華東師範大學
EAST CHINA NORMAL UNIVERSITY

Guided Image Filtering

Shang Gao

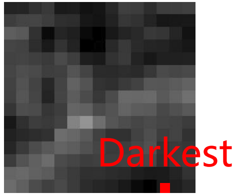
Multi-Agent Artificial Intelligence Laboratory
East China Normal University

Outline

- Introduction
- Guided Image Filtering
- Analysis
- Improvements & Conclusions

Introduction

Recall: Dark Channel Prior



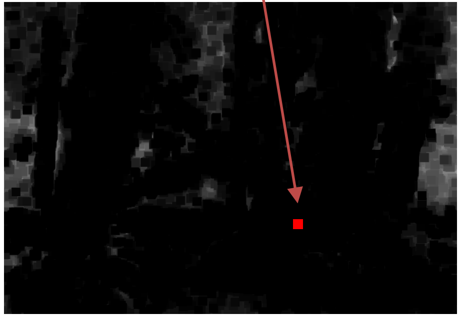
15 x 15 patch

For an image **J**, define:

$$J^{dark}(x) = \min_{y}\left(\min_{c}\left(J^{c}(y)\right)\right)$$

Input Image

min (r, g, b)



Dark Channel

Recall: Haze Imaging Model

$$I = J \cdot t + A \cdot (1 - t)$$

Atmospheric light

Hazy Image

Scene radiance
(Haze Free Image)

Transmission

- Transmission: 描述无法散射并到达照相机的光的介质传输率

Recall: Soft Matting

Contains some block effects since the transmission is not always constant in a patch.

Input I

Estimated t

Haze Free Image J

Recall: Soft Matting

Haze imaging model $$I = J \cdot t + A \cdot (1 - t)$$

Matting model [Levin et al., CVPR '06] $$I = F \cdot \alpha + B \cdot (1 - \alpha)$$

Input I

+

tri-map

α

Input I

Estimated $\tilde{t}$

Refined t

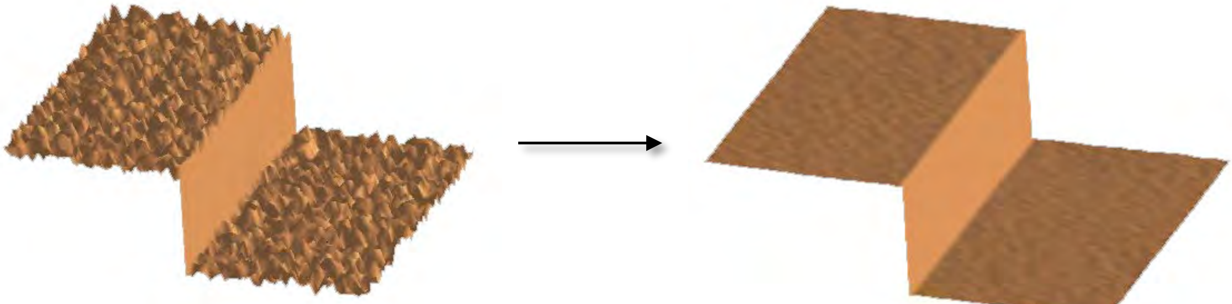
Edge-Preserving Filtering

Box filter/Gaussian filter:

- Erase some of the detail and reduce the performance of the edge in the picture, while smoothing the effects of noise

Bilateral filter:

- Smooth the image while preserving edges



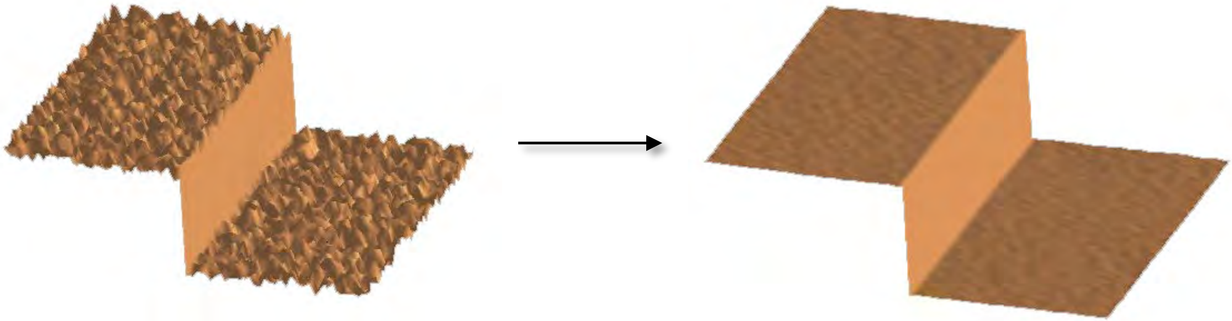
Bilateral Filtering

Advantages:

