를 다시 해본다. 의문은 구현된 알고리즘의 결과로 부터 나오는 데이터가 가우시안 분포를 따르는가? 이다. 왜 이런 의문이 들었냐면, 결과로써 얻어진 이미지의 히스토그램이 마치 가우시안 분포와 같아 보이기 때문이다.

지금 내가 사용하고 있는 이미지는 준혁이로 부터 받은 functional Slit lamp 의 conjunctival image 이다. 하드웨어가 빛번짐을 완벽히 잡아주지 못하여 얻어진 image는 non-uniform한 밝기를 가져 이는 일종의 noise 역할을 하여 image processing에 방해요소가 된다. 따라서 나는 image 전반에 퍼져있는 이러한 non-uniform 정도를 줄여 uniform 한 밝기를 가지는 image를 얻고자 한다.

이전에는 **median filter**를 사용(filter의 크기가 매우 큰) 하여 background image를 추출해내어 이를 원본 이미지와 빼기 연산을 해 uniform한 이미지를 얻었다. 하지만 non-linear한 median filter의 특성상 소실되는 정보도 존재하고 정확도 높은 background image를 추출하지 못하기 때문에 만족스러운 결과를 얻지 못하였다.

따라서 correction 하는 다른 논문을 찾아보았는데 retinal image에 적용한 것중 인용수가 가장 높아 선택하였다...(너무 단순하게 선택한 감이 없잖아 있다.)

Key Idea

해당 시스템의 image

- 1500 * 1700 pixel images.
- green ch
- normalized image [0,1], mean luminosity in this range [0.128, 0.508]

우리가 보는 Image는 background image와 foreground image(보고자 하는 부분)로 이루어져 있다.

$$I = f(I') = f(I_{back} + I_{fore})$$

I' 는 original image, I는 우리가 획득한 fundus / conjunctival image가 된다. background는 우리가 보고자 하는 Vessel이 없는 부분 (이상적인) 을 말하고 foreground는 vascular structure, optic disc, lesion 등 의미 있는 부분을 말한다. 이들은 조금 추상적이다. (정의가 딱 되어있지는 않다.)

$$I_{back} \text{ 을 확률모델로써 보도록 하자. } I_{back}(x, y) \approx \mathcal{N}(\mu_{back}, \sigma_{back})$$

즉, 이상적인 경우에, background image의 분포는 Gaussian distribution을 따른다.

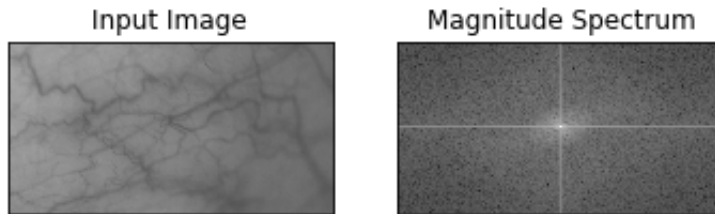
f 는 acquisition function 이다. 이에 대한 설명도 필요하다. (참고로 해당 모델은 blur / noise와 같은 것을 고려하지 않는다.) f 는 contrast / luminosity 를 말하게 된다. image안의 non-uniform한 contrast / luminosity는 다음과 같이 표현이 가능하다.

$$I(x, y) = f(I'(x, y)) = C(x, y)I'(x, y) + L(x, y)$$

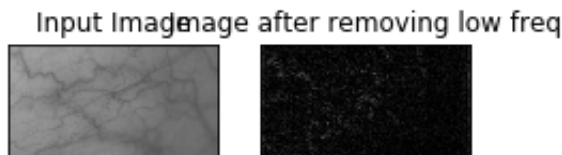
C = contrast, L = luminosity (어째서 저런 식이 나오지는 잘 모르겠다.) 추정 image를 구하기 위한 식인데 추정 image는 결과로써 나오는 image를 말하는 것 같다.

$$\hat{I}'(x, y) = \frac{I(x, y) - L'(x, y)}{C'(x, y)}$$

I', L', C' 은 추정된 image이다. 음... observed image인 I 와 L', C'를 가지고 I' 을 추정, 결과를 구하는 시스템인데, **여하튼 Image를 복원하는게 목표이다.** 자, 그럼 C와 L을 어떻게 추정 할 것인가. C와 L은 image의 low frequency 부분에 집중되어 있다.



```
(Text(0.5,1,'Image after removing low freq'),
([], <a list of 0 Text xticklabel objects>),
([], <a list of 0 Text yticklabel objects>))
```



< Figure 1. image low frequency 이해를 돕기 위함 > Figure 1을 보면 **luminosity와 contrast가 low frequency 영역에 집중적으로 분포되었음**을 알 수 있다. (밑의 그림은 low frequency 영역을 제거한 그림이다.) 그리고 우리는 또 다른 식을 얻을 수 있다.

$$I(x, y) = C(x, y)I'_{back} + C(x, y)I'_{fore} + L(x, y)$$

$$using, I = f(I'_{back} + I'_{fore})$$

하지만 우리가 **얻고자 하는것은 background image** 이므로 foreground를 빼면

$$I(x, y) = C(x, y)I'_{back} + L(x, y)$$

해당 식이 나오게 되고, **background image를 추정**하게 된다. 추정된 background image는 다음의 확률분포를 따르게 된다.

