MA3111: Mathematical Image Processing
Image Contrast Enhancement

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 320317, Taiwan

First version: May 29, 2021/Last updated: December 3, 2024

Outline of "image contrast enhancement"

In this lecture, we will briefly introduce some techniques for image contrast enhancement, including

- *Histogram equalization (HE，直方圖等化)*
- *Automatic color equalization (ACE，自動色彩均衡)*
- *Simplest color balance (SCB，最簡色彩平衡)*
- *Variational methods with split Bregman iterations*

The material of this lecture

The material of this lecture is based on the following text and papers:

- *Section 3.3: Histogram Processing* in [GW2018], pp. 133-153.
- P. Getreuer, Automatic color enhancement (ACE) and its fast implementation, *Image Processing On Line,* 2 (2012), pp. 266-277.
- P.-W. Hsieh, P.-C. Shao, and S.-Y. Yang, Adaptive variational model for contrast enhancement of low-light images, *SIAM Journal on Imaging Sciences,* 13 (2020), pp. 1-28.
- N. Limare, J.-L. Lisani, J.-M. Morel, A. B. Petro, and C. Sbert, Simplest color balance, *Image Processing On Line,* 1 (2011), pp. 297-315.

Contrast enhancement

The main purpose of contrast enhancement is to adjust the image intensity to enhance the quality and features of the image for a better human visual perception or machine vision identification.

A low-light image and its enhanced result

Histogram equalization (HE): $g(x,y) =: s = T(r) := T(f(x,y))$

- We are given a grayscale image $f : \overline{\Omega} \to [0,1]$. The cumulative histogram *(cumulative distribution function)* T is defined by considering f as a random variable: for $\eta \in [0,1]$, we define

$$\begin{aligned} T(\eta) &:= Prob(f \leq \eta) \\ &= \frac{1}{|\overline{\Omega}|} \Big| \{(x,y) \in \overline{\Omega} : f(x,y) \leq \eta\} \Big|. \end{aligned}$$

Then $T : [0,1] \to [0,1]$ is a monotonic increasing function.

- The histogram equalized image $g : \overline{\Omega} \to [0,1]$ is obtained by defining

$$g(x,y) := T(f(x,y)).$$

Histogram equalized image $g \sim \mathcal{U}(0,1)$ if T is invertible

If T is strictly increasing, then T is invertible and the *cumulative distribution function* of the histogram equalized image g is

$$\begin{aligned} Prob(g \le \eta) &= Prob(T(f) \le \eta) = Prob(f \le T^{-1}(\eta)) \\ &= T(T^{-1}(\eta)) = \eta. \end{aligned}$$

Hence, the *probability density function* of g is

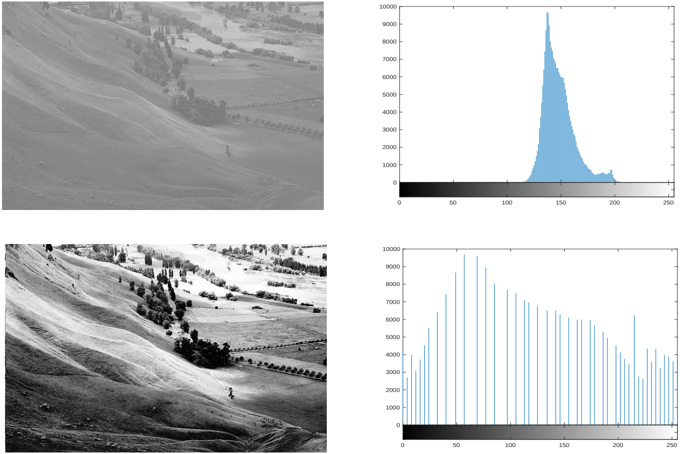
$$p(t) = \begin{cases} 1 & \text{for } 0 \le t \le 1, \\ 0 & \text{elsewhere.} \end{cases}$$

Therefore, g has a uniform distribution, i.e., $g \sim \mathcal{U}(0,1)$.

