CSE6706:

Advanced Digital Image Processing

Dr. Md. Monirul Islam

CSE-BUET

Types of Region Based Segmentation

- Region growing
- Region splitting and merging

Region Splitting and Merging

- Main Idea:
  - Subdivide an image into arbitrary regions
  - Split and merge the existing regions based some criteria

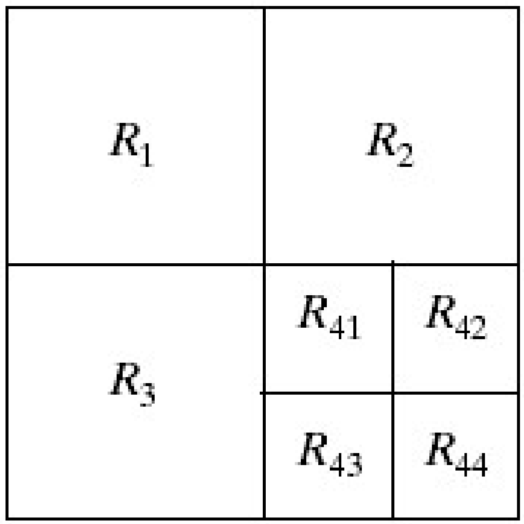
CSE-BUET

Region Splitting and Merging

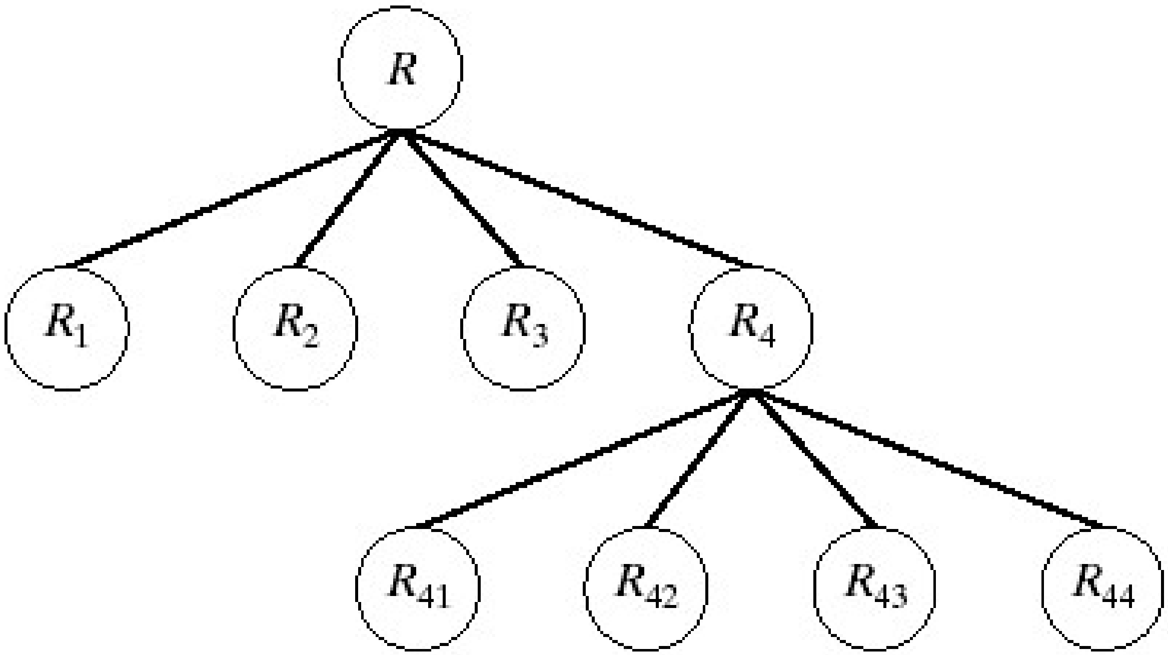
- Let,
  - R be the entire region/image
  - P be a predicate on R

- Splitting:
  - Start with R
  - If $P(R)$=FALSE
    - subdivide R successively into smaller and smaller quadrant regions R_i so that $P(R_i)$=TURE
  - for any quadrant R_i, if $P(R_i)$=FALSE, divide it into sub-quadrants, and so on

Region Splitting and Merging



An image, R



Quad Tree representation

CSE-BUET

Region Splitting and Merging

- Merging:
  - Merge two regions R_j and R_k if $P(R_j \cup R_k)$=TRUE

Region Splitting and Merging (1)

1. Split into four disjoint quadrants any region R_i for which $P(R_i)$=FALSE
2. When no splitting possible, merge any adjacent regions R_j and R_k if P(R_j U R_k)=TRUE
3. Stop when no merging is possible

Region Splitting and Merging (2)

