Multiple View Image Denoising using 3D Focus Image Stacks

Shiwei Zhou, Yu Hen Hu, Hongrui Jiang
University of Wisconsin-Madison
Outline

• Introduction
• Related Work
• Denoising Scheme
  – 3D Focus Image Stack
  – Disparity Map Estimation
  – Preliminary Denoising
  – Reliability Map
  – Handling Unreliable Pixels
• Experiments
Motivation

- CPS project: smart flexible camera sheet [6]
  - Ultra-thin semantic-guided cooperative micro-camera array

![Diagram showing miniaturization and cooperative micro-camera arrays]
Introduction

- Multiple view imaging
Introduction

- Multiple view imaging
  - 3D scene reconstruction
Introduction

- Multiple view imaging
  - 3D scene reconstruction
  - Object tracking and recognition
Introduction

- Multiple view imaging
  - 3D scene reconstruction
  - Object tracking and recognition
  - Environmental surveillance
Introduction

- Multiple view imaging
  - 3D scene reconstruction
  - Object tracking and recognition
  - Environmental surveillance
  - 3DTV
  - Etc.
Introduction

- Multiple view imaging
  - 3D scene reconstruction
  - Object tracking and recognition
  - Environmental surveillance
  - 3DTV
  - Etc.

- Multi-views: helps exploit redundancy and 3D information
Introduction

- Capture: array of cameras
  - Small miniature cameras
  - Limited exposure – avoid motion blur
  - Small aperture – large depth of field
Introduction

• Capture: array of cameras
  - Small miniature cameras
  - Limited exposure – avoid motion blur
  - Small aperture – large depth of field

• Drawback: noisy image
Related Work

- Single view image denoising
Related Work

- Single view image denoising

Gaussian Filtering

Related Work

- Single view image denoising

Bilateral Filtering

Related Work

- Single view image denoising

Related Work

- Single view image denoising

Related Work

- Multi-view image denoising
Related Work

- Multi-view image denoising

Zhang et al., “Multiple View Image Denoising”
Related Work

- Multi-view image denoising

Related Work

- Multi-view image denoising

Miyata et al., “Fast Multi-view Image Denoising Based on Image Reconstruction by Plane Sweeping”
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Multi-view Images

- Multi-view images $I_{s,t}(x, y)$
  - $x, y$ – image coordinates
  - $s, t$ – camera coordinates
Disparity
Disparity

(left camera) $(x_l, y_l)$

(right camera) $(x_r, y_r)$

$(X, Y, Z)$
Disparity

left image

\((x_l, y_l)\)

right image

\((x_r, y_r)\)
Disparity

\[(x_r, y_r) \quad d \quad (x_l, y_l)\]
Disparity

- Disparity: the distance between corresponding points in the left and right image
Disparity
Disparity

left camera

(right camera

$$(x_l, y_l)$$

$$(x_r, y_r)$$

$$(x'_r, y'_r)$$

$$(X, Y, Z)$$

$$(X', Y', Z')$$
Disparity

left image

\((x_i, y_i)\)

right image

\((x_r, y_r)\) \((x'_r, y'_r)\)
Disparity

\[(x_r, y_r)(x'_r, y'_r) (x_l, y_l)\]
Disparity

- Disparity is inversely proportional to depth
Denoising Scheme

1. Acquire multi-view images of the same scene
2. Construct 3D focus image stacks
3. Disparity map estimation
4. Preliminary denoising using disparity map
5. Compute reliability map
6. Handle unreliable pixels
3D Focus Image Stacks
3D Focus Image Stacks

\[(x_{0}, y_{0}) \quad (x_{1}, y_{1}) \quad (x_{+1}, y_{+1})\]
3D Focus Image Stacks

- For each disparity $d$,
  - Translate each pixel using
  
  $I_{s,t}^d(x, y) = I_{s,t}(x + (s - s_0)d, y + (t - t_0)d)$

- Pixels with correct disparity,
  - Corresponding pixels are stacked
  - Clear image, a.k.a. in-focus
For each disparity $d$,
- Translate each pixel using

$$I_{s,s}^d(x, y) = I_{s,s}(x + (s - s_0)d, y + (t - t_0)d)$$

- Pixels with incorrect disparity,
  - Corresponding pixels have displacement
  - Blurred image, a.k.a. out-of-focus

$$s = +1 \quad s = 0 \quad s = -1$$

$$(x_0, y_0) \quad (x_{+1}, y_{+1}) \quad (x_{-1}, y_{-1})$$
3D Focus Image Stacks

- Stack $I_{s,t}^d(x, y)$ into $F_d(x, y)$
  - $F_d$ is called 3D focus image stacks

Number of views
3D Focus Image Stack

- Visualization by simple averaging
Depth Estimation

- Similarity measure

\[ S_d(x, y) = \frac{1}{N} \sum_{k=1}^{N} \sum_{(i,j) \in W_{i,j}} |F_d(i, j, k) - I_{s_{d_{0}}} (i, j)| \]

- \( N \) – Number of views in focus image stack
- \( k \) – \( k \)th layer (view) of the focus image stack
- \( W_{i,j} \) – Window centered at pixel \((i, j)\)
Depth Estimation
Depth Estimation
Depth Estimation
Depth Estimation
Depth Estimation
Depth Estimation

