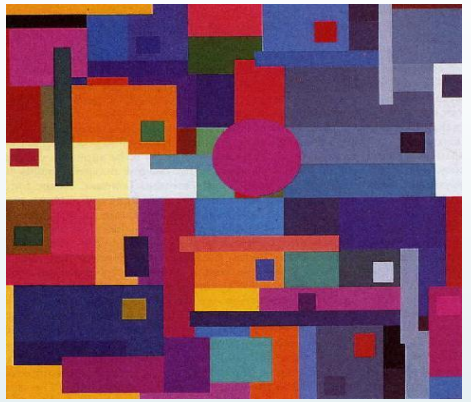
Fast RETINEX for Color Image Enhancement: Methods and Algorithms

Analysa M. Gonzales and Artyom M. Grigoryan

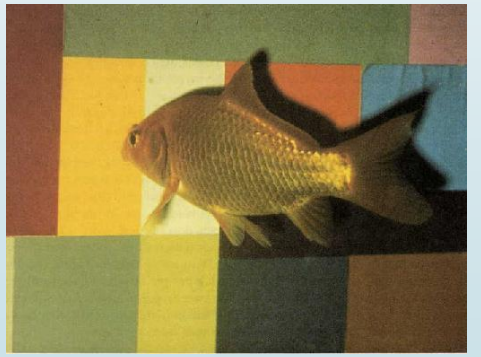
Department of Electrical and Computer Engineering, The University of Texas at San Antonio

Color Constancy

- Edwin Lands' experiments were designed to prove that despite variations (both large and small) in relative intensities of red, green, and blue illuminations upon a patchwork, the colors of each patch remain constant.

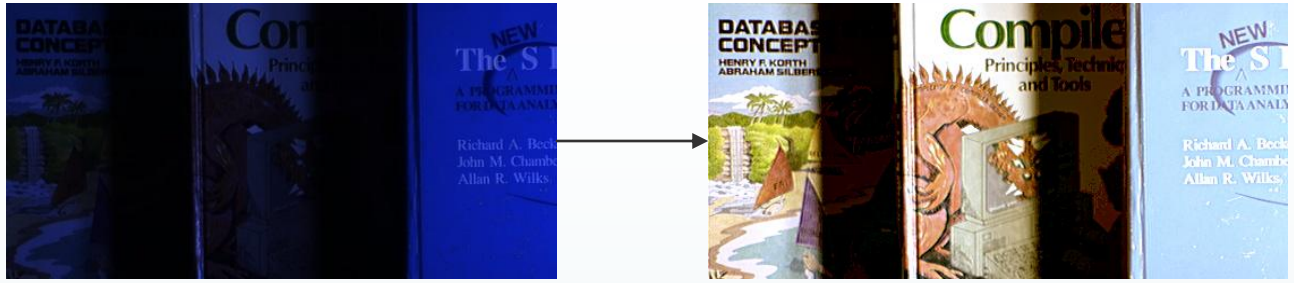


- David Ingle's Experiment: Despite changes in projector intensities used in to illuminate the patchwork, a goldfish was found to swim to the green patch every time regardless of the color of illumination being used, a behavior akin to the perceptual results in human found by Edwin Lands' experiments.