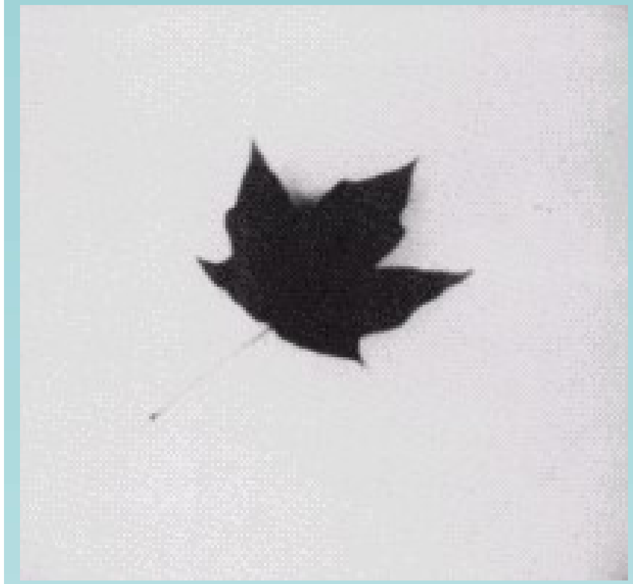
1. Split the image initially into set of blocks
2. Initial merging is limited to group of four blocks that are descendent and satisfy predicate
3. When no merging is possible as step 2, do final mergings as follows

   3.1 Merge any adjacent regions R_j and R_k if $P(R_j \cup R_k)$=TRUE

Region Splitting and Merging

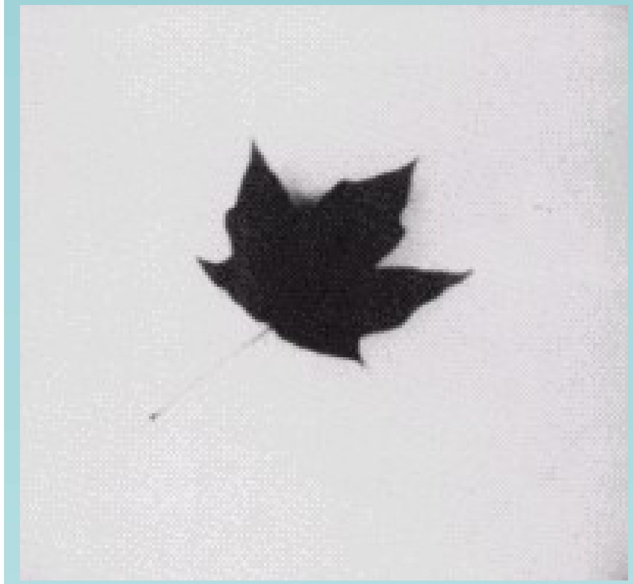
- Predicate: $P(R_i)$=TRUE if
  - At least 80% of pixels in R_i satisfy $|z-m_i|<2\sigma_i$, where
  - z: gray level of an pixel in R_i
  - m_i: average gray level in R_i
  - σ_i: standard deviation of gray level in R_i

Region Splitting and Merging



- Predicate: $P(R_i)$=TRUE if
  - At least 80% of pixels in R_i satisfy $|z-m_i|<2\sigma_i$, where
  - z: gray level of an pixel in R_i
  - m_i: average gray level in R_i
  - σ_i: standard deviation of gray level in R_i
- If satisfy, assign m_i to all pixel in R_i