$S_d \quad d = 14$
Depth Estimation

- Similarity measure

\[ S_d(x, y) = \frac{1}{N} \sum_{k=1}^{N} \sum_{\{(i, j) \in W_{x,y}\}} \left| F_d(i, j, k) - I_{s_{x,y}}(i, j) \right| \]

- Examples for different values of \(d\): 
  - \(d = 5\)
  - \(d = 6\)
  - \(d = 10\)
  - \(d = 14\)
Depth Estimation

- Disparity map \( d(x, y) = \arg \min_d S_d(x, y) \)
Preliminary Denoising

\[ I_{est}(x, y) = \frac{1}{W_{sum}} \sum_{k=1}^{N} w_k F_d(x, y)(x, y, k) \]
Preliminary Denoising

\[ I_{est}(x, y) = \frac{1}{w_{sum}} \sum_{k=1}^{N} w_k F_{d(x,y)}(x, y, k) \]

- For each pixel \((x, y)\):
  - Find its disparity \(d\)
For each pixel \((x, y)\):
- Find its disparity \(d\)
- Extract corresponding focus image stack \(F_d\)
  - This pixel will be in-focus at this disparity

\[
I_{est}(x, y) = \frac{1}{w_{sum}} \sum_{k=1}^{N} w_k F_{d(x,y)}(x, y, k)
\]

Preliminary Denoising
Preliminary Denoising

\[ I_{est}(x, y) = \frac{1}{W_{sum}} \sum_{k=1}^{N} w_k F_d(x, y)(x, y, k) \]

- For each pixel \((x, y)\):
  - Find its disparity \(d\)
  - Extract corresponding focus image stack \(F_d\)
    - This pixel will be in-focus at this disparity
  - Take weighted average of each pixel at \((x, y)\) in \(F_d\)
    - \(N\) – Number of views in \(F_d\)
    - \(k\) – \(k^{th}\) layer (view) of \(F_d\)
    - \(w_k\) – Weight of each layer (view) in \(F_d\) depending on distance to the center view
Preliminary Denoising
Preliminary Denoising
Preliminary Denoising
Preliminary Denoising

\[ d = 14 \]
Preliminary Denoising

\[ d = 14 \]
Preliminary Denoising

$d = 14$
Preliminary Denoising
Preliminary Denoising
Preliminary Denoising

\[ d = 14 \]

\[ F_{14}(x, y, 1) \ldots \]

\[ \ldots \]

\[ \ldots F_{14}(x, y, N) \]

\[ I_{est}(x, y) \]

\[ w_1 \]

\[ \ldots \]

\[ w_N \]
Preliminary Denoising

Noiseless true image

Noisy image

Preliminarily denoised image

Reliability map (binarized)
Preliminary Denoising

- Noiseless true image
- Noisy image
- Preliminarily denoised image
- Reliability map (binarized)
Reliability Evaluation

- Reliability map
  \[ R(x, y) = \frac{S_d(x, y)(x, y)}{L^2} \]
  - \( S_d \) - Similarity measure
  - \( L \) - Window size for computing \( S_d \)
  - Binarize \( R(x, y) \) using a threshold
  - Morphological transformations
    - Dilation and erosion
    - Remove isolated outliers
Reliability Evaluation

- Reliability map
  \[ R(x, y) = \frac{S_d(x, y)(x, y)}{L^2} \]
  - \( S_d \) - Similarity measure
  - \( L \) - Window size for computing \( S_d \)
  - Binarize \( R(x, y) \) using a threshold
  - Morphological transformations
    - Dilation and erosion
    - Remove isolated outliers
Handling Unreliable Pixels

- Non-local means

\[ NL(i) = \sum_{j \in W_i} w(i, j) I(j) \]

\[ w(i, j) = \frac{1}{Z(i)} \exp\left( -\frac{\| I(N_i) - I(N_j) \|^2}{h^2} \right) \]
Handling Unreliable Pixels

- Non-local means

\[ NL(i) = \sum_{j \in W_i} w(i, j) I(j) \]

\[ w(i, j) = \frac{1}{Z(i)} \exp \left( -\frac{\| I(N_i) - I(N_j) \|^2}{h^2} \right) \]
Handling Unreliable Pixels

- Non-local means

\[ NL(i) = \sum_{j \in W_i} w(i, j) I(j) \]

\[ w(i, j) = \frac{1}{Z(i)} \exp \left( - \frac{\| I(N_i) - I(N_j) \|^2}{h^2} \right) \]
Handling Unreliable Pixels

- Non-local means

\[
NL(i) = \sum_{j \in W_i} w(i, j) I(j)
\]

\[
w(i, j) = \frac{1}{Z(i)} \exp \left( -\frac{\|I(N_i) - I(N_j)\|^2}{h^2} \right)
\]
Experiments

- **PSNR (dB)**

\[
PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right)
\]

- **MSE** – mean squared error between denoised image and original image
- **MAX** – maximum possible pixel value, i.e. 255

- **Visual comparison**

Ohta

Knight

Tarot

Truck
Experiments

**PSNR (dB)**

- **NLM**, **Miyata et al.**, **Proposed**

**Running Time (s)**

- **NLM**, **Miyata et al.**, **Proposed**
Thank you!