Backlight Compensation Using Retinex Algorithm

ISL Lab Seminar

Han-Sol Kang

INDEX

01 Retinex Theory

02 Retinex Algorithm

03 Experiment Results

04 Conclusion

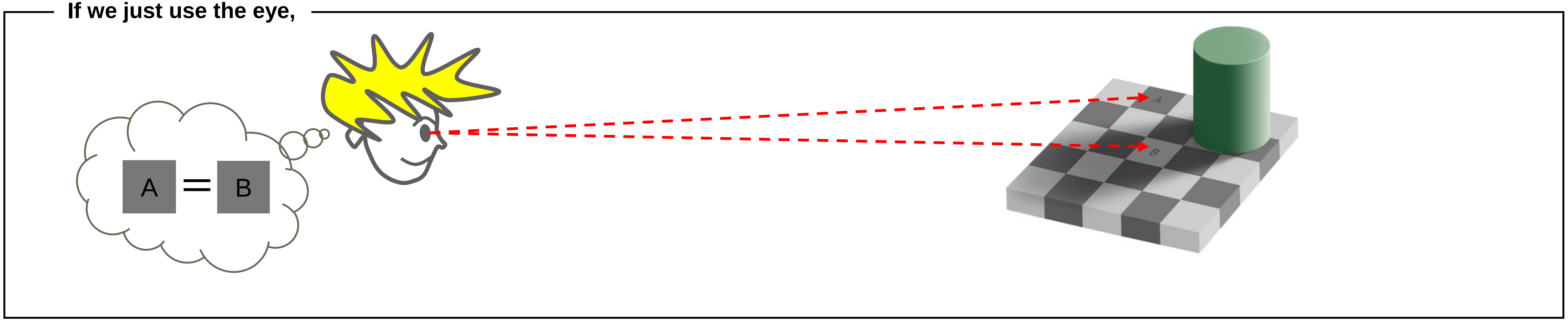
01 Retinex Theory*

∑ What's the Retinex Theory

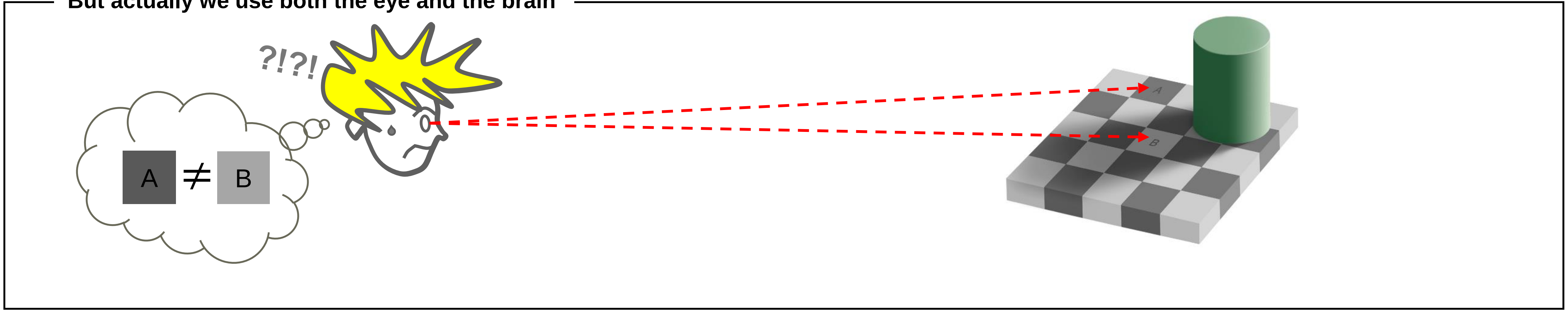
Retinex = Retina + Cortex

Both the eye and the brain are involved in the Image processing.

If we just use the eye,



But actually we use both the eye and the brain

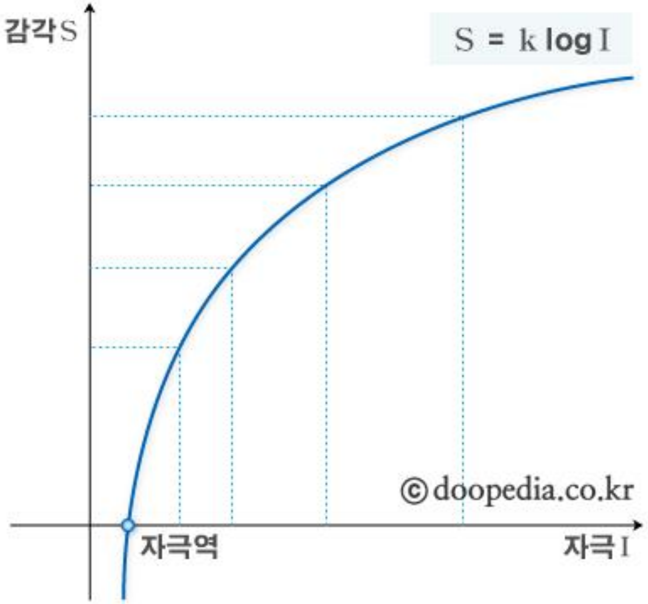


* Land, Edwin H., and John McCann. "Lightness and retinex theory." JOSA 61.1 (1971): 1-11.