Region Splitting and Merging

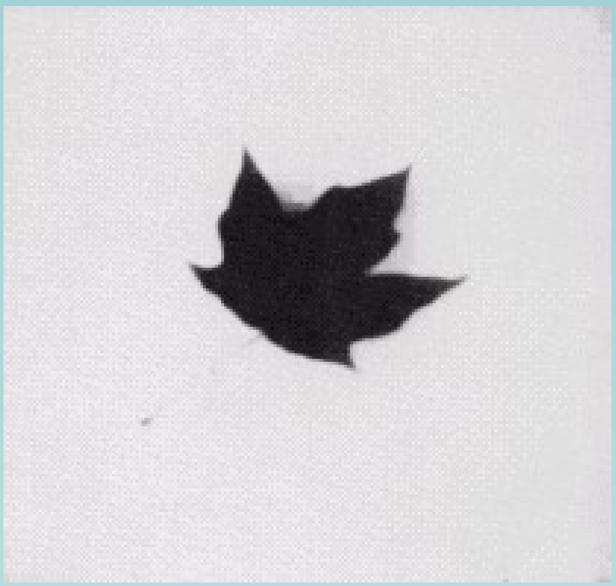


Using thresholding

CSE-BUET

Region Splitting and Merging

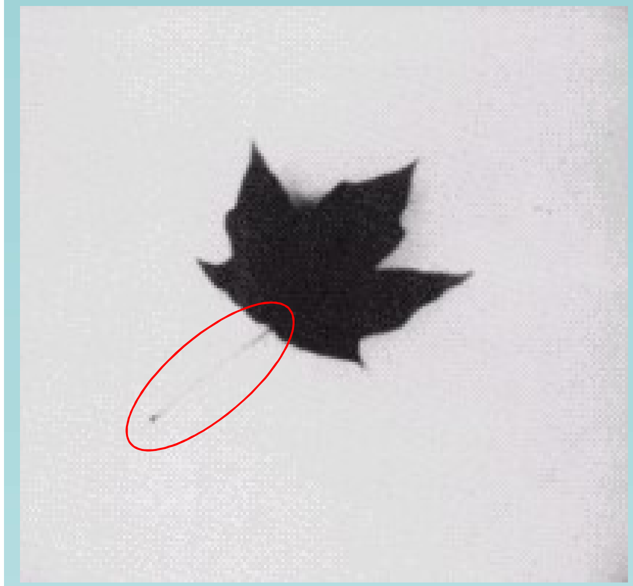
Using thresholding



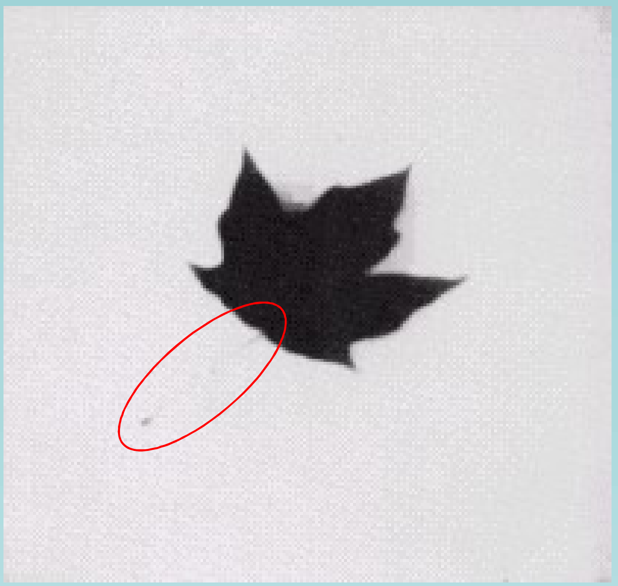
After region splitting and merging

CSE-BUET

Region Splitting and Merging



Using thresholding



After region splitting and merging

Motion in Segmentation

Motion

- relative displacement between imaging device and an object
- extracted from a sequence of image frames
- useful to find an object from a background of irrelevant detail

Motion in Segmentation

- Two way to extract motion:
  - Spatial domain technique
  - Frequency domain technique

Motion Extraction in Spatial Domain

- Motion is detected by monitoring the changes in subsequent image frames
- Compare pixel by pixel gray differences between two frames

Motion Extraction in Spatial Domain

- Let there be a reference frame consisting of stationary objects
- Other frames have objects with motion
- Find *diff_image = new_frame* MINUS *ref_image*
  - Canceling everything except nonzero entries corresponding to moving object

Motion Extraction in Spatial Domain

- Let there be two frames $f(x, y, t_{t_i})$ and $f(x, y, t_{t_j})$
- The difference is given by,

$$d_{i,j}(x, y) = \begin{cases} 1 & \text{if } \left| f(x, y, t_{t_i}) - f(x, y, t_{t_j}) \right| > T \\ 0 & \text{otherwise} \end{cases}$$

Motion Extraction in Spatial Domain

$$d_{i,j}(x,y) = \begin{cases} 1 & \text{if } \left| f(x,y,t_{t_i}) - f(x,y,t_{t_j}) \right| > T \\ 0 & \text{otherwise} \end{cases}$$

- All pixels in $d_{i,j}(x,y)$ having value 1 correspond to moving object(s)

Motion Extraction in Spatial Domain

$$d_{i,j}(x,y)=\begin{cases}1 & \text{if } \left|f(x,y,t_{t_i})-f(x,y,t_{t_j})\right|>T\\ 0 & \text{otherwise}\end{cases}$$

- All pixels in $d_{i,j}(x,y)$ having value 1 correspond to moving object(s)
- However, noise can also contribute to $d_{i,j}(x,y)$

CSE-BUET

Motion Extraction in Spatial Domain

$$d_{i,j}(x,y) = \begin{cases} 1 & \text{if } \left| f(x,y,t_{t_i}) - f(x,y,t_{t_j}) \right| > T \\ 0 & \text{otherwise} \end{cases}$$