Remark: Let X be a random variable and $p(t)$ the probability density function (pdf) of X. The cumulative distribution function (cdf) of X is

$$F(\eta) := Prob(X \le \eta) = \int_{-\infty}^{\eta} p(t)\, dt.$$

Example of histogram equalized image



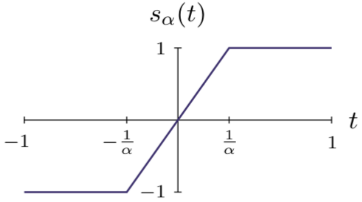
Histogram equalization of 400×600 *image: (top) before; (bottom) after; and the corresponding histograms*

Matlab commands: `imhist(A),histeq(A)`

Automatic color equalization (ACE)

We are given a grayscale image $f : \overline{\Omega} \to [0,1]$. First, the following operation is performed

$$\widetilde{f}(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \overline{\Omega} \setminus \{\boldsymbol{x}\}} \frac{s_\alpha(f(\boldsymbol{x}) - f(\boldsymbol{y}))}{\|\boldsymbol{x} - \boldsymbol{y}\|}, \quad \forall\, \boldsymbol{x} \in \overline{\Omega}.$$



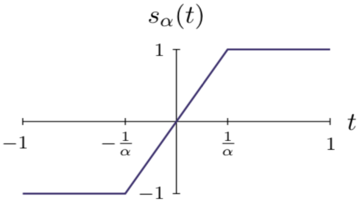
The slope function $s_\alpha(t) := \min\{\max\{\alpha t,\ -1\},\ 1\} \quad (\alpha > 1)$.

Then $\widetilde{f}$ is rescaled to $[0,1]$ as the ACE image

$$g(\boldsymbol{x}) = \frac{\widetilde{f}(\boldsymbol{x}) - \min \widetilde{f}}{\max \widetilde{f} - \min \widetilde{f}}, \quad \forall\, \boldsymbol{x} \in \overline{\Omega}.$$

ACE images for various α's and HE image

Input (352×480)



ACE, $\alpha = 2$

ACE, $\alpha = 4$

ACE, $\alpha = 8$

HE

Simplest color balance: given a grayscale image $f(x)$

- The algorithm simply stretches, as much as it can, the values of the three channels (R, G, B), so that they occupy the maximal possible range $[0, 255]$.
- The simplest way to do so is to apply an *affine function* $\tilde{z} = a \underbrace{z}_{f(\boldsymbol{x})} + b$ to each channel such that

$$\begin{cases} az_{\min} + b & = & 0, \\ az_{\max} + b & = & 255. \end{cases}$$

We solve a and b so that the maximal value in the channel becomes 255 and the minimal value 0.

$$a = \frac{255}{z_{\max} - z_{\min}}, \qquad b = -\frac{255 z_{\min}}{z_{\max} - z_{\min}}.$$

That is, the intensity of the resulting image is given by

$$\tilde{f}(\boldsymbol{x}) = \frac{255}{z_{\max} - z_{\min}} f(\boldsymbol{x}) - \frac{255 z_{\min}}{z_{\max} - z_{\min}} = 255 \left(\frac{f(\boldsymbol{x}) - z_{\min}}{z_{\max} - z_{\min}} \right), \boldsymbol{x} \in \overline{\Omega}.$$

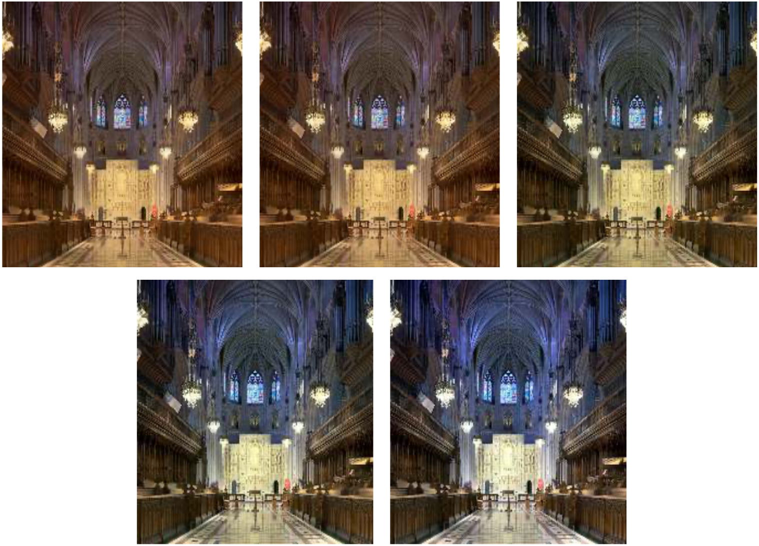
Simplest color balance (cont'd)