- Preserve edges in the smoothing process
- Simple and intuitive
- Non-iterative

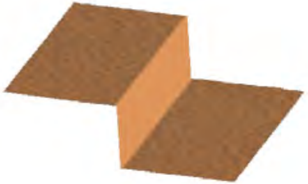
Flaws:

- Complexity (Brute-force): $O(r^2)$
- Gradient distortion



Guided Image Filtering

General Linear Translation-Variant Filtering Process



Output Image q

$$q_i = \sum_j W_{ij}(I)p_j$$

Input Image p

Filter Kernel of I

Guidance Image I

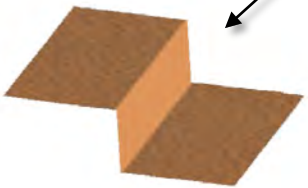
Assumption: Local Linear Model

Local Linear Model between the guidance I and the filter output q:

q is a linear transform of I in a window ω_k centered at the pixel k

$$q_i = a_k I_i + b_k, \forall i \in \omega_k$$

A square window of a radius r



Output Image q

Guidance Image I

Optimization

Goal:

$$\min \| n \| \Leftrightarrow \min \sum_{i \in w_k} \| q_i - p_i \| \Leftrightarrow argmin \sum_{i \in w_k} (a_k I_i + b_k - p_i)^2$$

Noise / Texture

$$q_i = p_i - n_i$$

Output Image q

Input Image p

Optimization

Cost Function:

$$E(a_k, b_k) = \sum_{i \in w_k} \left[(a_k I_i + b_k - p_i)^2 + \epsilon a_k^2 \right]$$

Regularization Parameter

Output Image q

$$q_i = p_i - n_i$$

Input Image p

Optimization

Solution:

$$a_k = \frac{\rho_{Ip}}{\sigma_k^2+\epsilon} = \frac{\frac{1}{|\omega|}\sum_{i\in\omega_k} I_i p_i - \mu_k \overline{p_k}}{\sigma_k^2+\epsilon}$$

$$b_k = \overline{p_k} - a_k \mu_k$$

where:

- ρ_{Ip} is the covariance of I and p,
- μ_k and σ_k are the mean and variance of I in ω_k,
- $|\omega|$ is the number of pixels in ω_k,
- $\overline{p_k} = \frac{1}{|\omega|}\sum_{i\in\omega_k} p_i$ is the mean of p in ω_k.

Filter the Entire Image

Compute the average of $a_k I_i + b_i$ in all ω_k that covers pixel q_i:

$$q_i = \frac{1}{|\omega|} \sum_{k:i \in \omega_k} (a_k I_i + b_k)$$
$$= \bar{a}_i I_i + \bar{b}_i$$

where:

- $\bar{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} a_k$,
- $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$.

