Group 6: Variational Models and Numerical Methods for Image Processing - Final Presentation

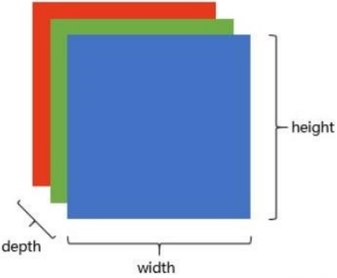
Members: Jia-Wei Liao (NTNU)
Chun-Hsien Chen (NCCU)
Chen-Yang Dai (NCTU)
Advisor: Suh-Yuh Yang (NCU), TA: Chen-Yan Wei (NCU)

2020 NCTS Undergraduate Summer Research Program

August 28, 2020

Outline

1. Introduction to image processing
   - Image processing
   - Preliminary knowledge of mathematics
   - Algorithm: split Bregman iteration
2. Image denoising
   - ROF model
   - Discretization
3. Image contrast enhancement
   - Model
   - Discretization
4. Image stitching
   - Find features: SIFT algorithm
   - Match features: affine map and homography
   - Blending and contrast enhancement



Remark

Let Ω be an open set in $\mathbb{R}^2$ and $u : \bar{\Omega} \to \mathbb{R}$

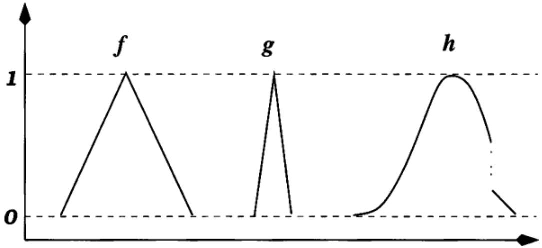
- $u(x) \in [0, 255]\,, \forall x \in \bar{\Omega}$
- $0 \to$ black, $255 \to$ white
- color image: RGB channels

Total variation

Definition (Total variation of one variable function)

Let $\Omega = (a, b) \subseteq \mathbb{R}$ and $\mathcal{P}_n = \{a = x_0, x_1, \cdots, x_{n-1}, x_n = b\}$, be an arbitrary partition of $\bar{\Omega}$. The total variation of a real-valued function $u : \Omega \to \mathbb{R}$ is defined as the quantity,

$$\|u\|_{TV(\Omega)} = \sup_{\mathcal{P}_n} \sum_{i=1}^{n} |u(x_i) - u(x_{i-1})| \,.$$



Total variation

Definition (Total variation of one variable function)

Let $\Omega = (a, b) \subseteq \mathbb{R}$ and $\mathcal{P}_n = \{a = x_0, x_1, \cdots, x_{n-1}, x_n = b\}$, be an arbitrary partition of $\bar{\Omega}$. The total variation of a real-valued function $u : \Omega \to \mathbb{R}$ is defined as the quantity,

$$\|u\|_{TV(\Omega)} = \sup_{\mathcal{P}_n} \sum_{i=1}^{n} |u(x_i) - u(x_{i-1})| .$$

Theorem

If u is a smooth function, then

$$\|u\|_{TV(\Omega)} = \int_{\Omega} |u'(x)| dx.$$

Total variation

Definition (Total variation of two variable function)

Let Ω be an open set of $\mathbb{R}^2$ and $u \in L^1(\mathbb{R})$. The total variation of u in Ω is defined as

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

where $C_c^1\left(\Omega, \mathbb{R}^n\right)$ is the set of continuously differentiable vector functions of compact support contained in Ω, and $\|\cdot\|_{L^\infty(\Omega)}$ is the essential supremum norm.

Theorem

If u is a smooth function, then

$$\|u\|_{TV(\Omega)} = \int_\Omega |\nabla u| d\boldsymbol{x}.$$

Definition (Bounded variation)

If $\|u\|_{TV(\Omega)} < \infty$, then we say that u is a function of bounded variation. Moreover, the space of functions of bounded variation $BV(\Omega)$ is defined as $u \in L^1(\Omega)$ such that the total variation is finite, i.e.,

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

Remark

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

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

Euler-Lagrange

Let $[a, b] \subseteq \mathbb{R}$. We consider the functional,

$$E[y] = \int_a^b L\left(x, y, y'\right) dx,$$

where we assume that $y \in C^2([a, b])$ and $L \in C^2$ with respect to its arguments x, y and y'.

Euler-Lagrange equation (1-dimension)

A necessary condition for a local minimum y of E is

$$\frac{\partial L}{\partial y} - \frac{d}{dx}\left(\frac{\partial L}{\partial y'}\right) = 0.$$

Euler-Lagrange

Let $\Omega \subseteq \mathbb{R}^2$ be an open set. We consider the functional,

$$E[u] = \int_\Omega L\left(x, y, u, u_x, u_y\right) d(x, y),$$

where we assume that $u \in C^2(\bar{\Omega})$ and $L \in C^2$ with respect to its arguments x, y, u, u_x and u_y.