http://hubel.med.harvard.edu/book/b45.htm

RETINEX

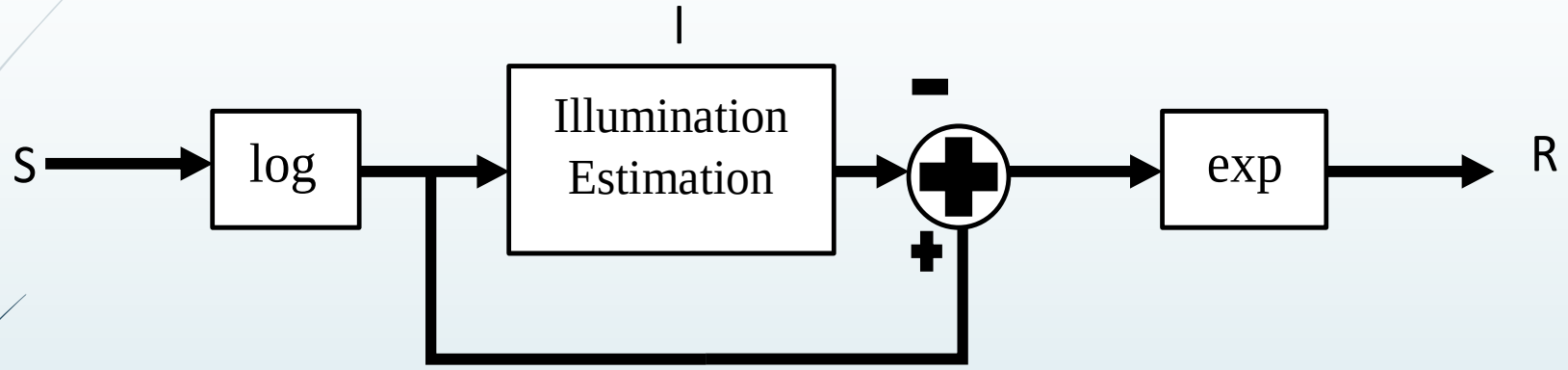


- Proposed by Edwin Land in 1986 and based upon his experimentation involving color constancy.
- A model of human perception of lightness and color which describes a method of image enhancement seeking to bridge the gap between digital images and those images observed by the human eyes.
- More specifically, an imaging process that offers a means of sharpening an image, improving its color consistency regardless of variations in illumination, and the means to achieve dynamic range compression.

Purpose of Presentation

- To give a broad overview of RETINEX including Single-Scale, Multi-Scale, and Multi-Scale with Color Restoration techniques.
- To show and compare various existing algorithms of RETINEX.
- To present a new and markedly more efficient algorithm which employs the implementation of the Fast Fourier Transform (FFT).
- To offer a new form of measure for the enhancement capabilities of any RETINEX algorithm.

General RETINEX Form



In this figure, $S(x, y)$ is the input color image to be enhanced, which is presented in the multiplicative form

$$S(x,y):\ \log(S(x,y)) = s \quad \log(R(x,y)) = r \quad \log(L(x,y)) = l$$

$$S(x,y) = R(x,y) \bullet L(x,y)$$

$$\log(S(x,y)) = \log(R(x,y)) + \log(L(x,y)) \ \rightarrow \ R(x,y) = \exp[\ \log S(x,y) - \log L(x,y)\ )\ ]$$

(where S(x,y) = original image, R(x,y) = reflectance and L(x,y) = illumination)

Existing Methods

- Single Scale Retinex (SSR)
- Multi-Scale Retinex (MSR)
- Multi-Scale Retinex with Color Restoration (MSRCR)
  - NASA
  - "Image Magick" (Dr. Fred Weinhaus)
  - Gretinex (Fabien Pelisson)
  - McCann99

Single Scale RETINEX (SSR)

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

$$R_i(x,y) = log\left(\frac{I_i(x,y)}{F(x,y) * I_i(x,y)}\right) = log\left(\frac{I_i(x,y)}{\overline{I}_i(x,y)}\right)$$

- $I_i(x,y)$: the image distribution in the i[th] spectral band
  $R_i(x,y)$: Retinex output
  C: Gaussian surround space constant
  F(x,y): Gaussian function $F(x,y) = Ke^{-(x2+y2)/c2}$

- Serious tradeoff: while both color/lightness rendition (using a large scale or large C value) and dynamic range compression (using a small scale or small C) can both be achieved, SSR cannot do both simultaneously.

Multi-Scale RETINEX (MSR)

$$R_{MSRi} = \sum_{n-1}^{N} \omega_n \{log[I_i(x,y)] - log[F_n(x,y) * I_i(x,y)]\}$$

- N: number of scales
  ω_n : the weight associated with the nth scale
  empirical values: $N = 3$, $\omega_n = 1/3$, $C = 15$, 80 and 250 correspondingly for each scale in F_n.

- Notably better than SSR in terms of dynamic compression and color rendition in that both can be done simultaneously
- Method does not always do well with colored images due to tendency of MSR to desaturate colors in a given image.