Analysis

Analysis

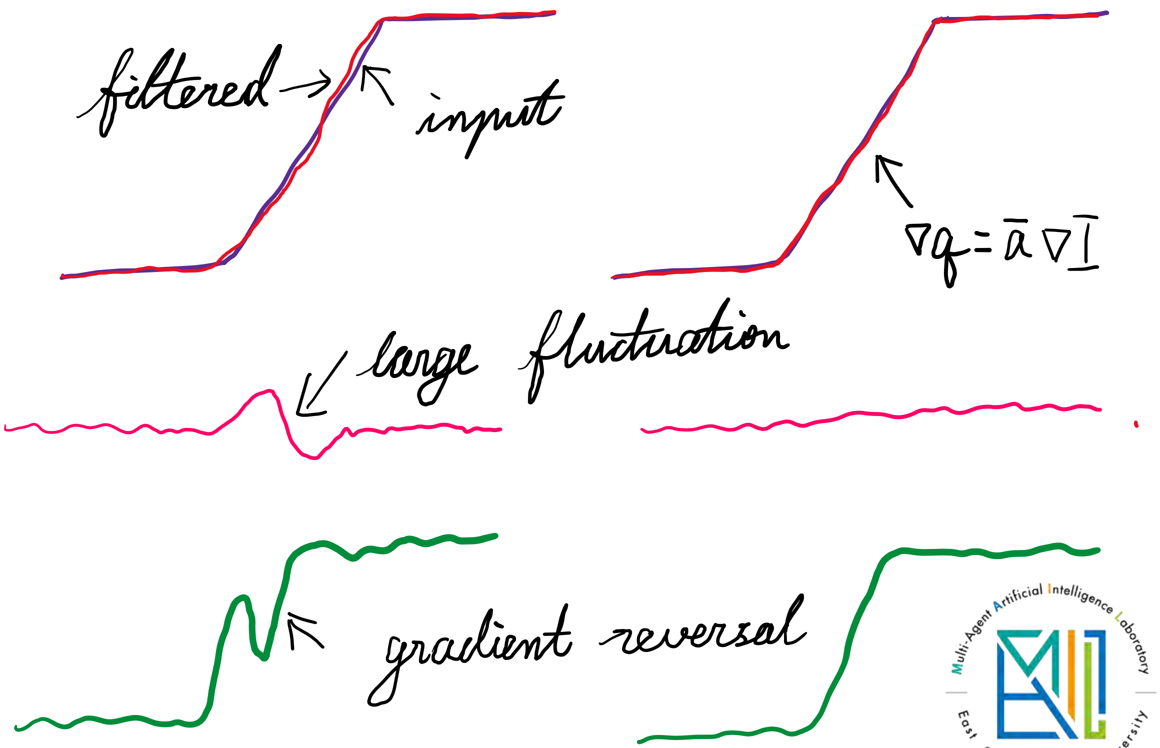
Advantages:

- Edge-preserving filtering
- Non-iterative
- Time complexity: *O(N)*
- Gradient Preserving