- However, many images contain a few aberrant pixels that already occupy the 0 and 255 values. Thus, an often spectacular image color improvement is obtained by *"clipping" a small percentage s% of the pixels with the highest values to* 255 *and a small percentage of the pixels with the lowest values to* 0*, before applying the affine transform.*
- Notice that this saturation can create flat white regions or flat black regions that may look unnatural. *Thus, the percentage of saturated pixels must be as small as possible.*
- In our numerical experiments of the proposed adaptive method below, we apply the simplest color balance (SCB) to the resulting images *with a* 0.1% *of saturation.*

SCB images

original image, SCB images with $s\% = 0\%, 1\%, 2\%,$ *and* 3%

SCB images



original image, SCB images with s% = 0%, 1%, 2%, and 3%

The proposed adaptive method with SCB and $s\% = 0.1\%$

A landscape of Da-Xi

A simple variational model

Let $f : \overline{\Omega} \to \mathbb{R}$ be a given grayscale image. The Morel-Petro-Sbert model *(IPOL 2014)* is given by

$$\min_u \underbrace{\frac{1}{2}\int_\Omega |\nabla u - \nabla f|^2 \, d\boldsymbol{x}}_{\textit{data fidelity}} + \underbrace{\frac{\lambda}{2}\int_\Omega (u - \overline{u})^2 \, d\boldsymbol{x}}_{\textit{regularizer}} .$$

- The constant $\overline{u} := \frac{1}{|\Omega|}\int_\Omega u \, d\boldsymbol{x}$ is the mean value of u over Ω.
- The data fidelity term preserves image details presented in f and the regularizer reduces the variance of u to eliminate the effect of nonuniform illumination.
- The parameter $\lambda > 0$ balances between detail preservation and variance reduction.

Two modified variational models

- The original model is simple but difficult to solve due to the $\overline{u}$ term. Therefore, by assuming that $\overline{u} \approx \bar{f} := \frac{1}{|\Omega|}\int_\Omega f\, d\boldsymbol{x}$, it was simplified to

$$\min_u \frac{1}{2}\int_\Omega |\nabla u - \nabla f|^2\, d\boldsymbol{x} + \frac{\lambda}{2}\int_\Omega (u - \bar{f})^2\, d\boldsymbol{x}.$$

- Petro-Sbert-Morel *(MAA 2014)* further improved their model by using the L^1 norm to obtain sharper edges:

$$\min_u \int_\Omega |\nabla u - \nabla f|\, d\boldsymbol{x} + \frac{\lambda}{2}\int_\Omega (u - \bar{f})^2\, d\boldsymbol{x}.$$

  Note that requiring the desired image u being close to a pixel-independent constant $\bar{f}$ highly contradicts the requirement of ∇u being close to ∇f and restrains the parameter λ to be very small.

An adaptive variational model

Hsieh-Shao-Yang *(SIIMS 2020)* proposed two adaptive functions g and h to replace $\bar{f}$ and the original input image f,

$$\min_{u} \int_{\Omega} |\nabla u - \nabla h| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_{\Omega} (u - g)^2 \, d\boldsymbol{x} + \chi_{[0,255]}(u),$$

where g and h are devised respectively as

$$g(\boldsymbol{x}) = \begin{cases} \alpha \bar{f}, & \boldsymbol{x} \in \Omega_d, \\ f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_b, \end{cases} \quad h(\boldsymbol{x}) = \begin{cases} \beta f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_d, \\ f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_b, \end{cases}$$

$$\Omega_d := \{\boldsymbol{x} \in \overline{\Omega} : f(\boldsymbol{x}) \leq \bar{f}\}, \quad \Omega_b := \{\boldsymbol{x} \in \overline{\Omega} : f(\boldsymbol{x}) > \bar{f}\},$$

with a brightness parameter $\alpha > 0$ and a contrast-level parameter $\beta > 1$, and the characteristic function is defined as

$$\chi_{[0,255]}(u) = \begin{cases} 0, & \text{range}(u) \subseteq [0, 255], \\ \infty, & \text{otherwise.} \end{cases}$$

Generally speaking, Ω_d contains relatively dim elements, while Ω_b contains relatively bright elements.