Multi-Scale RETINEX with Color Restoration (MSRCR)

$$R_i(x_1, x_2) = \sum_{k=1}^{K} W_k(log\ I_i\ (x_1, x_2) - log[F_k(x_1, x_2) * I_i(x_1, x_2)]) \quad i = 1, \dots, N,$$

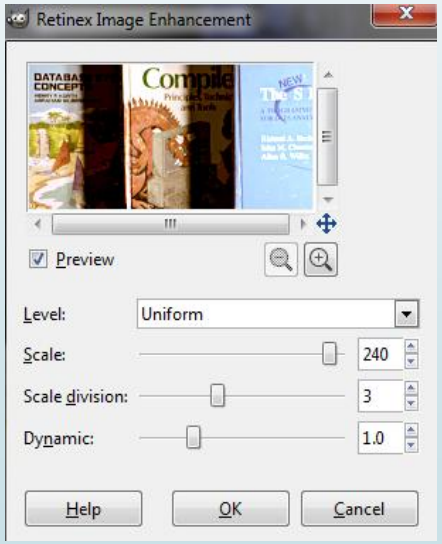
$$F_k(x_1, x_2) = k\ exp[-(x_1{}^2, x_2{}^2)/\sigma_k^2]$$

- *N: number of spectral bands*
  F_k: k^{th} surround function
  W_k: the weights

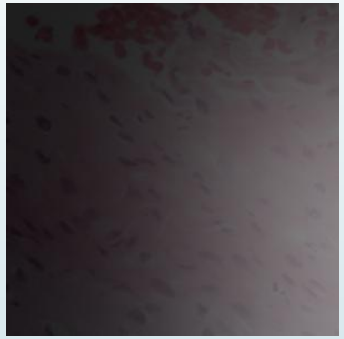
- To avoid desaturation problem of MSR, a post-processing technique for color restoration (CR) is applied in order to enhance local contrast.

GRETINEX using GIMP

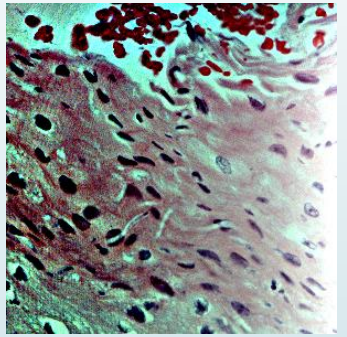
- Known bug: algorithm implemented its own linear Gaussian filter routine that shifted the results of the blurring to the right of the image when using more than three scales, the optimal number of scales used.



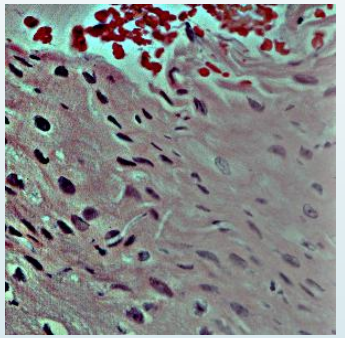
User Interface



Original



Uniform (1.2 dynamic)



Uniform (2.5 dynamic)

Original

Uniform (1.2 dynamic)

Uniform (2.5 dynamic)

GRETINEX Algorithm

- GRETINEX Algorithm – follows the general algorithm of MSR:

  1) Log of original image

  2) Estimation of the illumination using coefficients input by the user

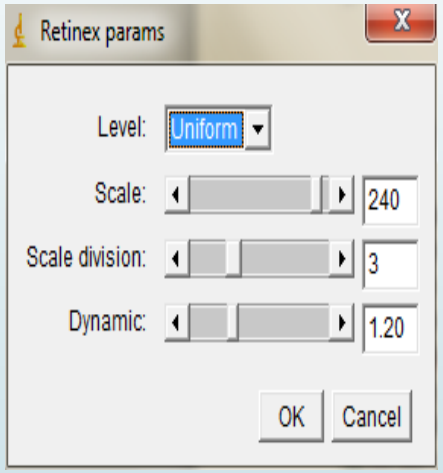
  3) Gaussian filter applied to blur image and remove details to make image enhancement in the log scale easier using a mirror method in both the X and Y directions