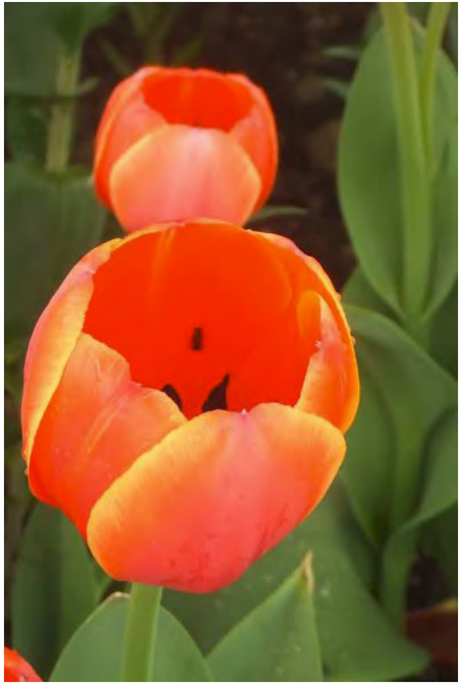
Bilateral Filter | Guided Filter

Detail

Enhanced



Gradient Preserving



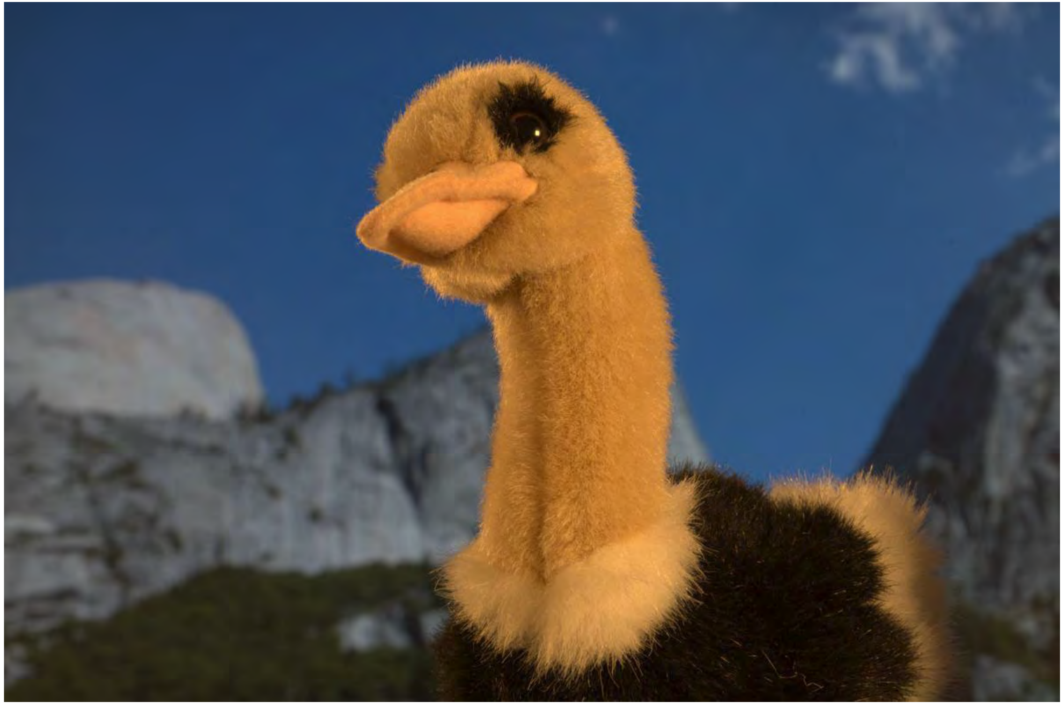
Input $(I = p)$

Gradient reversal

Bilateral Filter

Guided Filter

Feathering

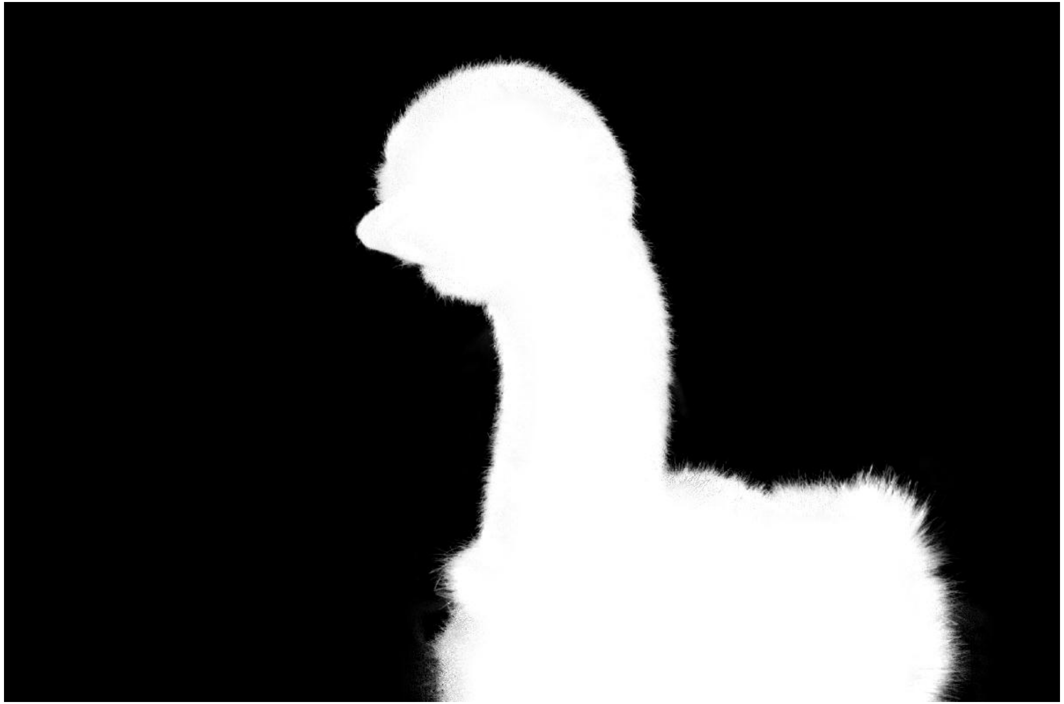


Guide I (size 3000x2000)

Feathering

Filter input p (binary segmentation)

Feathering



filter output q (alpha matte)