Euler-Lagrange equation (2-dimension)

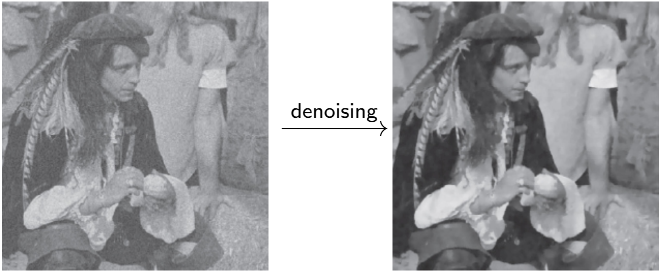
A necessary condition for a local minimum u of E is

$$\frac{\partial L}{\partial u} - \nabla \cdot \left(\frac{\partial L}{\partial u_x}, \frac{\partial L}{\partial u_y} \right) = 0.$$

Image denoising

Mathematics method of image processing

1. Fourier transform
2. Heat-type equation



Mathematics method of image processing

1. Fourier transform
2. Heat-type equation
3. Machine learning

Image denoising

denoising $\longrightarrow$

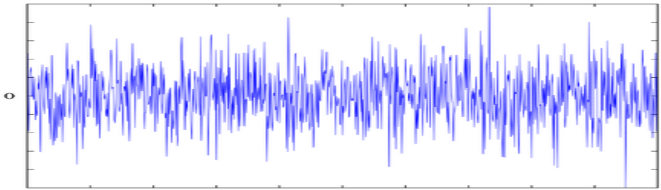
Mathematics method of image processing

1. Fourier transform
2. Heat-type equation
3. Machine learning
4. **Variational method (energy functional)**

Signal denoising

Denoising (1-dimension)

$$\text{minimizes} \left(\int_\Omega |u'(x)|\, dx + \text{ (some data fidelity term) } \right)$$



ROF model

ROF model (Physica D, 1992)

Let $f : \bar{\Omega} \subseteq \mathbb{R}^2 \to \mathbb{R}$ be a given noisy image. Rudin, Osher, and Fatemi proposed the model for image denoising:

$$\min_{u \in BV(\Omega)} \Big(\underbrace{\|u\|_{TV(\Omega)}}_{\text{regularizer}} + \frac{\lambda}{2} \underbrace{\int_{\Omega} (u-f)^2 d\boldsymbol{x}}_{\text{data fidelity}} \Big),$$

where $\lambda > 0$ is a tuning parameter which controls the regularization strengt.

ROF model

ROF model (Physica D, 1992)

Let $f : \bar{\Omega} \subseteq \mathbb{R}^2 \to \mathbb{R}$ be a given noisy image. Rudin, Osher, and Fatemi proposed the model for image denoising:

$$\min_{u \in BV(\Omega)} \Bigg(\underbrace{\|u\|_{TV(\Omega)}}_{\text{regularizer}} + \underbrace{\frac{\lambda}{2} \int_\Omega (u-f)^2 d\boldsymbol{x}}_{\text{data fidelity}} \Bigg),$$

where $\lambda > 0$ is a tuning parameter which controls the regularization strengt.

Remark

1. A smaller value of λ will lead to a more regular solution.
2. The space of functions with bounded variation help remove spurious oscillations (noise) and preserve sharp signals (edges).
3. The *TV* term allows the solution to have discontinuities.

- **ROF Model:**

$$\min_{u \in BV(\Omega)} \left(\|u\|_{TV(\Omega)} + \frac{\lambda}{2} \int_{\Omega} (u-f)^2 d\boldsymbol{x} \right)$$

Discretization of the ROF model

- **ROF Model:**

$$\min_{u \in BV(\Omega)} \left(\|u\|_{TV(\Omega)} + \frac{\lambda}{2} \int_{\Omega} (u - f)^2 d\boldsymbol{x} \right)$$

- **Discretization:**

$$\min_{u} \left(\sum_{i,j} |(\nabla u)_{i,j}| + \frac{\lambda}{2} \sum_{i,j} (u_{i,j} - f_{i,j})^2 \right)$$

Discretization of the ROF model

- **ROF Model:**

$$\min_{u\in BV(\Omega)}\left(\|u\|_{TV(\Omega)}+\frac{\lambda}{2}\int_{\Omega}(u-f)^2 d\boldsymbol{x}\right)$$

- **Discretization:**

$$\min_{u}\left(\sum_{i,j}|(\nabla u)_{i,j}|+\frac{\lambda}{2}\sum_{i,j}(u_{i,j}-f_{i,j})^2\right)$$

- **Constraint:**

$$\min_{d,u}\left(\sum_{i,j}|d_{i,j}|+\frac{\lambda}{2}\sum_{i,j}(u_{i,j}-f_{i,j})^2\right)$$

subject to $d_{i,j}=\nabla u_{i,j}$

Discretization of the ROF model

- **Constraint:**

$$\min_{d,u}\left(\sum_{i,j}|d_{i,j}| + \frac{\lambda}{2}\sum_{i,j}(u_{i,j} - f_{i,j})^2\right)$$

subject to $d_{i,j} = \nabla u_{i,j}$

- **Bregman iteration:**

$$\min_{d,u}\left(\sum_{i,j}|d_{i,j}| + \frac{\lambda}{2}\sum_{i,j}(u_{i,j} - f_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j}|d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2\right)$$

Split Bregman algorithm

Bregman iteration

$$\min_{d,u}\left(\sum_{i,j}|d_{i,j}| + \frac{\lambda}{2}\sum_{i,j}(u_{i,j}-f_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j}|d_{i,j}-\nabla u_{i,j}-b_{i,j}|^2\right)$$