  4) Ratio calculated between original values obtained for the original image and filtered values resulting in the final image

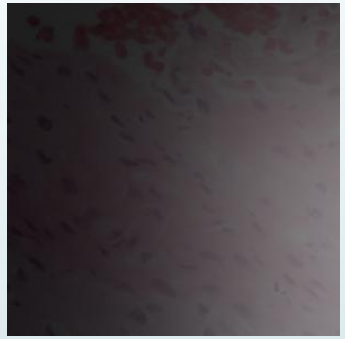
- Post Processing (Color Restoration):
  - Color values refocused based on the average color of the output of the Retinex algorithm.
  - The dynamic setting allows users to reduce or enlarge dispersion.

GRETINEX using ImageJ

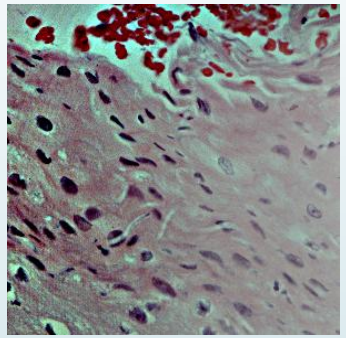
- Francisco Jiménez Hernández rewrote the plugin script in java code. Doing so enabled users to take advantage ImageJ's much more powerful Gaussian filter routine while achieving a much greater image processing speed.



User Interface



Original



Uniform (1.2 dynamic)

Original

Uniform (1.2 dynamic)

Overall, this method used the same overall structure of the Retinex algorithm, differing only in its blurring algorithm.

ImageMagickScripts – Dr. Weinhaus

- Modeled after NASA RETINEX implementation:
  - Averages the log of the ratio of the original image to each of three Gaussian blurred versions of the image (low, medium, high).
- Post Processing (CR): *color boost* image computed from the log ratio of original image to its grayscale version. The color boosted image and the image resulting from the MSR algorithm are then multiplied together.

Original

NASA

IMS-RGB

IMS-HSL

Comparison of Results

Original

NASA

IMS

GIMP

ImageJ

Comparison of Results

Original

NASA

IMS

GIMP

ImageJ

Comparison of Results

Original

NASA

IMS

GIMP

ImageJ

McCann99 RETINEX

- Designed by Brian Funt and Florian Ciurea to work in MATLAB.
- Program creates multi-resolution pyramid from input by averaging image data where the log of the image is averaged down to lowest resolution of pyramid and pixel comparisons are made in a recursive process at each level to obtain estimate lightness, where the initial lightness estimate for that level is that of the level preceding it.
  - Pixel lightness computed by taking the log of all 8 neighboring pixels in clockwise order and subtracting each of their luminance values. This difference is then added to the result of the image product.
- Variable parameters: only the number of iterations must be specified as it controls the amount of dynamic range compression.

Original

MATLAB Result

Comparison of Results

Original

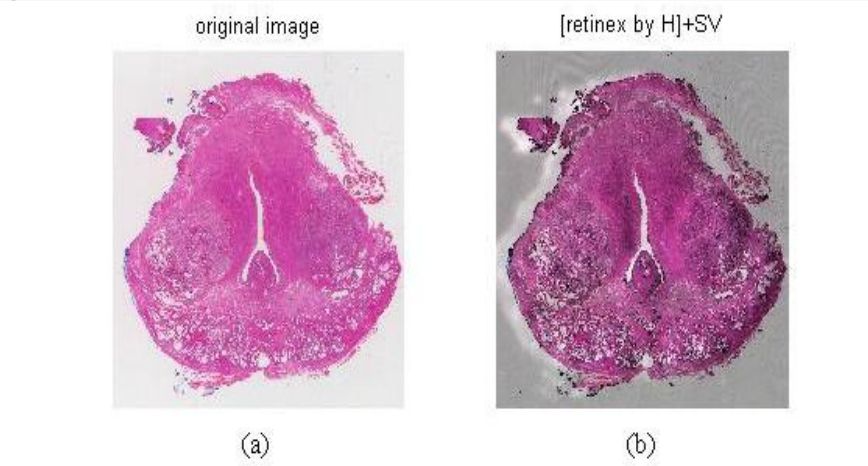
GIMP (Uniform)