Haze Removing

Guide I

Input p

Output q

Improvements & Conclusions

Improvement: Fast Guided Filter

- Subsamples (nearest-neighbor or bilinear) the input p and the guidance I by a ratio s.
- All the box filters are performed on the low-resolution maps, which are the major computation of the guided filter.
- The two coefficient maps $\bar{a}$ and $\bar{b}$ are bilinearly up-sampled to the original size.
- Reduces the time complexity from $O(N)$ to $O(\frac{N}{s^2})$

Improvement: Fast Guided Filter

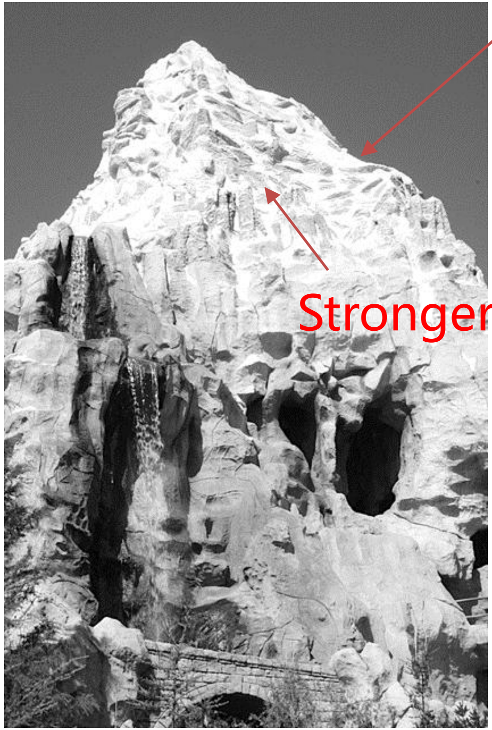
Algorithm 1 Guided Filter.

1: $\text{mean}_I = f_{\text{mean}}(I, r)$
   $\text{mean}_p = f_{\text{mean}}(p, r)$
   $\text{corr}_I = f_{\text{mean}}(I .* I, r)$
   $\text{corr}_{Ip} = f_{\text{mean}}(I .* p, r)$
2: $\text{var}_I = \text{corr}_I - \text{mean}_I .* \text{mean}_I$
   $\text{cov}_{Ip} = \text{corr}_{Ip} - \text{mean}_I .* \text{mean}_p$
3: $a = \text{cov}_{Ip} ./ (\text{var}_I + \epsilon)$
   $b = \text{mean}_p - a .* \text{mean}_I$
4: $\text{mean}_a = f_{\text{mean}}(a, r)$
   $\text{mean}_b = f_{\text{mean}}(b, r)$
5: $q = \text{mean}_a .* I + \text{mean}_b$

Algorithm 2 Fast Guided Filter.

1: $\underline{I' = f_{\text{subsample}}(I, s)}$
   $\underline{p' = f_{\text{subsample}}(p, s)}$
   $\underline{r' = r/s}$
2: $\text{mean}_I = f_{\text{mean}}(I', r')$
   $\text{mean}_p = f_{\text{mean}}(p', r')$
   $\text{corr}_I = f_{\text{mean}}(I' .* I', r')$
   $\text{corr}_{Ip} = f_{\text{mean}}(I' .* p', r')$
3: $\text{var}_I = \text{corr}_I - \text{mean}_I .* \text{mean}_I$
   $\text{cov}_{Ip} = \text{corr}_{Ip} - \text{mean}_I .* \text{mean}_p$
4: $a = \text{cov}_{Ip} ./ (\text{var}_I + \epsilon)$
   $b = \text{mean}_p - a .* \text{mean}_I$
5: $\text{mean}_a = f_{\text{mean}}(a, r')$
   $\text{mean}_b = f_{\text{mean}}(b, r')$
6: $\underline{\text{mean}_a = f_{\text{upsample}}(\text{mean}_a, s)}$
   $\underline{\text{mean}_b = f_{\text{upsample}}(\text{mean}_b, s)}$
7: $q = \text{mean}_a .* I + \text{mean}_b$

Limitations



Input

Guided filter

Conclusions

- Guided Filter
  - Edge-preserving filtering
  - Non-iterative
  - Time complexity: *O(N)*
  - Gradient Preserving

- Improvements
  - Fast Guided Filter [He et al., '15]: time complexity of $O(\frac{N}{s^2})$

Thank You