# Denoising

By Qun Feng Tan

Psych 221 Project

Winter 2008

# Motivation

- Acquisition of Image is usually noisy
- Application of denoising algorithms prior to image processing algorithms
- Each denoising algorithm has pros and cons
  - Median Filter
  - Forward/Backward Recursive Algorithm
  - Discrete Universal DEnoising (DUDE)

# Algorithm 1: Median Filter

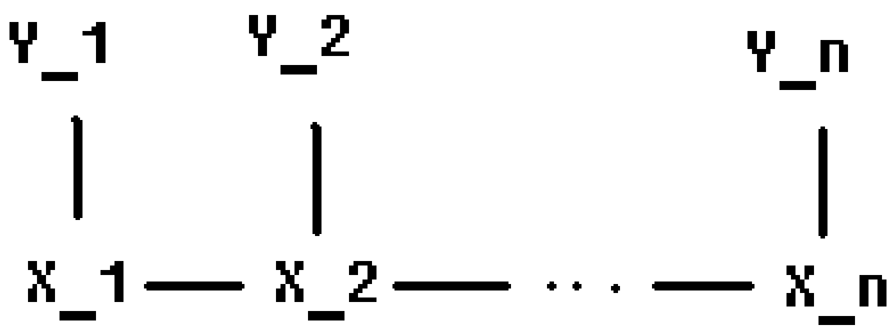
- Non-linear, sliding window technique
- Edge effects
- Example:  [12 55 23 1 7]  with window size 3
  - Output[1] = Median[12 12 55] = 12
  - Output[2] = Median[12 55 23] = 23
  - Output[3] = Median[55 23 1] = 23
  - Output[4] = Median[23 1 7] = 7
  - Output[5] = Median[1 7 7] = 7

# Algorithm 2: Hidden Markov Models

- *A Markov Chain:*

$$P(X, Z \mid Y) = P(X \mid Y) P(Z \mid Y)$$

- *Hidden Markov Process:*



# Algorithm 2: Hidden Markov Models (Continued)

- Forward/Backward recursive algorithm
  - Derivation based on properties of probability densities and markovity
  - Goal: Determine $P(x_t \mid y_{1\to t-1})$ for t = 1,…,n

# Algorithm 2: Hidden Markov Models – Forward Recursion

1) Initialization:

i. Some initial distribution $P(x_1 \mid y_{1\to 0}) \doteq P(x_1)$

→ Setup to propagate the recursion.

ii. Markov kernel: $P(x_t \mid x_{t-1})$

iii. Corruption channel: $P(y_t \mid x_t)$

iv. Noisy observations: $Y_1, Y_2, Y_3 \ldots$

# Algorithm 2: Hidden Markov Models – Forward Recursion

2) Recursive step

For t = 1 to n step 1

$$P(x_t \mid y_{1\to t}) = \frac{P(x_t \mid y_{1\to t-1})P(y_t \mid x_t)}{\sum_{\widetilde{x}_t} P(\widetilde{x}_t \mid y_{1\to t-1})P(y_t \mid \widetilde{x}_t)}$$

(Measurement update)

$$P(x_{t+1} \mid y_{1\to t}) = \sum_{x_t} P(x_t \mid y_{1\to t})P(x_{t+1} \mid x_t, y_{1\to t})$$

(Time update)

End For-loop

# Algorithm 2: Hidden Markov Models – Backward Recursion

Goal: Determine $P(x_t \mid y_{1 \to n})$ for t = 1,…,n

1 - Initialization:

i.From the Forward Recursive Algorithm we have

$$P(x_t \mid y_{1 \to t})$$

for t = 1,…,n

ii. Markov kernel:

$$P(x_t \mid x_{t-1})$$

# Algorithm 2: Hidden Markov Models – Backward Recursion

2 - Recursive step

For t = n-1 to 1 step – 1

$$P(x_t \mid y_{1\to n}) = \sum_{x_{t+1}} \frac{P(x_t \mid y_{1\to t})P(x_{t+1} \mid x_t)}{\sum_{\widetilde{x_t}} P(\widetilde{x_t} \mid y_{1\to t})P(x_{t+1} \mid \widetilde{x_t})} P(x_{t+1} \mid y_{1\to n})$$