- All pixels in $d_{i,j}(x,y)$ having value 1 correspond to moving object(s)
- However, noise can also contribute to $d_{i,j}(x,y)$
- Remedy: use 4- or 8-connecivity check to find isolated noise

CSE-BUET

Motion Extraction in Spatial Domain

$$d_{i,j}(x,y)=\begin{cases}1 & \text{if } \left|f(x,y,t_{t_i})-f(x,y,t_{t_j})\right|>T\\ 0 & \text{otherwise}\end{cases}$$

- All pixels in $d_{i,j}(x,y)$ having value 1 correspond to moving object(s)
- However, noise can also contribute to $d_{i,j}(x,y)$
- Remedy: use 4- or 8-connecivity check to find isolated noise
- This can cancel out small / slow moving object as well

CSE-BUET

Motion Extraction in Spatial Domain

- Alternatively, monitor the change pattern of pixel values over *many* frames
- That means we need to use some *memory*

CSE-BUET

Motion Extraction in Spatial Domain

- Alternatively, monitor the change pattern of pixel values over *many* frames
- That means we need to use some *memory*
- The changes are accumulated in memory called *accumulative difference image* (ADI)

Motion Extraction in Spatial Domain

- The changes are accumulated in memory called *accumulative difference image* (ADI)
- Let there are fames $f(x, y, t_1), f(x, y, t_2), \cdots, f(x, y, t_n)$
- For simplicity, let they are $f(x, y, 1), f(x, y, 2), \cdots, f(x, y, n)$
- Let, $R(x,y) = f(x, y, 1)$

CSE-BUET

Motion Extraction in Spatial Domain

- The *accumulative difference image* (ADI) is calculated as

$$A_k(x,y)=\begin{cases} A_{k-1}(x,y)+1 & \text{if } \left|R(x,y)-f(x,y,k)\right|>T \\ A_{k-1}(x,y) & \text{otherwise} \end{cases}$$

Motion Extraction in Spatial Domain

$$A_k(x,y)=\begin{cases}A_{k-1}(x,y)+1 & \text{if } \left|R(x,y)-f(x,y,k)\right|>T\\ A_{k-1}(x,y) & \text{otherwise}\end{cases}$$

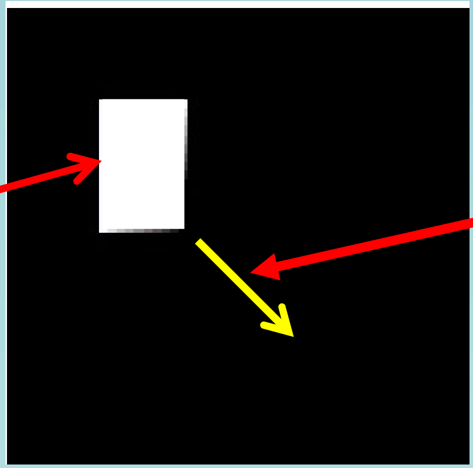
- Other two accumulators:

$$P_k(x,y)=\begin{cases}P_{k-1}(x,y)+1 & \text{if } R(x,y)-f(x,y,k)>T\\ P_{k-1}(x,y) & \text{otherwise}\end{cases}$$

$$N_k(x,y)=\begin{cases}N_{k-1}(x,y)+1 & \text{if } R(x,y)-f(x,y,k)<-T\\ N_{k-1}(x,y) & \text{otherwise}\end{cases}$$

Motion Extraction in Spatial Domain

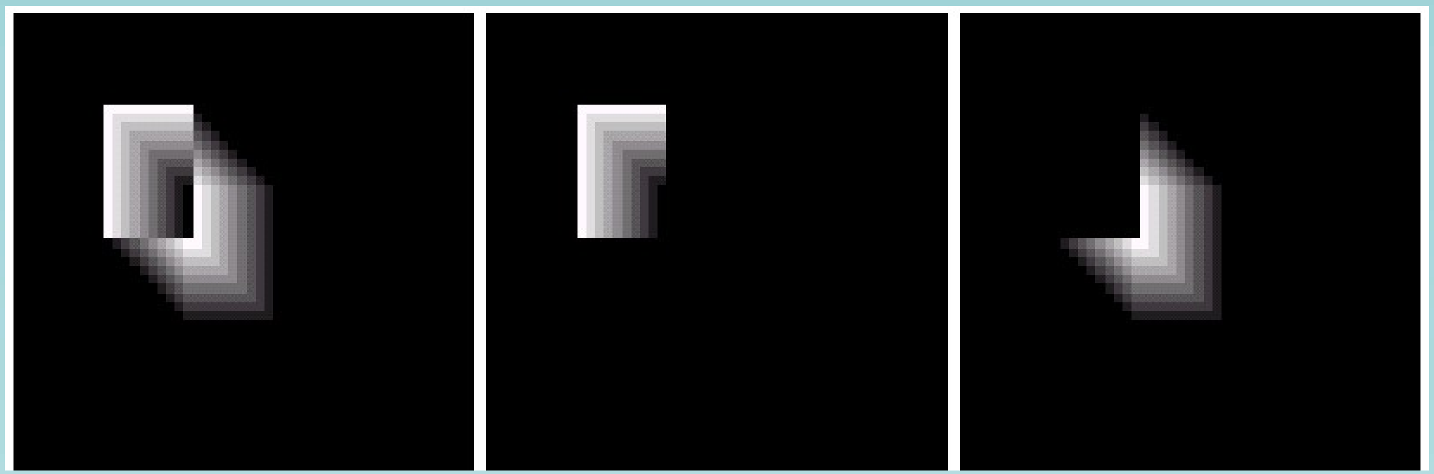
A frame (256X256)