$$I_{back}(x, y) \approx \mathcal{N}(L(x, y), C(x, y)), (x, y) \in \mathcal{B}$$

즉, 우리는 background image의 **L과 C를 이용하여 background image를 추정**해 낼 것이다. 정확히는 L과 C로 image를 복원 한다. 수식을 좀 깔끔히 정리를 해보자면

$$\hat{I}'(x, y) = \frac{I(x, y) - L'(x, y)}{C'(x, y)}$$

$$I(x, y) = C(x, y)I'_{back} + C(x, y)I'_{fore} + L(x, y)$$

$$I(x, y) = C(x, y)I'_{back} + L(x, y)$$

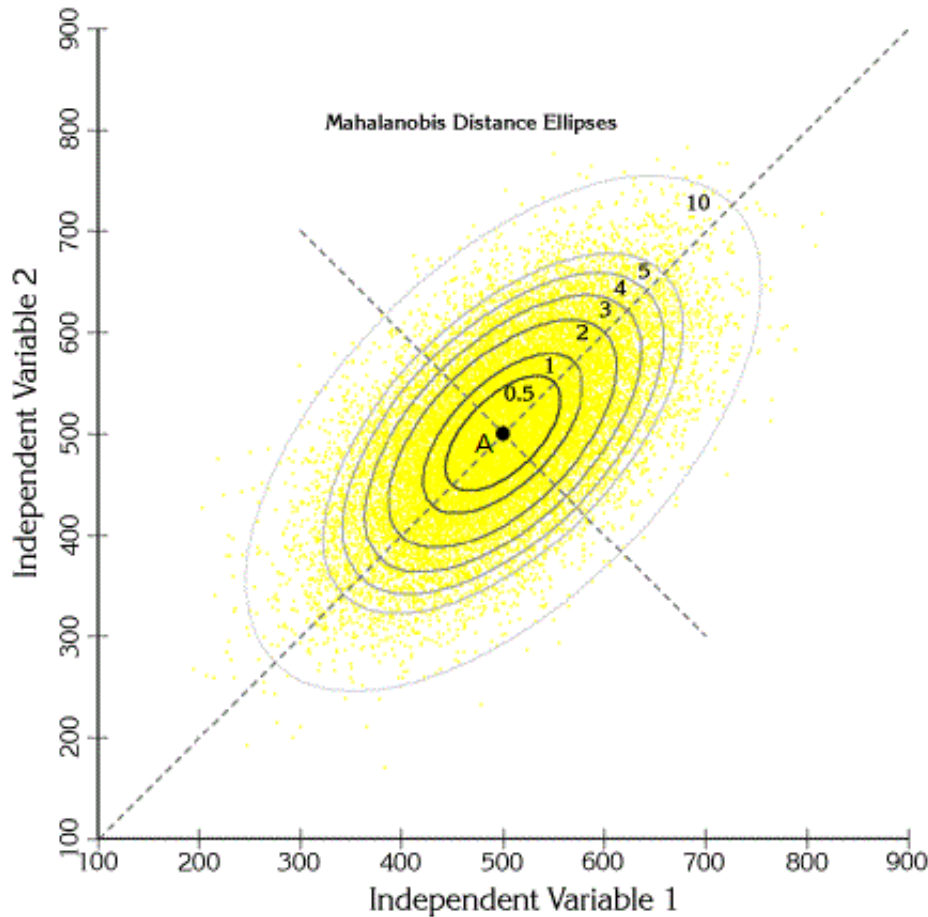
직관적인 key idea를 정리해보자. contrast(물체를 다른 물체와 구별하게 만들어주는 값, 시각적인 차이)

luminosity(단위 면적당 빛의 양, 여기서는 pixel의 intensity에 비례할듯) 를 이용하여 background image를 추정한다.

용어 / 가정

먼저 해당 논문에서 마할라노비스 거리 Mahalanobis distance 라는 용어를 사용한다. 이는 확률분포 상의 거리를 의미한다. (일종의 Norm) 데이터가 평균으로 부터 떨어진 정도가 표준편차에 얼마나 비례하느냐를 나타낸 거리다. 좌표계를 확률분포의 중앙으로 끌어왔다고 생각하면 될 것 같다.

$$d_m(X, Y) = \sqrt{(X - Y)^T * S * (X - Y)}, S = \text{Covariance Matrix}, Y = \text{Mean}(X)$$



< Figure 2. Mahalanobis distance, 확률분포상의 거리 >

그리고 몇가지 가정을 하고 시작한다.

1. L과 C는 상수이다.
2. 전체 이미지에서 50%는 배경이미지다.
3. 배경이미지와 원하는 이미지의 pixel intensity는 뚜렷히 차이가 난다.

그리고 연산시 Global 한 pixel 연산을 하게 되면, 연산량이 너무 많아지므로 local한 방법 사용하자. local 한 방법에서 full image를 얻을 때는, bicubic interpolation 사용하여 얻는다. (from sub sampled images.)

Global 한 방법도 구현해보면 좋을 것 같다. (오래전 논문이므로) 여하튼 이들 논문에서는 window 's'를 정하는것이 중요한 과정이었고 이에 대한 방법은 따로 나오지 않았다. (경험적, 실험적으로 찾은 것 같다.) window 선택에 따라 결과값이 차이가 나므로, 사양이 좋아진 지금, Global 한 방식 구현해봅시다. window 내에 background 이미지만 존재 할 경우, underresimation 되는 문제가 발생

Algorithm

이는 코드를 보면서 이해하는게 빠를듯.