02 Retinex Algorithm*

∑ Retinex Algorithm

Weber-Fechner's law



Land의 시각적 모델

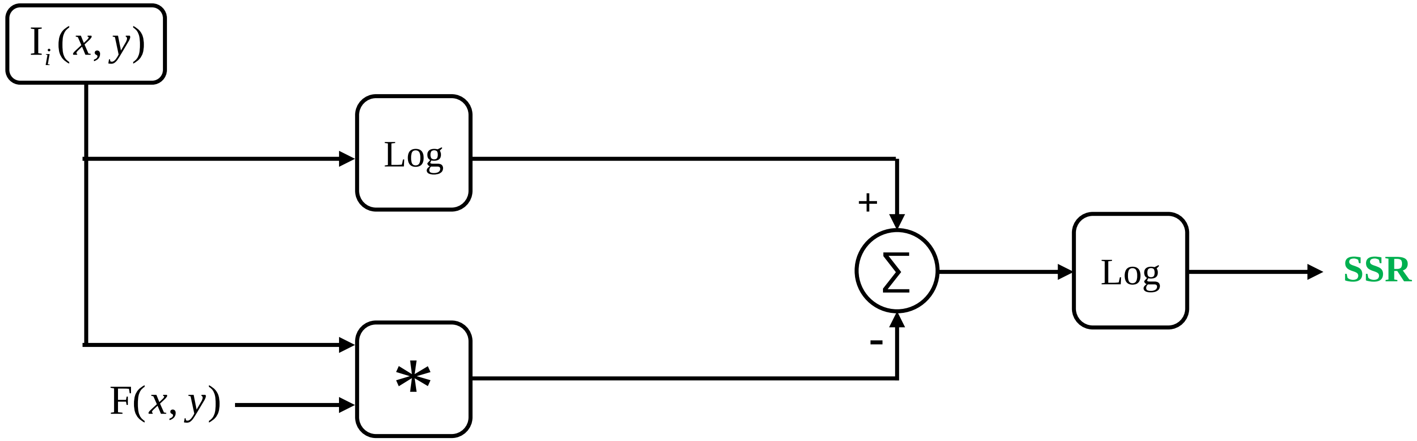
+

$$I_i(x,y) = R_i(x,y) \times L_i(x,y)$$

* Jobson, Daniel J., Zia-ur Rahman, and Glenn Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes." *Image Processing, IEEE Transactions on* 6.7 (1997): 965-976.

02 Retinex Algorithm[*]

∑ Single-Scale Retinex (SSR)



$$R_i = \log I_i(x,y) - \log(F(x,y) * I_i(x,y))$$

$I_i(x,y)$: i – th color ($i = 1,2,3$)

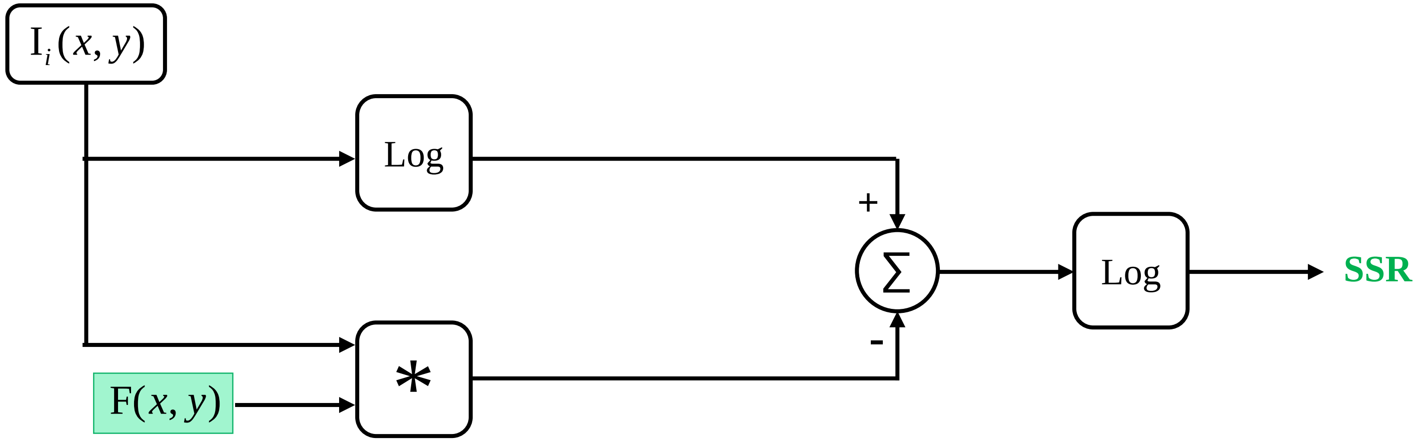
$F(x,y)$: Surround function

$R_i(x,y)$: Retinex output

[*] Jobson, Daniel J., Zia-ur Rahman, and Glenn Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes." *Image Processing, IEEE Transactions on* 6.7 (1997): 965-976.