Moving object (75X50)

Moving towards southeastern at $5\sqrt{2}$ pixels per frame

Motion Extraction in Spatial Domain



Absolute Positive Negative

- Image 256X256
- Moving object: 75X50
- Moving towards southeastern at $5\sqrt{2}$ pixels per frame

CSE-BUET

Finding a reference image

- When the moving object in a frame completely moved out of its original position in a reference image,
  - Copy the background to the reference

CSE-BUET

Motion Extraction and Filtering in Spectral/Frequency Domain

CSE-BUET

Motion Extraction in Spectral/Frequency Domain

- Background:
  - Fourier transform

CSE-BUET

Fourier Series

‘Any function that *periodically repeats itself* can be expressed as the *sum of sines and/or cosines* of different frequencies’

- Joseph Fourier

$$f(t) = \sum_{n=-\infty}^{n=\infty} c_n e^{j\frac{2\pi n}{T}t}$$

with T = period of $f(t)$

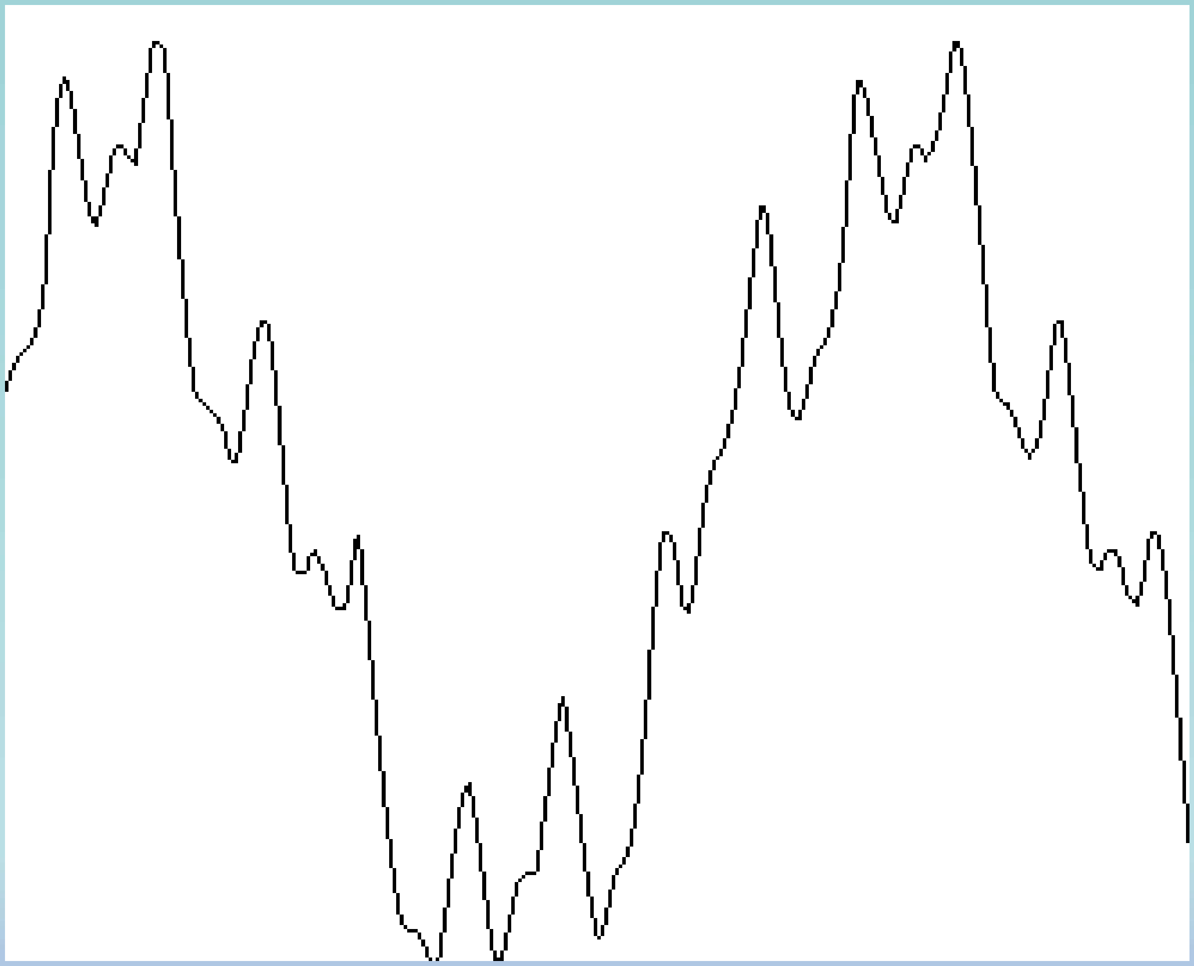
Fourier Series

Fourier says:

Any function that *periodically repeats itself* can be expressed as the *sum of sines and/or cosines* of different frequencies

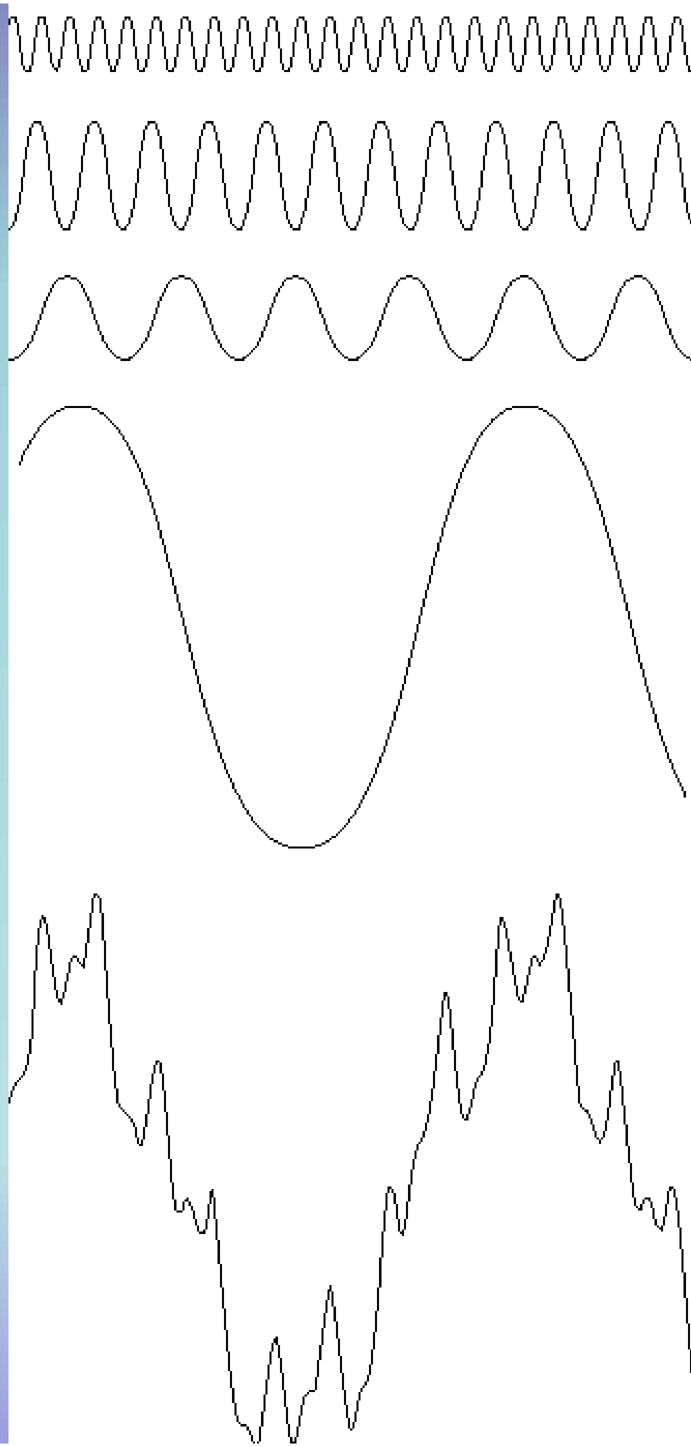
- Doesn't matter how complicated the function is!
- The summation is called *Fourier Series*