- **u-subproblem:**

With d fixed, we solve

$$u^{(k+1)} = \arg\min_u\left(\frac{\lambda}{2}\sum_{i,j}(u_{i,j}-f_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j}\left|d_{i,j}^{(k)}-\nabla u_{i,j}-b_{i,j}^{(k)}\right|^2\right).$$

Then consider the minimization problem

$$\min_u \int_\Omega \left(\frac{\lambda}{2}(u-f)^2 + \frac{\gamma}{2}|d-\nabla u-b|^2 d\boldsymbol{x}\right).$$

By Euler-Lagrange equation, we have

$$\lambda(u-f) - \gamma\left[\nabla\cdot(\nabla u - d + b)\right] = 0,$$

- **u-subproblem (continue):**

  or equivalently,

$$\lambda u - \gamma \Delta u = \lambda f - \gamma \nabla \cdot (d - b).$$

Notice that:

$$\begin{aligned}\Delta u_{i,j} &= (u_{i,j-1} + u_{i,j+1} - 2u_{i,j}) + (u_{i-1,j} + u_{i+1,j} - 2u_{i,j}) \\ &= u_{i-1,j} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j} - 4u_{i,j}\end{aligned}$$

So, we have

$$(\lambda + 4\gamma)u_{i,j} = c_{i,j} + \gamma \left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\right),$$

where $c_{i,j} = (\lambda f - \gamma \nabla \cdot (d - b))_{i,j}$.

- **u-subproblem (continue):**

$$(\lambda + 4\gamma)u_{i,j} = c_{i,j} + \gamma \left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}\right),$$

which is a symmetric and strictly diagonally dominant linear system, by the Jacobi iterative method:

$$u_{i,j}^{(k+1)} = \left[c_{i,j}^{(k)} + \gamma \left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}\right)\right] / (\lambda + 4\gamma).$$

Bregman iteration

$$\min_{d,u}\left(\sum_{i,j}|d_{i,j}| + \frac{\lambda}{2}\sum_{i,j}(u_{i,j}-f_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j}|d_{i,j}-\nabla u_{i,j}-b_{i,j}|^2\right)$$

- **d-subproblem:**

  With u fixed, we solve

$$d^{(k+1)} = \arg\min_{d}\left(\sum_{i,j}|d_{i,j}| + \frac{\gamma}{2}\sum_{i,j}\left|d_{i,j}-\nabla u_{i,j}^{(k+1)}-b_{i,j}^{(k)}\right|^2\right).$$

Notice that: Consider the simple 1-D case,

$$\arg\min_{x}\left(\tau|x| + \frac{\rho}{2}(x-y)^2\right) = \begin{cases} y-\tau/\rho, & y>\tau/\rho \\ 0, & |y|\le\tau/\rho \\ y+\tau/\rho, & y<-\tau/\rho \end{cases}.$$

- **d-subproblem (continue):**

  Notice that: Consider the simple 1-D case,

$$\arg\min_x \left(\tau|x| + \frac{\rho}{2}(x-y)^2 \right) = \frac{y}{|y|} \max\left\{ |y| - \tau/\rho, 0 \right\}.$$

  Then we have

$$d_{i,j}^{(k+1)} = \frac{\nabla u_{i,j}^{(k+1)} + b_{i,j}^{(k)}}{\left| \nabla u_{i,j}^{(k+1)} + b_{i,j}^{(k)} \right|} \max\left\{ \left| \nabla u_{i,j}^{(k+1)} + b_{i,j}^{(k)} \right| - \frac{1}{\gamma}, 0 \right\}.$$

- **Updating b:** $b_{i,j}^{(k+1)} = b^{(k)} + \nabla u^{(k+1)} - d^{(k+1)}$.

Split Bregman algorithm

Split Bregman algorithm

Initialize $u = f, b = 0, d = 0$

while $\frac{\|u - u_{prev}\|}{\|u_{prev}\|} > tolerance$ do

for $n = 1$ to $maxstep$ do

Solve the **u-subproblem**

Solve the **d-subproblem**

$b \leftarrow b + \nabla u - d$

end for

end while

Numerical experiments

ROF Model

$$\min_{u \in BV(\Omega)} \left(\|u\|_{TV(\Omega)} + \frac{\lambda}{2} \int_\Omega (u-f)^2 d\boldsymbol{x} \right)$$

Some Indices

Let $\widetilde{u}$ be the clean image, $\bar{u}$ be the mean intensity of the clean image, and u be the produced image.