02 Retinex Algorithm*

∑ Surround Function



- Surround function by E.Land'86 (Inverse square spatial surround)

$$F(x,y)=\frac{1}{1+(r^2+c^2)}$$

- Surround function by Hurlbert'89(Gaussian)

$$F(x,y)=e^{\frac{-r^2}{c^2}}$$

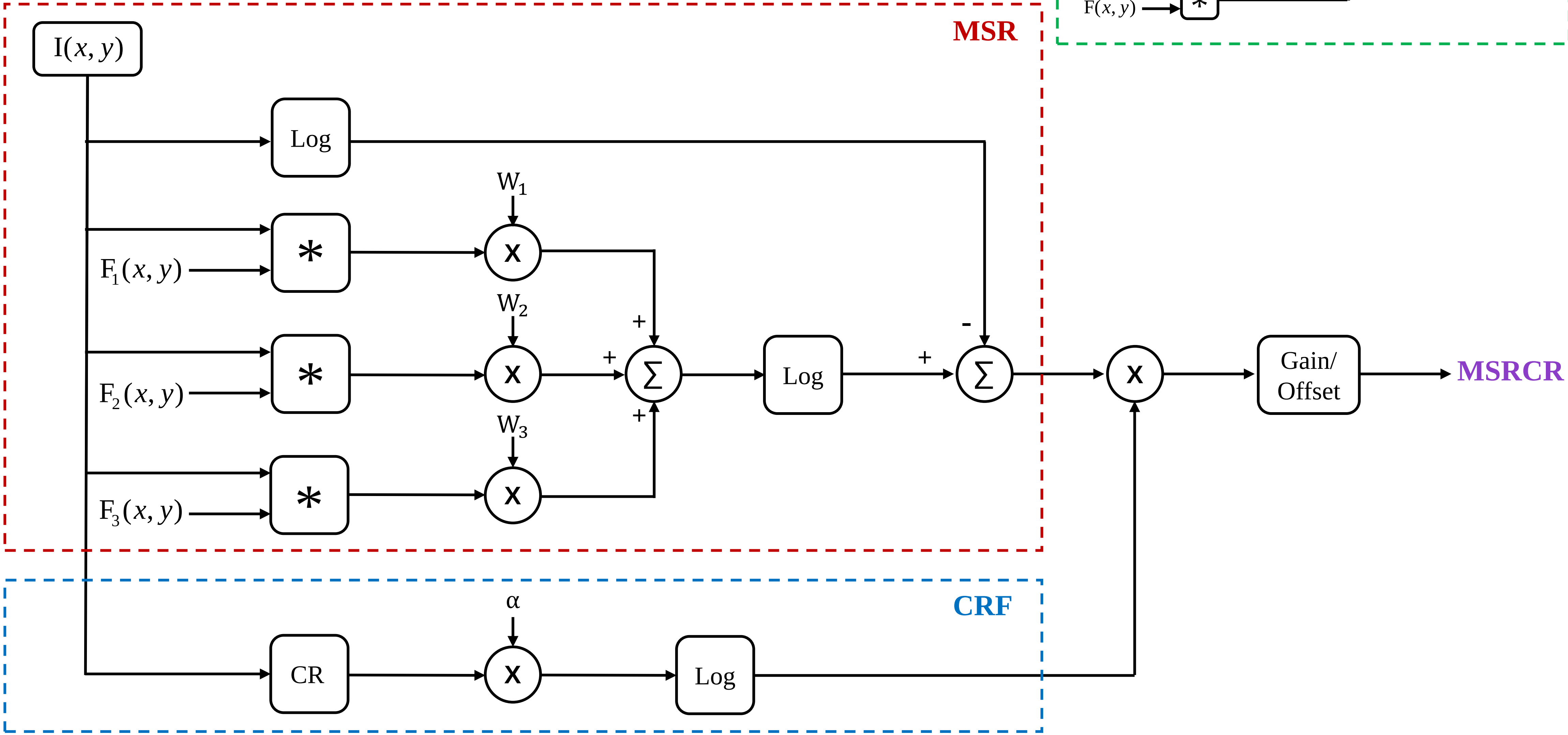
$$r=\sqrt{x^2+y^2}$$

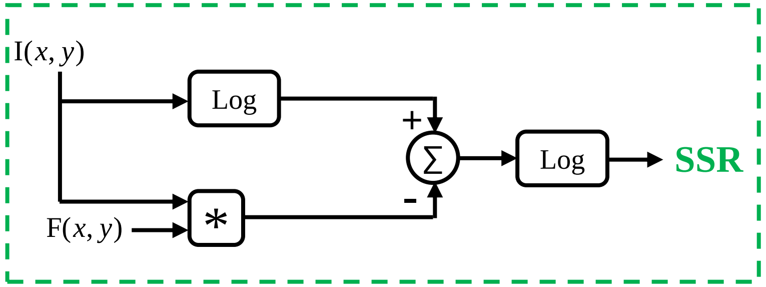
c : Surround space Constant

* Jobson, Daniel J., Zia-ur Rahman, and Glenn Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes." *Image Processing, IEEE Transactions on* 6.7 (1997): 965-976.