Differentiability of h

To ensure the differentiability of h, in practice we smooth the coefficients and redefine the adaptive function h as

$$h(\boldsymbol{x}) = G * \left(\beta 1_{\Omega_d}(\boldsymbol{x}) + 1_{\Omega_b}(\boldsymbol{x})\right) f(\boldsymbol{x}), \quad \boldsymbol{x} \in \overline{\Omega},$$

where the indicator function $1_A(\boldsymbol{x}) = 1$, if $x \in A$, otherwise $1_A(\boldsymbol{x}) = 0$, and $G*$ represents suitable Gaussian convolution such that ∇h is well-defined.

Color RGB images

- The domain division for color RGB images denoted by (f_R, f_G, f_B) is conducted as follows. First, we define the maximum image as

$$f_{\max}(\boldsymbol{x}) := \max\{f_R(\boldsymbol{x}), f_G(\boldsymbol{x}), f_B(\boldsymbol{x})\}, \quad \forall\, \boldsymbol{x} \in \overline{\Omega}.$$

- Let $\bar{f}_{\max} := \frac{1}{|\Omega|} \int_\Omega f_{\max}\, d\boldsymbol{x}$. Then we divide the image domain Ω into two parts

$$\begin{aligned} \Omega_d &:= \{\boldsymbol{x} \in \overline{\Omega} : f_{\max}(\boldsymbol{x}) \le \bar{f}_{\max}\}, \\ \Omega_b &:= \{\boldsymbol{x} \in \overline{\Omega} : f_{\max}(\boldsymbol{x}) > \bar{f}_{\max}\}. \end{aligned}$$

- As an example, consider an element $\boldsymbol{x}^\star \in \overline{\Omega}$ with color intensities $(f_R(\boldsymbol{x}^\star), f_G(\boldsymbol{x}^\star), f_B(\boldsymbol{x}^\star)) = (25, 25, 200)$, then $f_{\max}(\boldsymbol{x}^\star) = 200$, a large value which should be classified into Ω_b.

Domain division for color images

(top row): low-light images *(bottom row): domain-division results*

Adaptive variational model for color images

- With the help of the maximum image $f_{\max}$, we can now process color images channelwise. For every $f \in \{f_R, f_G, f_B\}$, we solve

$$\min_u \int_\Omega |\nabla u - \nabla h_c| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_\Omega (u - g_c)^2 \, d\boldsymbol{x} + \chi_{[0,255]}(u),$$

where the adaptive functions g_c and h_c are defined as

$$g_c(\boldsymbol{x}) := \begin{cases} \alpha \bar{f}, & \boldsymbol{x} \in \Omega_d, \\ f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_b, \end{cases}$$

and

$$h_c(\boldsymbol{x}) := \begin{cases} \beta f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_d, \\ f(\boldsymbol{x}), & \boldsymbol{x} \in \Omega_b. \end{cases}$$

- There is no evidence shown that chooses different λ, α and β for each channel separately can have specific benefit. Therefore, for simplicity, we fix λ, α, and β across channel.

The bounded variation space $BV(\Omega)$

Let Ω be an open subset of $\mathbb{R}^2$. The space of functions of bounded variation $BV(\Omega)$ is defined as the space of real-valued function $u \in L^1(\Omega)$ such that the total variation is finite, i.e.,

$$BV(\Omega) = \{u \in L^1(\Omega) : \|u\|_{TV(\Omega)} < \infty\},$$

where

$$\|u\|_{TV(\Omega)} := \int_\Omega |Du|$$
$$:= \sup \left\{ \int_\Omega u(\nabla \cdot \varphi)\, d\mathbf{x} : \varphi \in C_c^1(\Omega, \mathbb{R}^2), \|\varphi\|_{(L^\infty(\Omega))^2} \leq 1 \right\},$$

$C_c^1(\Omega, \mathbb{R}^2)$ is the space of continuously differentiable vector functions with compact support in Ω, $L^1(\Omega)$ and $L^\infty(\Omega)$ are the usual $L^p(\Omega)$ space for $p = 1$ and $p = \infty$, respectively.

Then $BV(\Omega)$ is a Banach space with the norm,

$$\|u\|_{BV(\Omega)} := \|u\|_{L^1(\Omega)} + \|u\|_{TV(\Omega)}.$$

Existence and uniqueness of minimizer

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain with Lipschitz boundary and let $h \in BV(\Omega)$ be the input image. Then the variational problem

$$\min_u \int_\Omega |\nabla u - \nabla h| \, d\mathbf{x} + \frac{\lambda}{2} \int_\Omega (u - g)^2 \, d\mathbf{x} + \chi_{[0,255]}(u)$$

admits a unique minimizer in $BV(\Omega) \cap L^2(\Omega)$.