Fourier Series



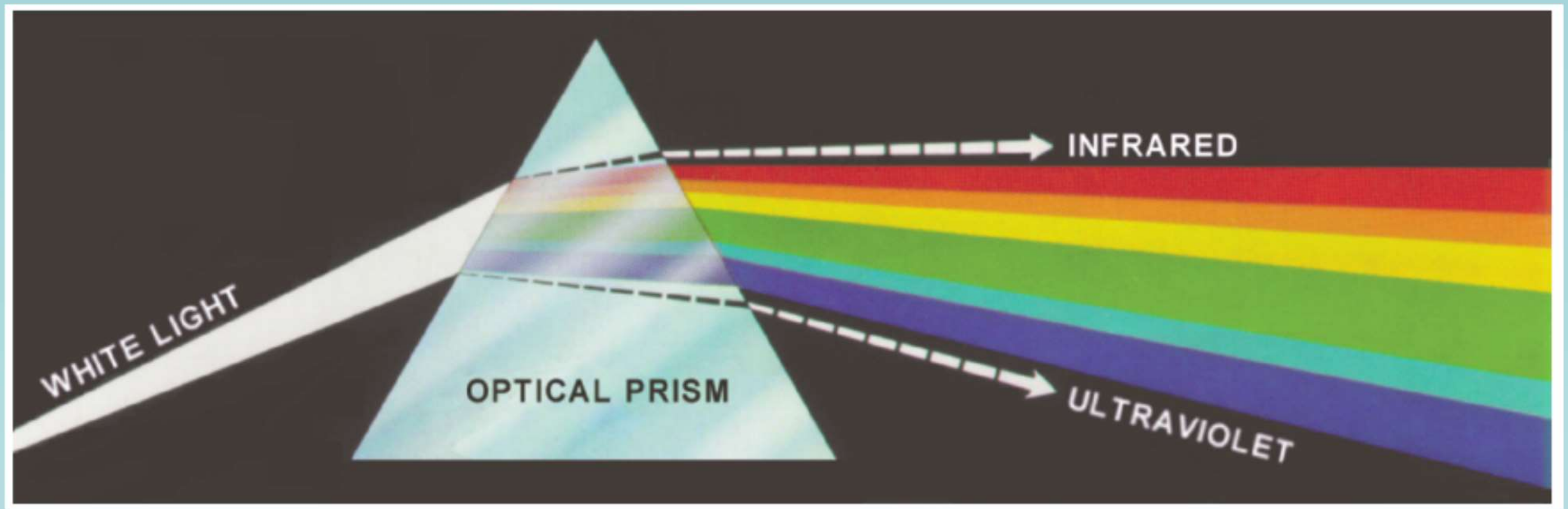
A complicated function

CSE-BUET



CSE-BUET

Fourier Series and Light Spectrum



A complicated function

Individual functions

CSE-BUET

Fourier for Aperiodic Function

- Even *aperiodic* function but *with finite area under curve* can be represented as *integral* of sines and cosines

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ut} du$$

Fourier for Aperiodic Function

- Even *aperiodic* function but *with finite area under curve* can be represented as *integral* of sines and cosines

- This integral is called *Inverse Fourier Transform*

Fourier Transform

$$F(u) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ut} dt$$

Fourier Transform/Series

- Important property:
  - A function *represented in Fourier transform or Series* can be *reconstructed (recovered)* by an *inverse process*

1D Fourier Series and its Inverse

$$f(t) = \sum_{n=-\infty}^{n=\infty} c_n e^{j\frac{2\pi n}{T}t}$$

- The inverse Fourier series is

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-j\frac{2\pi n}{T}t}$$

1D Fourier Transform and its Inverse

$$F(u) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ut} dt$$

- The inverse Fourier transform is

$$f(t) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ut} du$$

FT of Some Common Functions:

Impulse Function

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

Continuous Impulse Function

$$\delta(t) = \begin{cases} \infty & \text{if } \; t = 0 \\ 0 & \text{if } \; t \neq 0 \end{cases}$$