02 Retinex Algorithm

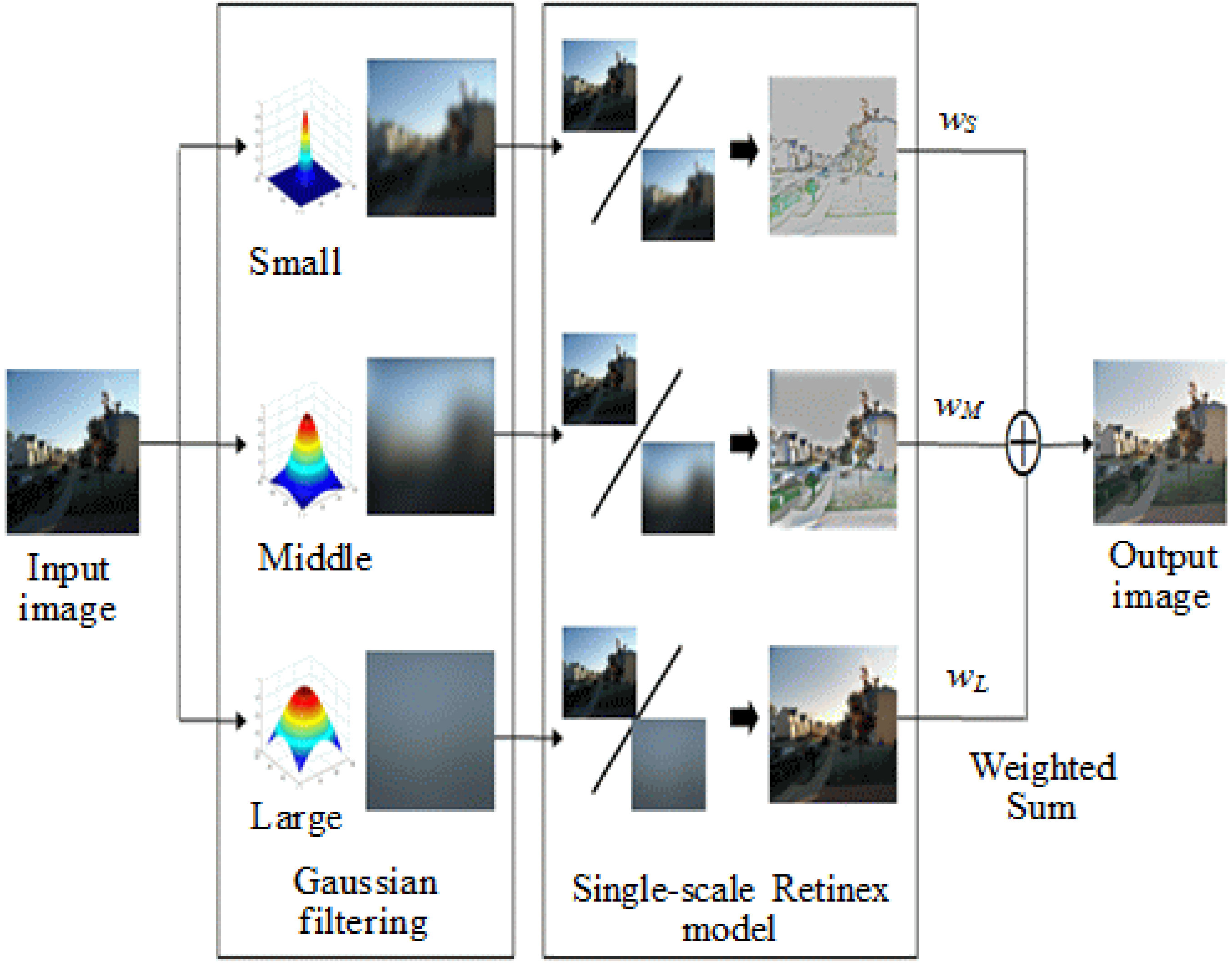
∑ **Multi-Scale Retinex (MSR) & MSR with Color Restoration (MSRCR)**





02 Retinex Algorithm

∑ **Multi-Scale Retinex (MSR) & MSR with Color Restoration (MSRCR)**



$$R_{MSR_i} = \sum_{n=1}^{N} W_n R_{n_i}, \qquad \sum_{n=1}^{N} W_n = 1$$

02 Retinex Algorithm

∑ MSRCR

$$C_i(x,y) = \beta\left\{\log[\alpha I_i(x,y)] - \log\left[\sum_{i=1}^{S} I_i(x,y)\right]\right\}$$