1. **Mean square error:** $\text{MSE} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (\widetilde{u}_{i,j} - u_{i,j})^2$
2. **Peak signal to noise ratio:** $\text{PSNR} = 10 \log \left(\frac{255^2}{\text{MSE}(\widetilde{u}, u)} \right)$
3. **Signal to noise ratio:** $\text{SNR} = 10 \log \left(\frac{\text{MSE}(\widetilde{u}, \bar{u})}{\text{MSE}(\widetilde{u}, u)} \right)$

Numerical experiments: grayscale image

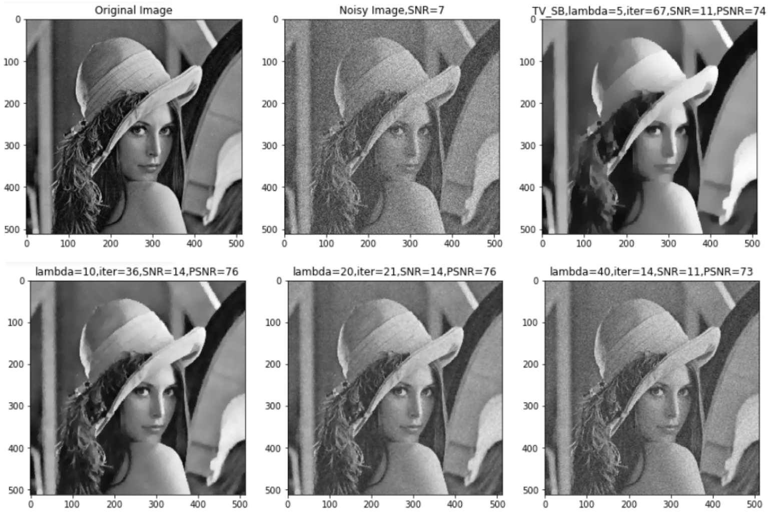


Figure: Lenna

Numerical experiments: color image

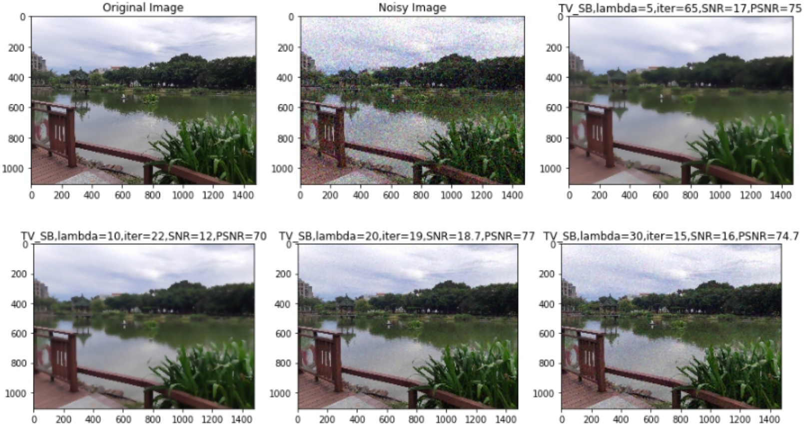


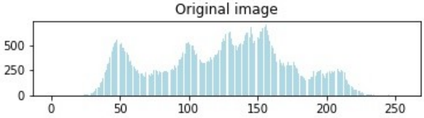
Figure: Drunken moon lake at NTU

Image contrast enhancement

contrast enhancement $\longrightarrow$

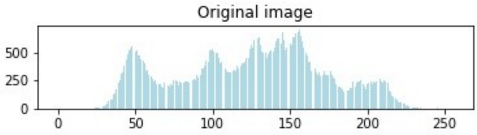
Histogram equalization (HE)

Origin



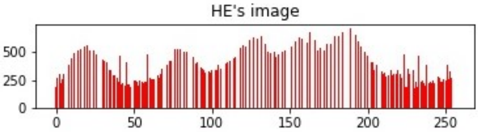
Histogram

Origin image



Origin histogram

HE's image

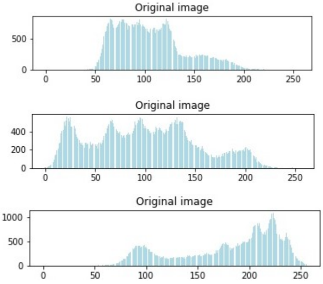


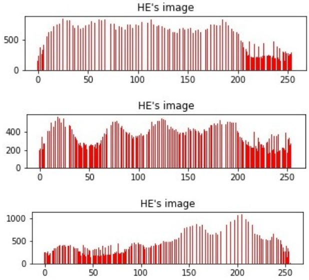
HE's histogram

Histogram equalization (HE)

Origin image

HE's image





Morel-Petro-Sbert model (IPOL 2014)

Let $f : \overline{\Omega} \to \mathbb{R}$ be a given grayscale image. The Morel-Petro-Sbert oposed the model for image contrast enhancementis:

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

where $\overline{u} = \frac{1}{|\Omega|}\int_\Omega u \, dx$ is the mean value of u over Ω and $\lambda > 0$ balances between detail preservation and variance reduction.

Remark

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.

Contrast enhancement

The original model is simple but difficult to solve due to the $\overline{u}$ term. So, we assuming that $\overline{u} \approx \overline{f}$.

Contrast enhancement

The original model is simple but difficult to solve due to the $\overline{u}$ term. So, we assuming that $\overline{u} \approx \overline{f}$.

Petro-Sbert-Morel model (MAA 2014)

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

$$\min_{u} \left(\int_{\Omega} |\nabla u - \nabla f| \, d\boldsymbol{x} + \frac{\lambda}{2} \int_{\Omega} (u - \overline{f})^2 \, d\boldsymbol{x} \right).$$

Remark

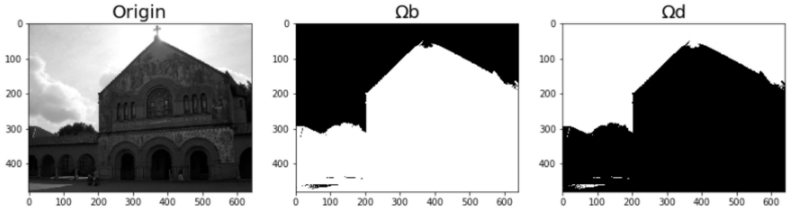
Requiring the desired image u being close to a pixel-independent constant $\overline{f}$ highly contradicts the requirement of ∇u being close to ∇f and restrains the parameter λ to be very small.