Remarks:

- $\int_\Omega |\nabla u| \, d\mathbf{x}$ should be realized as the total variation $\int_\Omega |Du|$.
- Let $w = u - h$, then the energy can be rewritten as the TV denoising one proposed by Goldstein-Osher (SIIMS 2009).
- direct method (Lebesgue dominated convergence) $\longrightarrow$ existence.
- strict convexity $\longrightarrow$ uniqueness.

The alternating minimization algorithm

- The discrete gradient of u is defined as $\nabla u_{i,j} = (\nabla_x^+ u_{i,j}, \nabla_y^+ u_{i,j})$,

$$\nabla_x^+ u_{i,j} := \begin{cases} (u_{i,j+1} - u_{i,j})/h, & 1 \le j \le N-1, \\ 0, & j = N, \end{cases}$$

$$\nabla_y^+ u_{i,j} := \begin{cases} (u_{i+1,j} - u_{i,j})/h, & 1 \le i \le N-1, \\ 0, & i = N, \end{cases}$$

- The continuous model can be discretized as

$$\min_u \sum_{i,j} |\nabla u_{i,j} - \nabla h_{i,j}| + \frac{\lambda}{2}(u_{i,j} - g_{i,j})^2 + \chi_{[0,255]}(u).$$

- Applying the operator splitting, it is then equivalent to

$$\min_{u,d,v} \sum_{i,j} \left(|d_{i,j}| + \frac{\lambda}{2}(u_{i,j} - g_{i,j})^2 \right) + \chi_{[0,255]}(v),$$

$$\text{subject to } d = \nabla u - \nabla h \text{ and } v = u.$$

The Bregman iterations

- The splitted problem can be solved by using the Bregman iteration. Introducing the penalty parameter $\gamma > 0$ and $\delta > 0$, we arrive at the following unconstrained minimization problem:

$$\min_{u,d,v} \sum_{i,j} \Big(|d_{i,j}| + \frac{\lambda}{2}(u_{i,j} - g_{i,j})^2 + \frac{\gamma}{2}|d_{i,j} - \nabla u_{i,j} + \nabla h_{i,j} - b_{i,j}|^2 + \frac{\delta}{2}(v_{i,j} - u_{i,j} - c_{i,j})^2 \Big) + \chi_{[0,255]}(v),$$

  where b and c are the variables related to the Bregman iterations.

- Then the problem is solved by alternating the search directions of u, d, and v.

The split Bregman iterations: 3 subproblems + 2 identities

- **u-subproblem:**

$$u^{n+1} = \arg\min_u \sum_{i,j} \Big(\frac{\lambda}{2}(u_{i,j} - g_{i,j})^2 + \frac{\gamma}{2}\big|d^n_{i,j} - (\nabla u)_{i,j} + (\nabla h)_{i,j} - b^n_{i,j}\big|^2 + \frac{\delta}{2}(v^n_{i,j} - u_{i,j} - c^n_{i,j})^2 \Big);$$

- **d-subproblem:**

$$d^{n+1} = \arg\min_d \sum_{i,j} \Big(|d_{i,j}| + \frac{\gamma}{2}\big|d_{i,j} - (\nabla u^{n+1})_{i,j} + (\nabla h)_{i,j} - b^n_{i,j}\big|^2 \Big);$$

- **v-subproblem:**

$$v^{n+1} = \arg\min_v \sum_{i,j} \Big(\frac{\delta}{2}(v_{i,j} - u^{n+1}_{i,j} - c^n_{i,j})^2 \Big) + \chi_S(v);$$

- **Bregman variables b and c:**

$$b^{n+1} = b^n + \nabla u^n - \nabla h - d^{n+1}, \quad c^{n+1} = c^n + u^{n+1} - v^{n+1}.$$

u-subproblem

u-subproblem:

$$u^{n+1} = \arg\min_u \sum_{i,j} \Big(\frac{\lambda}{2}(u_{i,j} - g_{i,j})^2 + \frac{\gamma}{2} \big| d_{i,j}^n - (\nabla u)_{i,j} + (\nabla h)_{i,j} - b_{i,j}^n \big|^2 + \frac{\delta}{2}(v_{i,j}^n - u_{i,j} - c_{i,j}^n)^2 \Big).$$