β : Gain Constant

α : Controls the strength of non - linearity

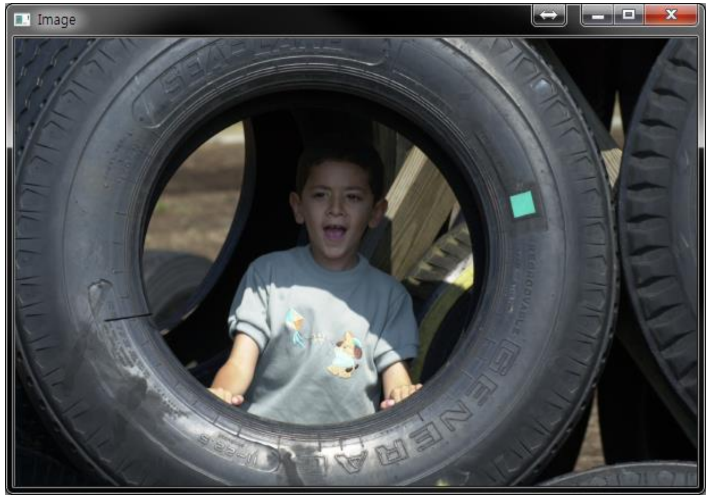
$$R_{MSRCR_i}(x,y) = G[C_i(x,y) * R_{MSR_i}(x,y) + b]$$