First, We define

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

as the dark part and the bright part of the image Ω.
Second, define the adaptive functions

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



Hsieh-Shao-Yang model (SIIMS 2020)

Hsieh-Shao-Yang proposed two adaptive functions g and h to replace $\overline{f}$ and the original input image f

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

where g and h are devised respectively as

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

with $\alpha > 0$ and $\beta > 1$ and the characteristic function is defined as

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

Discretization of model

- **Model:**

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

Discretization of model

- **Model:**

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

- **Discretization:**

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

Discretization of model

- **Model:**

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

- **Discretization:**

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

- **Constraint:**

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

subject to $d = \nabla u - \nabla h$ and $v = u$

Discretization of model

- **Discretization:**

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

- **Bregman iteration:**

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

Split Bregman algorithm

Initialize $u = h, v = h, d = 0, b = 0, c = 0$

while $\frac{\|u - u_{prev}\|}{\|u_{prev}\|} > tolerance$ do

  for $n = 1$ to $maxstep$ do

    Solve the **u-subproblem**

    Solve the **d-subproblem**

    Solve the **v-subproblem**

    $b \leftarrow b + \nabla u - \nabla h - d$

    $c \leftarrow c + u - v$

  end for

end while

Numerical experiments: grayscale image

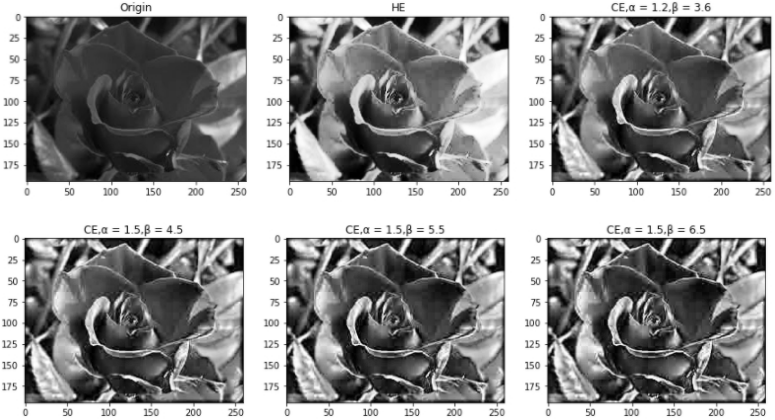


Figure: Rose ($\lambda = 0.0005$)

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

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

- For example,

65	27	100
22	31	47
112	54	78

58	21	10
145	213	48
132	2	9

15	122	200
189	32	45
12	52	79

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

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

- For example,

65	27	100
22	31	47
112	54	78

58	21	10
145	213	48
132	2	9

15	122	200
189	32	45
12	52	79

$\downarrow$

65	122	200
189	213	48
132	54	79

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

$$\Omega_d = \{\boldsymbol{x} \in \overline{\Omega} : f_{\max}(\boldsymbol{x}) \leq \overline{f}_{\max}\},$$
$$\Omega_b = \{\boldsymbol{x} \in \overline{\Omega} : f_{\max}(\boldsymbol{x}) > \overline{f}_{\max}\}.$$

Numerical experiments: color image

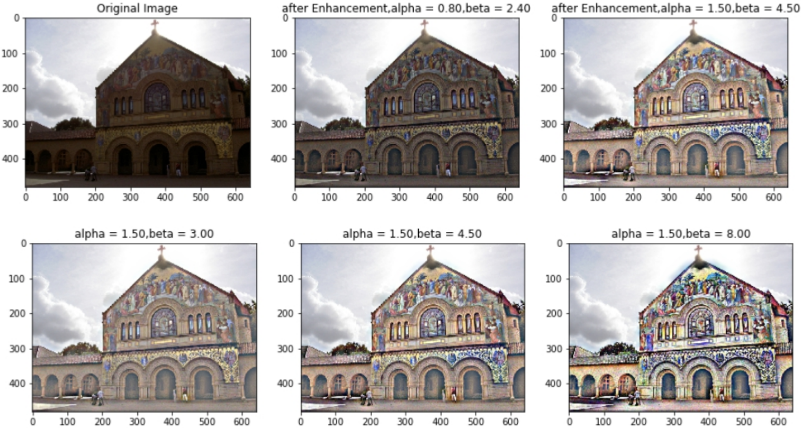


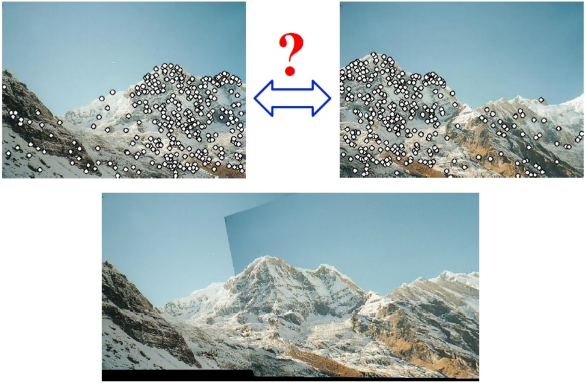
Figure: House ($\lambda = 0.0005$)