It can be viewed as the discretization of the minimization problem:

$$\min_u \frac{\lambda}{2} \int_\Omega (u - g)^2 \, d\boldsymbol{x} + \frac{\gamma}{2} \int_\Omega |d - \nabla u + \nabla h - b|^2 \, d\boldsymbol{x} + \frac{\delta}{2} \int_\Omega (v - u - c)^2 \, d\boldsymbol{x}.$$

The EL equation of the above minimization problem is given by

$$(\lambda + \delta) u - \gamma \Delta u = \lambda g - \gamma \big(\mathrm{div}(d + \nabla h - b) \big) + \delta (v - c).$$

Note: $\frac{\partial L}{\partial u} - \nabla \cdot (\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y})^\top = 0$ in Ω, $\quad \frac{\partial L}{\partial u_x} n_1 + \frac{\partial L}{\partial u_y} n_2 = 0$ on $\partial\Omega$.

u-subproblem (cont'd)

We obtain the discrete equations:

$$(\lambda+\delta)u_{i,j}^{n+1}-\gamma(\Delta u^{n+1})_{i,j}=\lambda g_{i,j}-\gamma\big(\mathrm{div}(d^n+\nabla h-b^n)\big)_{i,j}+\delta(v_{i,j}^n-c_{i,j}^n).$$

The discrete operators div and Δ are defined as follows:

- Given $p=(p^1,p^2)$ with $p^1,p^2\in\mathbb{R}^{N\times N}$, we define

  $$(\mathrm{div}\,p)_{i,j}:=(\nabla_x^- p^1)_{i,j}+(\nabla_y^- p^2)_{i,j}:=(p_{i,j}^1-p_{i,j-1}^1)+(p_{i,j}^2-p_{i-1,j}^2).$$

- The discrete Laplacian is then defined as the composite of ∇ and div as $\Delta u:=\mathrm{div}(\nabla u)$.
- Since the discretized problem produces a symmetric and diagonally dominant linear system, some iterative solvers such as Jacobi method or Gauss-Seidel method can be employed for efficiently solving u.

d-subproblem

d-subproblem:

$$d^{n+1} = \arg\min_{d} \sum_{i,j} \left(|d_{i,j}| + \frac{\gamma}{2} \left| d_{i,j} - (\nabla u^{n+1})_{i,j} + (\nabla h)_{i,j} - b^n_{i,j} \right|^2 \right).$$

The objective function is strictly convex and it has the following closed-form solution:

$$\begin{aligned} d^{n+1}_{i,j} &= \frac{(\nabla u^{n+1})_{i,j} - (\nabla h)_{i,j} + b^n_{i,j}}{\left|(\nabla u^{n+1})_{i,j} - (\nabla h)_{i,j} + b^n_{i,j}\right|} \\ &\quad \times \max\left\{ \left|(\nabla u^{n+1})_{i,j} - (\nabla h)_{i,j} + b^n_{i,j}\right| - \frac{1}{\gamma},\ 0 \right\}. \end{aligned}$$

v-subproblem, Bregman variables, and initialization

v-subproblem:

$$v^{n+1} = \arg\min_{v} \sum_{i,j} \left(\frac{\delta}{2} (v_{i,j} - u_{i,j}^{n+1} - c_{i,j}^{n})^2 \right) + \mathcal{X}_S(v).$$

For the v-subproblem, it can be solved by pixel-wise orthogonal projection of $u + c$ onto the predefined interval $S := [s_1, s_2]$

$$v_{i,j} = \min \Big\{ \max \{ u_{i,j} + c_{i,j}, s_1 \}, s_2 \Big\}.$$

Note that we take $S = [s_1, s_2] := [0, 255]$.

Bregman variables b and c:

$$b^{n+1} = b^n + \nabla u^n - \nabla h - d^{n+1}, \quad c^{n+1} = c^n + u^{n+1} - v^{n+1}.$$

Initialization: $u = h, v = h, d = 0, b = 0, c = 0.$

Numerical experiments and comparisons

(T): f, u_{MPS}, u_{HE} (B): u_{VCE}, u_{CLAHE}, $u_{MLHE-HE}$

Numerical experiments and comparisons

(T): $u_{ACE}(\alpha = 2, 4, 6)$ (B): $u_{Adaptive}(\alpha = 0.8, 1.0, 1.2), \beta = 3\alpha$

Surprisingly, under the same parameter setting, the iteration number of our model is far less than that of the MPS model.