End For-loop

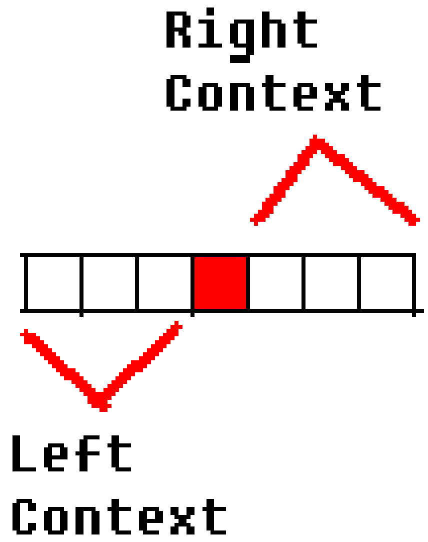
# Algorithm 3: Discrete Universal DEnoising Algorithm (DUDE)

- No assumptions about underlying probability distribution
- Channel statistics known
- A loss function is specified
  - Hamming Loss:

$$1 \text{ if } \; x \neq \hat{x}, \; 0 \text{ if } \; x = \hat{x}$$

# Algorithm 3: Discrete Universal DEnoising Algorithm (DUDE)

Step 1: First pass through the data set – To compute count vectors




$$m(a^n, b^k, c^k)[\beta] = |\{i : k+1 \le i \le n-k, a_{i-k \to i+k} = b^k \beta c^k\}|$$

# Algorithm 3: Discrete Universal DEnoising Algorithm (DUDE)

<u>Step 2: Second Pass through the data set – Denoising step</u>

We correct each pixel according to this rule:

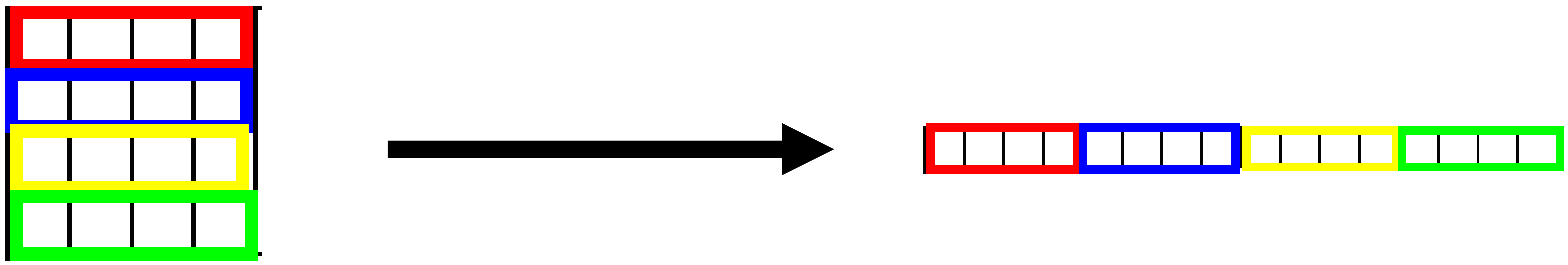
$$\arg\min_{\hat{x}\in A} m^T(y^n, b^k, c^k)\Pi^T[\Pi\Pi^T]^{-1}[\lambda_{\hat{x}} \odot \pi_{z_i}]$$

where $\Pi = [\pi_1 | \ldots | \pi_M]$ and $\Lambda = [\lambda_1 | \ldots | \lambda_M]$

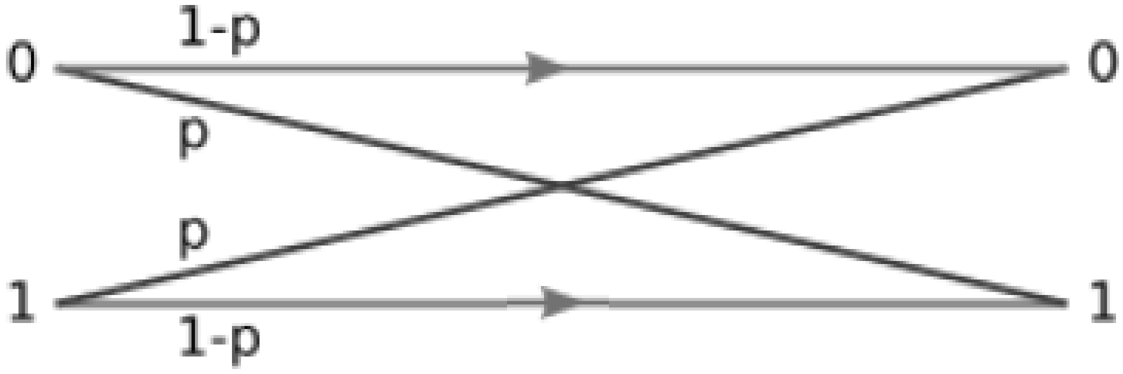
$\odot$ represents component-wise multiplication.