Image stitching

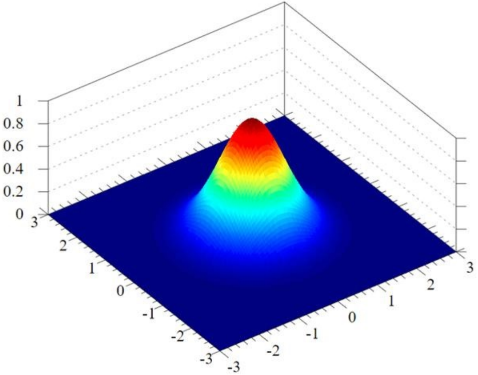
1. Image alignment
    - Scale-invariant feature transform (SIFT): find features
    - Homography: match features

Image stitching

1. Image alignment
   - Scale-invariant feature transform (SIFT): find features
   - Homography: match features
2. Image blending: linear blending



SIFT: Gaussian blur

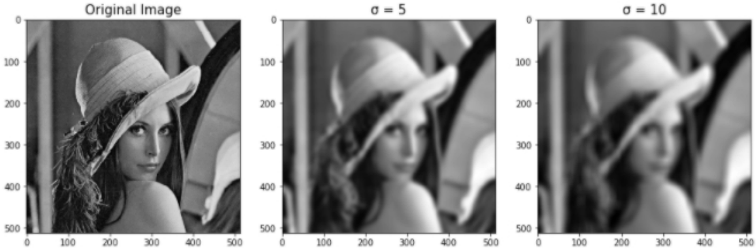


$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

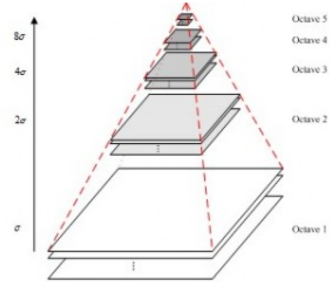
Convolution

Let original image be $I(x, y)$ and gaussian kernel $G(x, y, \sigma)$.

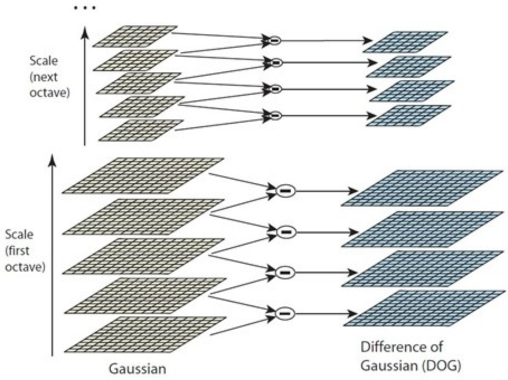
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$



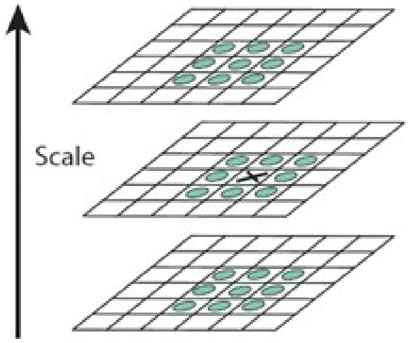
SIFT: Gaussian pyramid



SIFT: difference of Gaussian pyramid



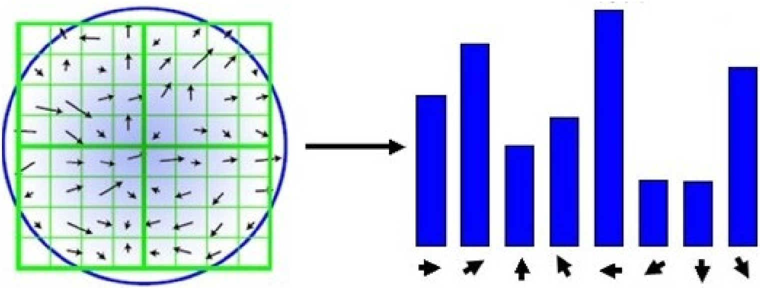
SIFT: find extrema



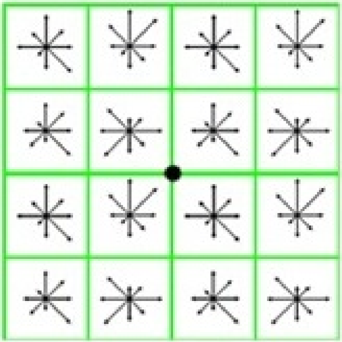
SIFT: rotation invariance

Direction

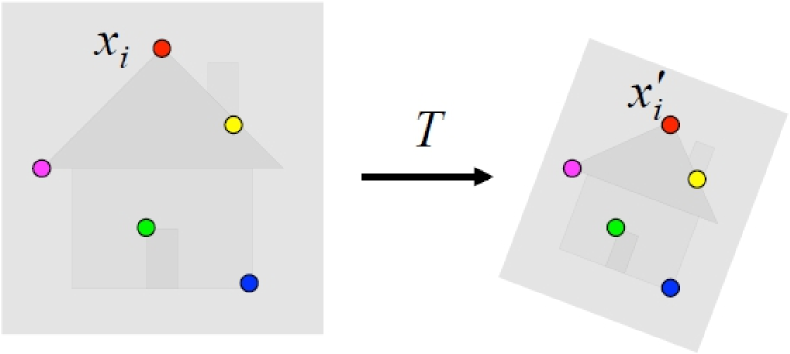
- $m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$
- $\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right)$