Numerical experiments and comparisons

(T): f, u_{MPS}, u_{HE} (B): u_{VCE}, u_{CLAHE}, $u_{MLHE-HE}$

© Suh-Yuh Yang (楊肅煜), Math. Dept., NCU, Taiwan

Numerical experiments and comparisons

(T): $u_{ACE}(\alpha = 2, 4, 6)$ (B): $u_{Adaptive}(\alpha = 0.8, 1.0, 1.2), \beta = 3\alpha)$

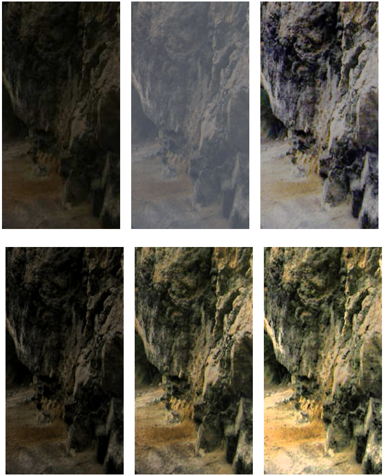
Numerical experiments and comparisons

(T): f, u_{MPS}, u_{HE} (B): u_{VCE}, u_{CLAHE}, $u_{MLHE-HE}$

Numerical experiments and comparisons

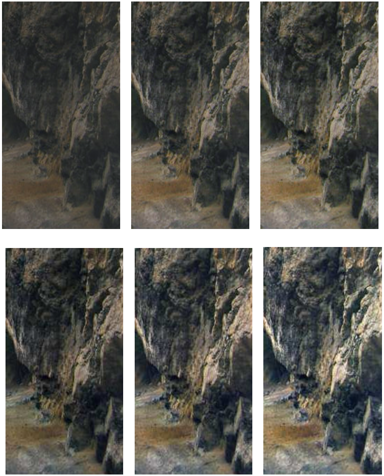
(T): $u_{ACE}(\alpha = 2, 4, 6)$ (B): $u_{Adaptive}(\alpha = 0.8, 1.0, 1.2), \beta = 3\alpha)$

Numerical experiments and comparisons



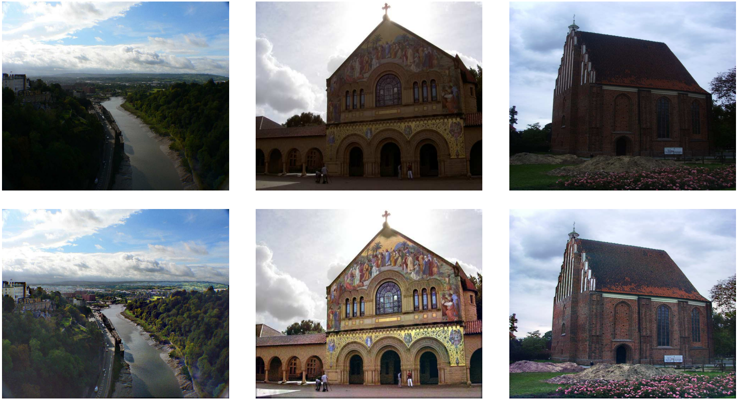
(T): f, u_{MPS}, u_{HE} (B): $u_{VCE}, u_{CLAHE}, u_{MLHE-HE}$

Numerical experiments and comparisons



(T): $u_{ACE}(\alpha = 2, 4, 6)$ (B): $u_{Adaptive}(\alpha = 0.8, 1.0, 1.2, \beta = 3\alpha)$

Numerical results of the proposed method



(T): low-light images (B): enhanced images

Numerical results of the proposed method

(T): low-light images   *(B): enhanced images*

Summary

1. We have proposed a simple and efficient adaptive variational model for image contrast enhancement.
2. This model is designed for enhancing low-light images by dividing the image domain into bright and dim parts.
3. The existence and uniqueness of minimizer for the minimization problem is established, and a convergent algorithm is provided.
4. The most distinguished feature of our model is that colors are preserved as close as possible to the original ones.