G : Gain Constant

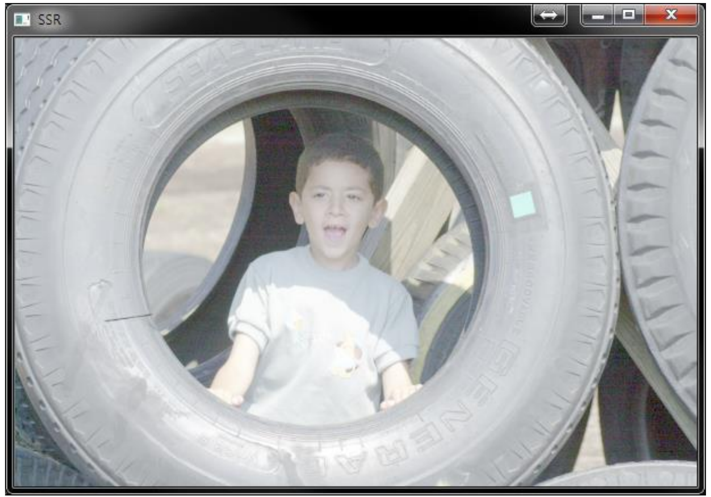
b : Gain Offset value

03 Experiment Results

∑ Backlight Compensation

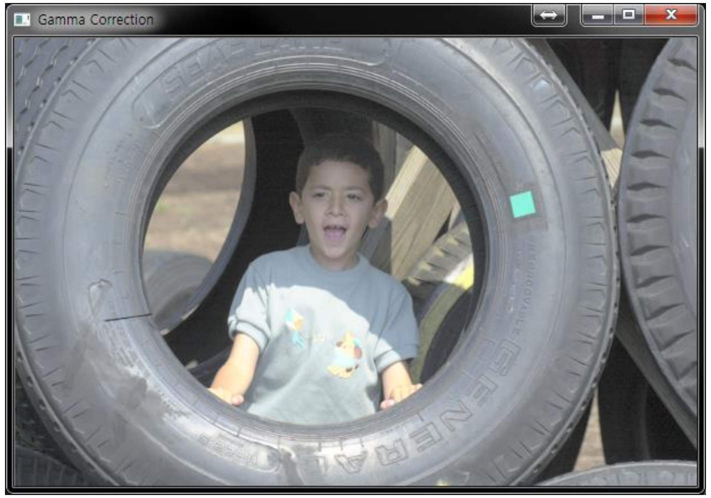


Origin



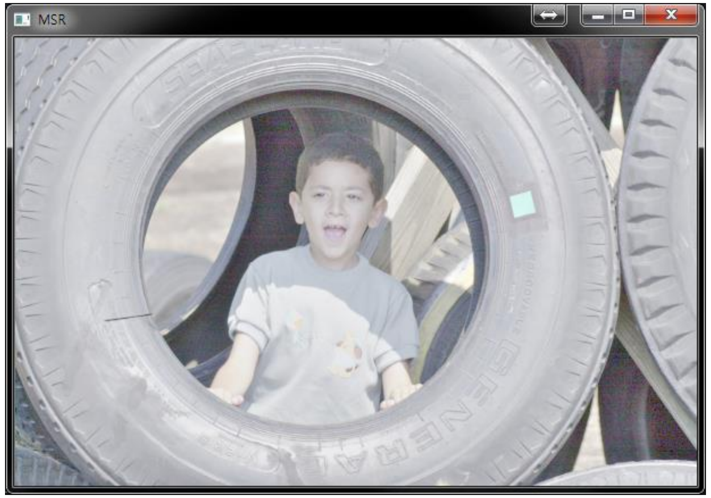
sigma = 250

SSR



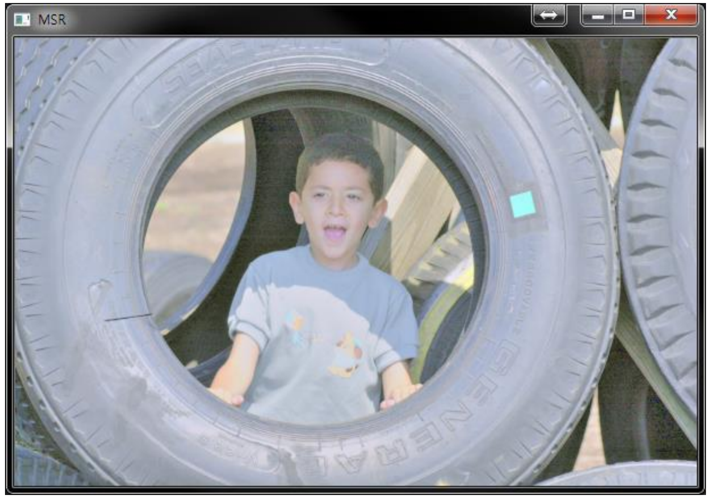
Gamma = 2.2

Gamma Correction



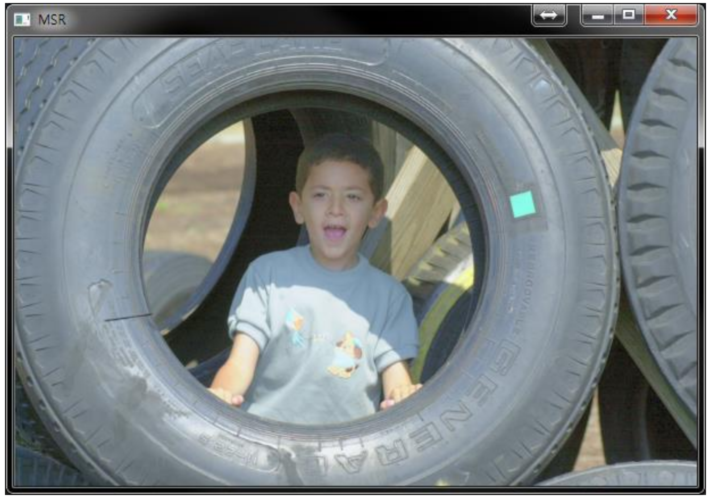
sigma =15, 80, 250

MSR



sigma =15, 80, 250
G = 130, b = 1

MSRCR

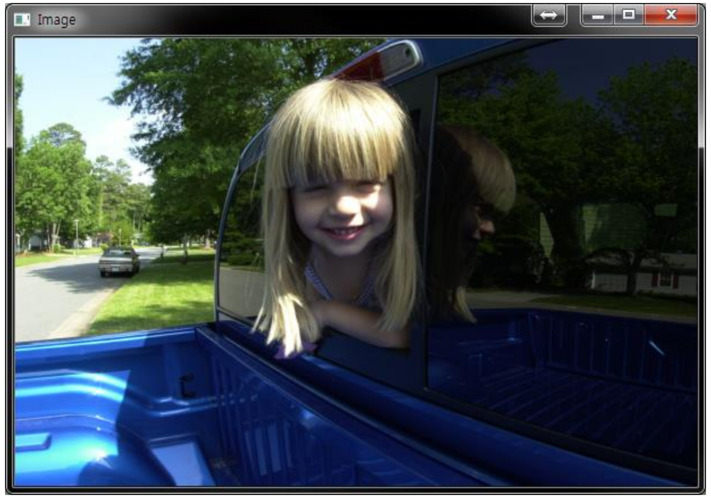


sigma =80, 250
G = 130, b = 1

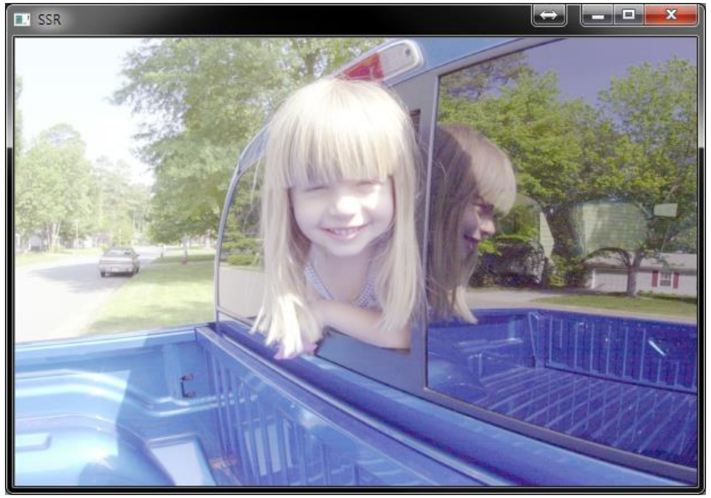