ImageJ (Uniform)

RETINEX using Fast Fourier Transform (FFT)

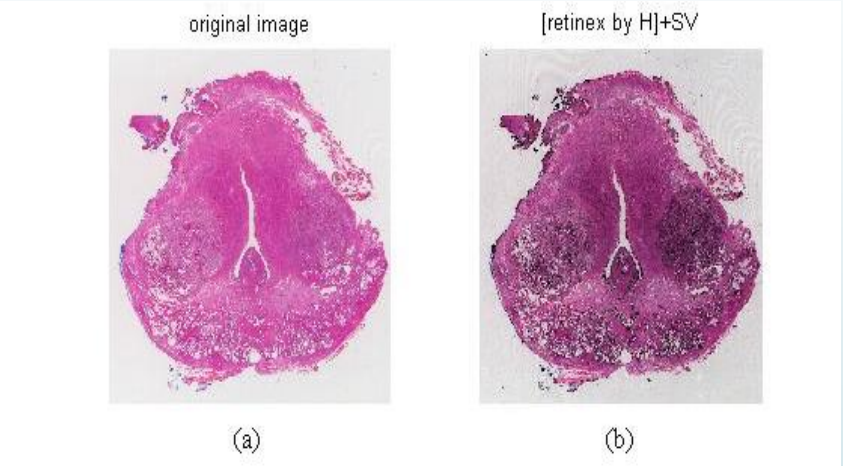
- Two major flaws in previously developed MSRCR algorithm:
  - Processing extremely slow
  - Enhanced output image itself had an embossed mask that made it 3-dimensional, giving the image properties the image itself did not have.
- "*imfilter*" replaced with the fft function so that Gaussian smoothing filter could be applied via multiplication (convolution in time domain maps to multiplication in frequency domain).
- Expected to achieve faster processing time since generally, *fft* is considered the fastest enhancement method in comparison to "*conv*" and "*imfilter*" operations.

Experimental Results



Original program using imfilter

Processing time: 35.6371s



Enhanced Program using FFT

Processing time: 8.0224s

New Measure: EMEC

- To calculate the color image enhancement, we can use the measure CEME (color image enhancement measure) introduced by Grigoryan and Agaian which generalizes the concept of the known EME concept for the gray-scale images.
- The quantitative measure of enhancement of the color image processed:

$$E_q(\alpha) = CEME_\alpha(\hat{f}) = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20\, log_{10} \left[\frac{max_{k,l}(\hat{f}_R, \hat{f}_G, \hat{f}_B)}{min_{k,l}(\hat{f}_R, \hat{f}_G, \hat{f}_B)} \right]$$

- $CEME_\alpha(\hat{f})$ is called *a measure of enhancement*, or *measure of improvement of the image* $f_{n,m}$. The "best" image enhancement parameter a is considered to be the one which maximizes the value of the $CEME$.
- For color image, we can apply the following measure of enhancement calculated on the image gradients:

$$\mathrm{CEME}(\hat{f}) = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \log_{10} \left[\frac{\max_{k,l} G_{x+y}(\hat{f})}{\min_{k,l} G_{x+y}(\hat{f})} \right]$$

Conclusion

- The Retinex algorithm taking advantage of the fft proved not only much faster than the original McCann algorithm used, but resulted in a truer image.
- All of the systems presented implement the Retinex algorithm in ways that vary mostly in terms of how the illumination in an image is estimated. While some offer more variability and control to the user, others are very strict with how their algorithm works.
- There are clear advantages and disadvantages to each of the systems, some of which are far more useful (or limiting in the ladder case) than others.
- Concluding which is the best is difficult since the purpose of Retinex is to mimic the radiance interpretation of the human eyes and comparing a momentary sensation of the human brain to the lasting Retinex images is an extremely difficult (and even more subjective) task.
- Further testing to better understand the scaling variables for each algorithm and use of some measure of noise such as SNR would likely allow for more conclusiveness.

Questions?