# Proof-of-Concept

- Application on Bi-level Images



# Corruption Mechanism




$$\Pi = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

# Simplification of DUDE

If $x = 1$

If $\dfrac{m(y^n, b^k, c^k)[x]}{m(y^n, b^k, c^k)[x] + m(y^n, b^k, c^k)[\bar{x}]} > 2p(1-p)$

$\hat{x} = 1$

Else

$\hat{x} = 0$

End If

Else

If $\dfrac{m(y^n, b^k, c^k)[\bar{x}]}{m(y^n, b^k, c^k)[x] + m(y^n, b^k, c^k)[\bar{x}]} > 2p(1-p)$

$\hat{x} = 0$

Else

$\hat{x} = 1$

End if

End if

# Results

- Original Image:

# Results

- Noisy Image (p=0.25):

# Results

- Median Filter (Window length = 9):

post_error_rate =0.0986

# Results

- Forward/Backward Recursive Algorithm
- 3 iterations

post_error_rate =0.0589

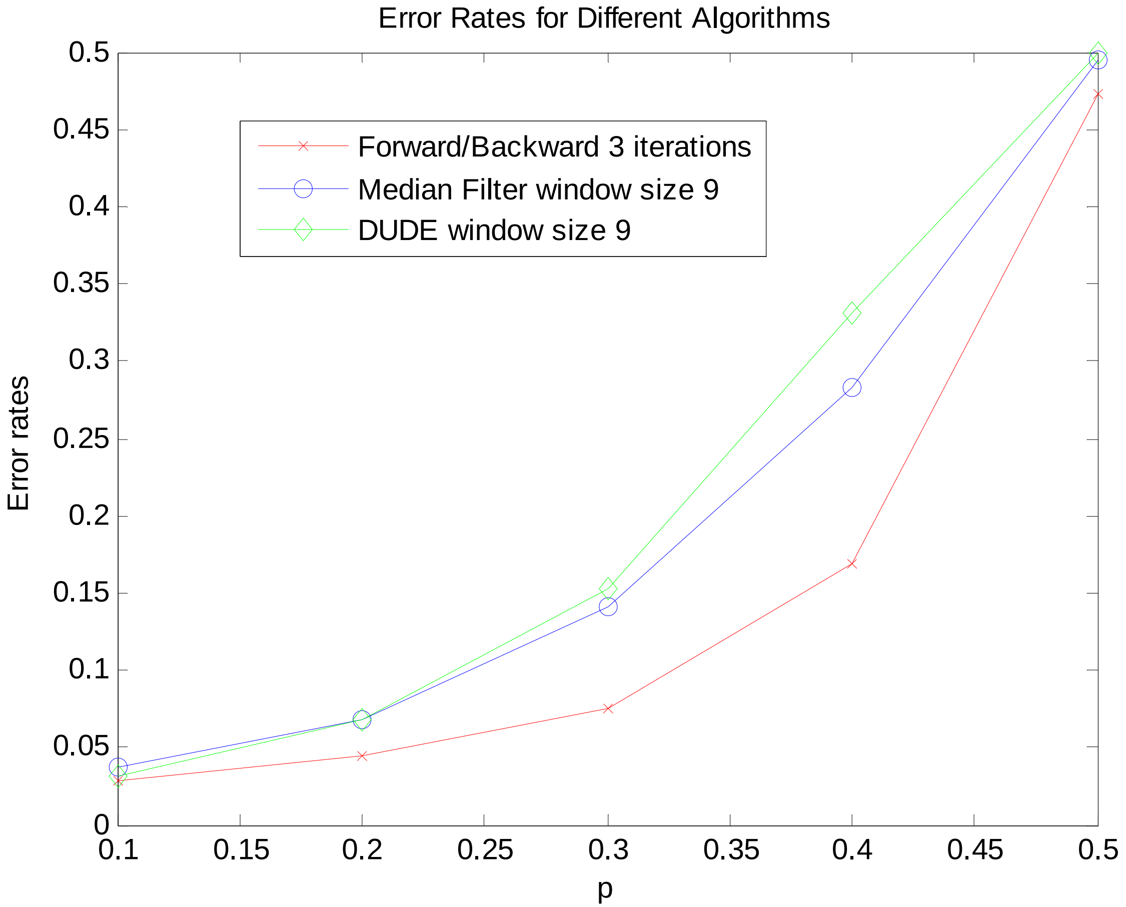
# Results

- DUDE (Window length = 9)

post_error_rate = 0.0995

# Results

- Performance Analysis



# Conclusion

- Median filter decent – easy to implement
- Forward/Backward best – but huge storage
- DUDE is pretty amazing given assumptions
  - Has rooms for expansion
    - Eg. Continuous-tone images

# Thank You!

- Questions???