SIFT: descriptor



Alignment as fitting



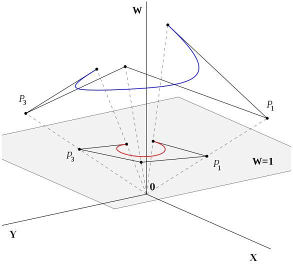
Homogeneous coordinates

- Converting to homogeneous image coordinates:

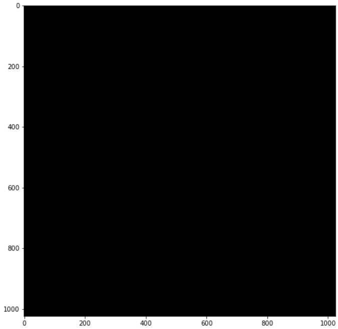
$$\begin{bmatrix} x \\ y \end{bmatrix} \longrightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Converting from homogeneous image coordinates:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix}, \text{ where } w \neq 0 \longrightarrow \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$



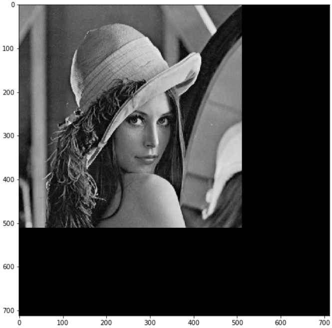
Transformation: scale



$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix},$$

where $a, b > 0$.

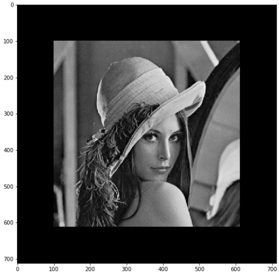
Transformation: translation



$$\begin{bmatrix} 1 & 0 & e \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix},$$

where $e, f \in \mathbb{R}$.

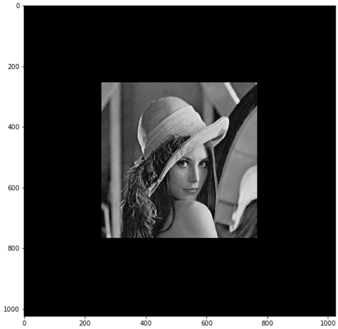
Transformation: rotation



$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix},$$

where $\theta \in (0, 2\pi)$.

Transformation: shear



$$\begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

or

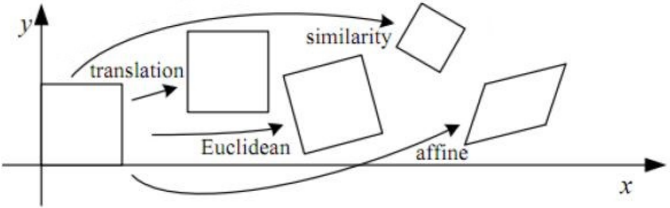
$$\begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix},$$

where $a, b > 0$.

Affine transformation

Remark

Scale + Translation + Rotation + Shear = Affine transform



Affine transformation

Remark

Scale + Translation + Rotation + Shear = Affine transform

Affine transformation

A 2D affine transformation is composed of a linear transformation by $\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \mathbb{R}^{2\times 2}$ and a translation by a vector $\begin{bmatrix} e \\ f \end{bmatrix} \in \mathbb{R}^2$, given as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

Affine transformation

Homogeneous expression of 2D affine transformation

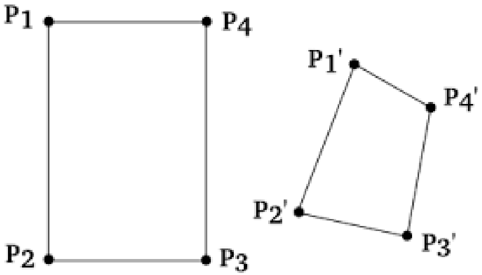
The homogeneous expression of the affine transformation is given as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

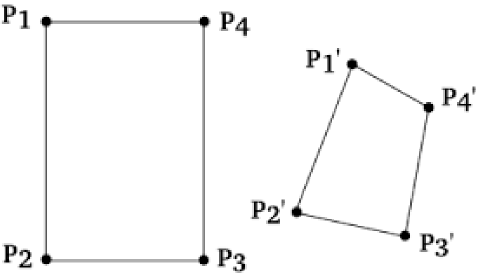
Remark

The affine transformation has 6 degree of freedom.

Homography



Homography



Homogeneous expression of homography

The homogeneous expression of the homography is given as

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Homography

Homogeneous expression of homography

The homogeneous expression of the homography is given as

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Remark

The homogeneous expression of the homography has 8 degree of freedom (9 parameters, but scale is arbitrary).

Why do we need to do image blending?

Why do we need to do image blending?