MMSRCR

03 Experiment Results

∑ Backlight Compensation

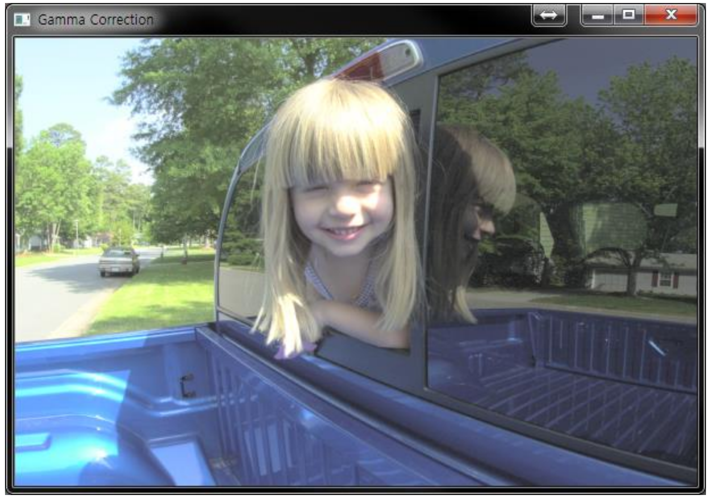


Origin



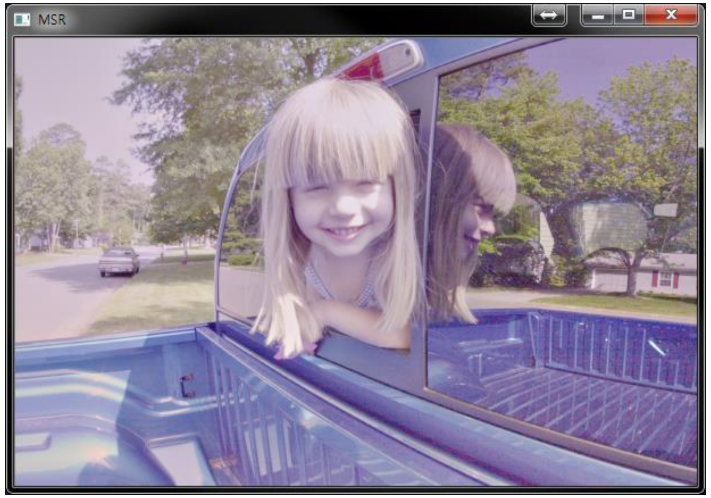
sigma = 250

SSR



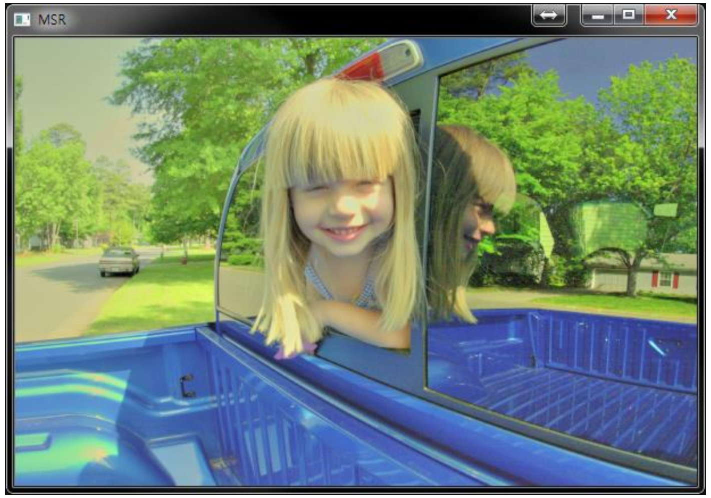
Gamma = 2.2

Gamma Correction

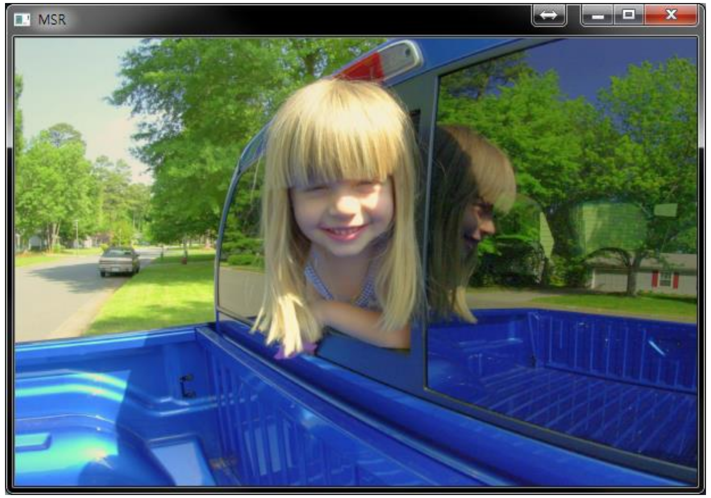


sigma =15, 80, 250

MSR



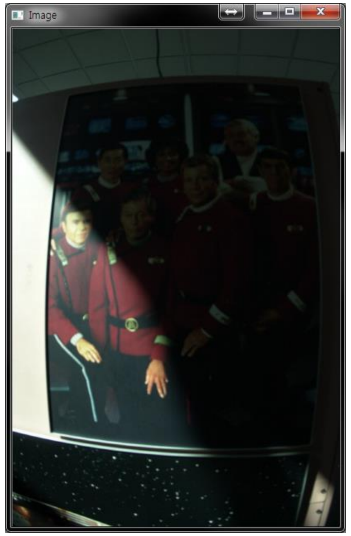
sigma =15, 80, 250
G = 130, b = 1
MSRCR



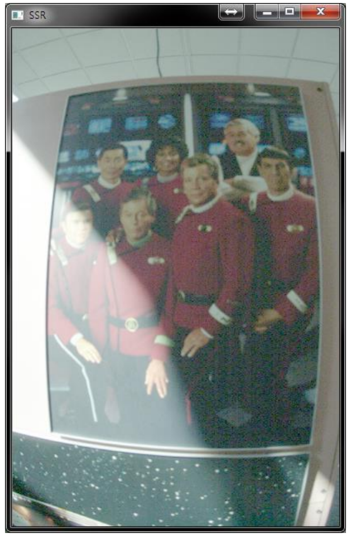
sigma =80, 250
G = 130, b = 1
MMSRCR

03 Experiment Results