Image blending

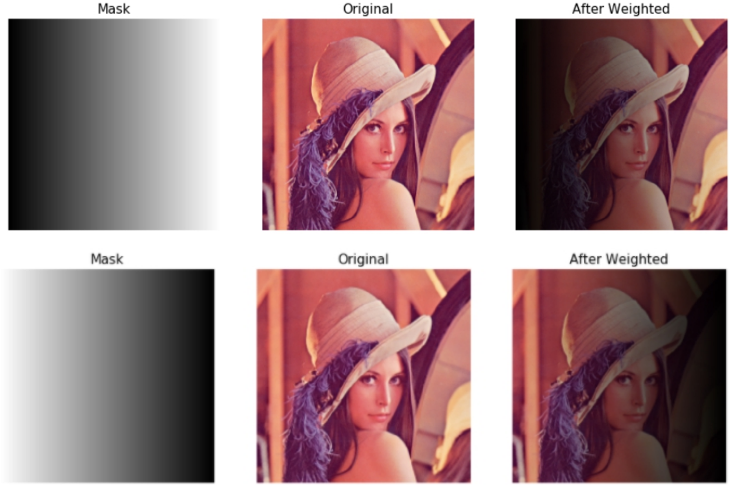


Figure: Linear blending

Experiments: draw matches

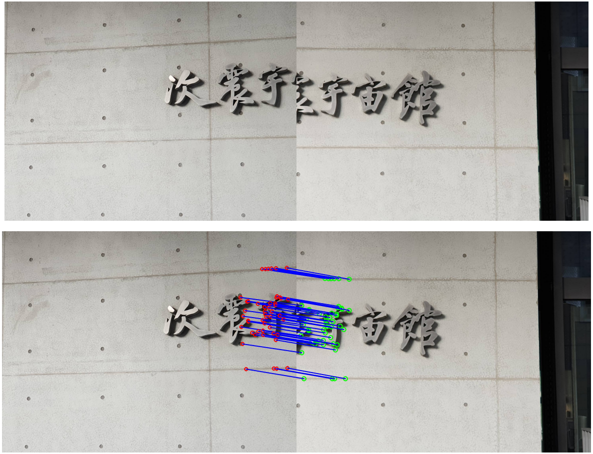


Figure: Draw matches

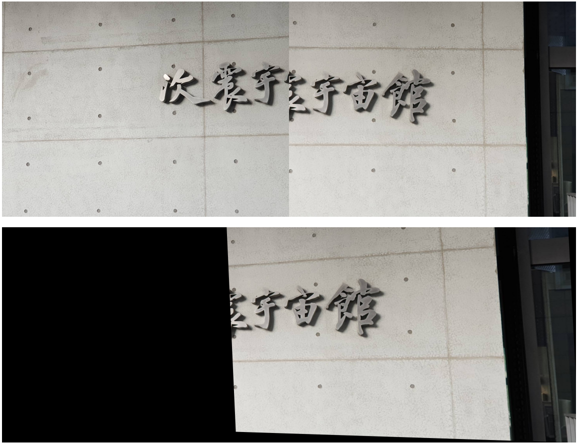


Figure: Warp perspective

Experiments: blending

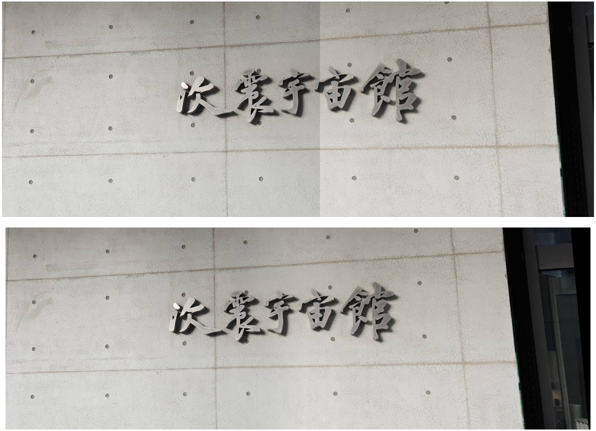


Figure: No blending versus linear blending

Experiments: panorama

Figure: Drunken moon lake at NTU

Experiments: contrast enhancement

Origial Image

CE

Figure: Drunken moon lake at NTU

Experiments: contrast enhancement

Figure: Street tree at NTU

Codes on Github

1. Image denoising
   https://github.com/SeanChenTaipei/ImageProcessing/blob/master/Adaptive-Model.ipynb
2. Image contrast enhancement
   https://github.com/SeanChenTaipei/ImageProcessing/blob/master/Contrast_Enhencement.ipynb
3. Image stitching
   https://github.com/SeanChenTaipei/ImageProcessing/blob/master/Image_Stitching.ipynb

References

- M. Brown and D. G. Lowe, Recognising panoramas, Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 1218-1225 vol.2, doi: 10.1109/ICCV.2003.1238630.
- L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60 (1992), pp. 259-268.
- 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.
- Tom Goldstein and Stanley Osher,The Split Bregman Method for L1-Regularized Problems,SIAM J. Imaging Sci., 2(2), 323–343. (21 pages)
- S.-Y. Yang. Website: http://www.math.ncu.edu.tw/~syyang/research/2020NCTS_USRP.pdf

Thanks for listening!