∑ Backlight Compensation

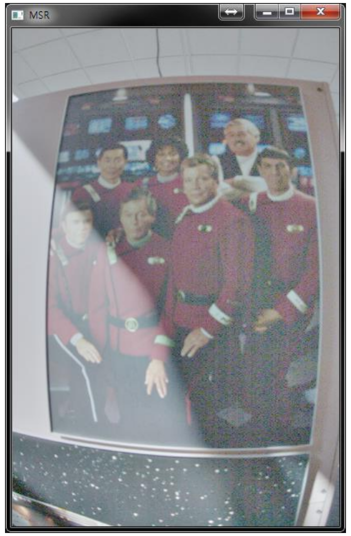


Origin



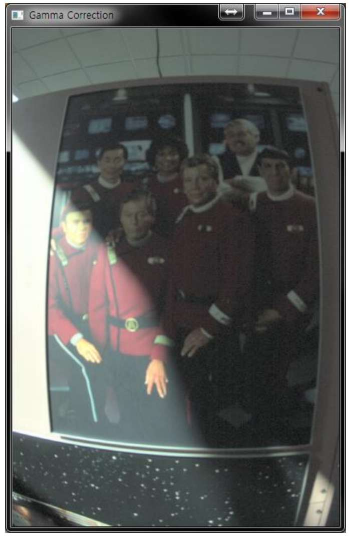
SSR

sigma = 250



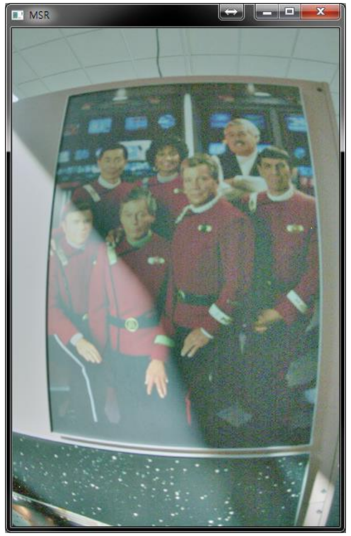
MSR

sigma =15, 80, 250



Gamma Correction

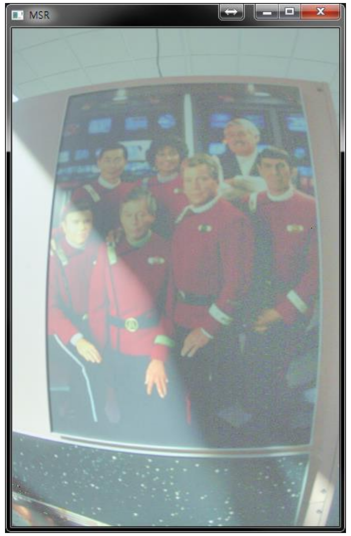
Gamma = 2.2



MSRCR

sigma =15, 80, 250

G = 200, b = -20, α=100, β=40



MMSRCR

sigma =80, 250

G = 200, b = 30

04 Conclusion

- ✓ SSR 보다는 MSR, MSRCR의 성능이 더 좋음을 확인.
- ✓ 처리 속도가 훨씬 오래 걸림.
- ✓ 실시간 적용은 힘들 것으로 예상.

http://dragon.larc.nasa.gov/retinex/pao/news/

Q & A