with the constraint,

$$\int_{-\infty}^{\infty} \delta(t)\,dt = 1$$

CSE-BUET

Sifting Property of Continuous Impulse Function

$$\int_{-\infty}^{\infty} \delta(t) f(t) dt = f(0)$$

CSE-BUET

Sifting Property of Continuous Impulse Function

$$\int_{-\infty}^{\infty} \delta(t) f(t) dt = f(0)$$

- Evaluates the function at the location of the impulse

CSE-BUET

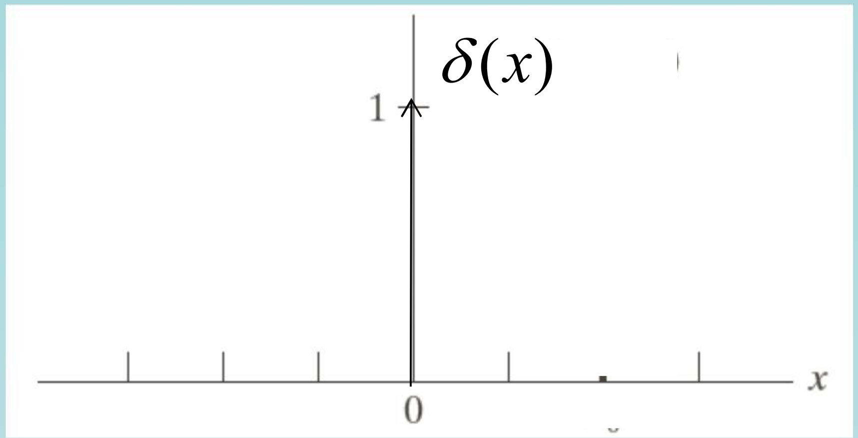
Sifting Property of Continuous Impulse Function

$$\int_{-\infty}^{\infty} \delta(t - t_0) f(t) dt = f(t_0)$$

- Evaluates the function at the location of the impulse

Discrete Unit Impulse Function

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

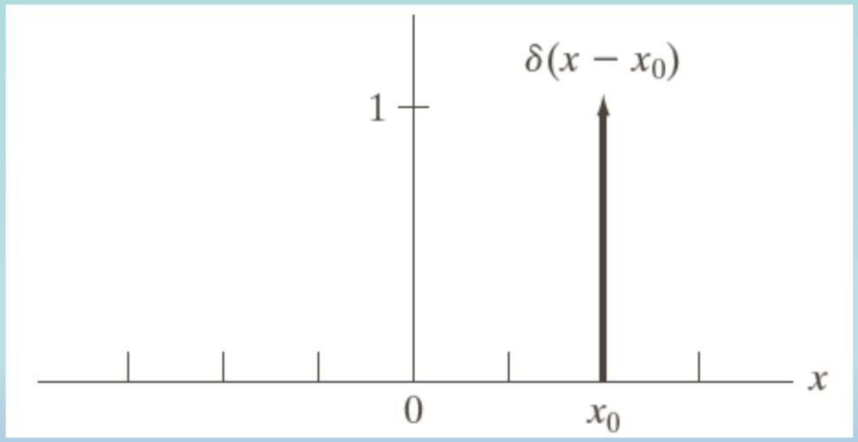


with the constraint,

$$\sum_{x=-\infty}^{x=\infty} \delta(x) = 1$$

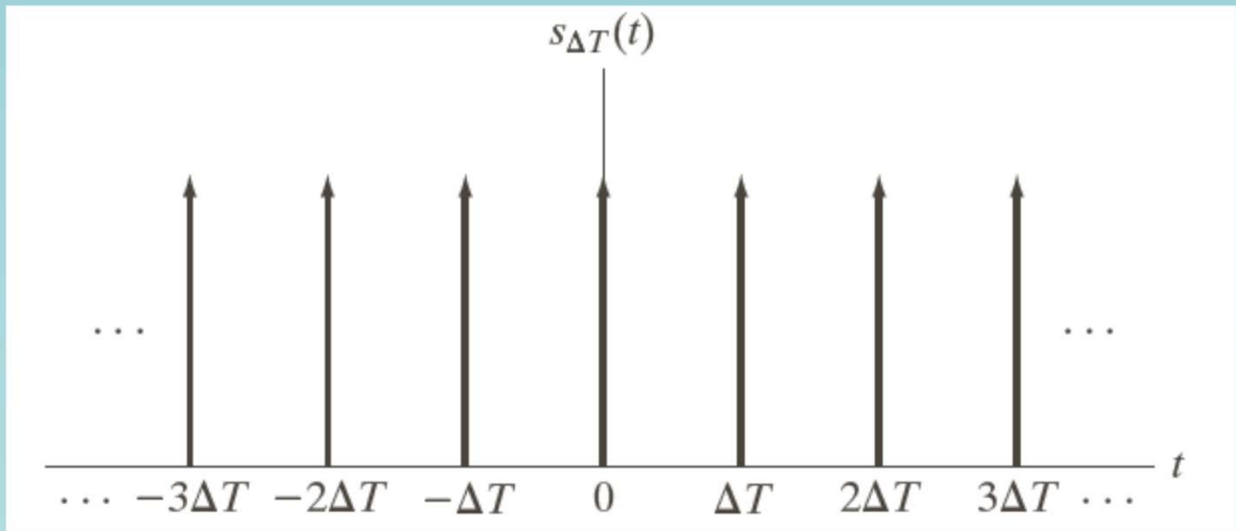
Sifting Property of Discrete Unit Impulse Function

$$\sum_{x=-\infty}^{x=\infty} \delta(x - x_0) f(x) = f(x_0)$$

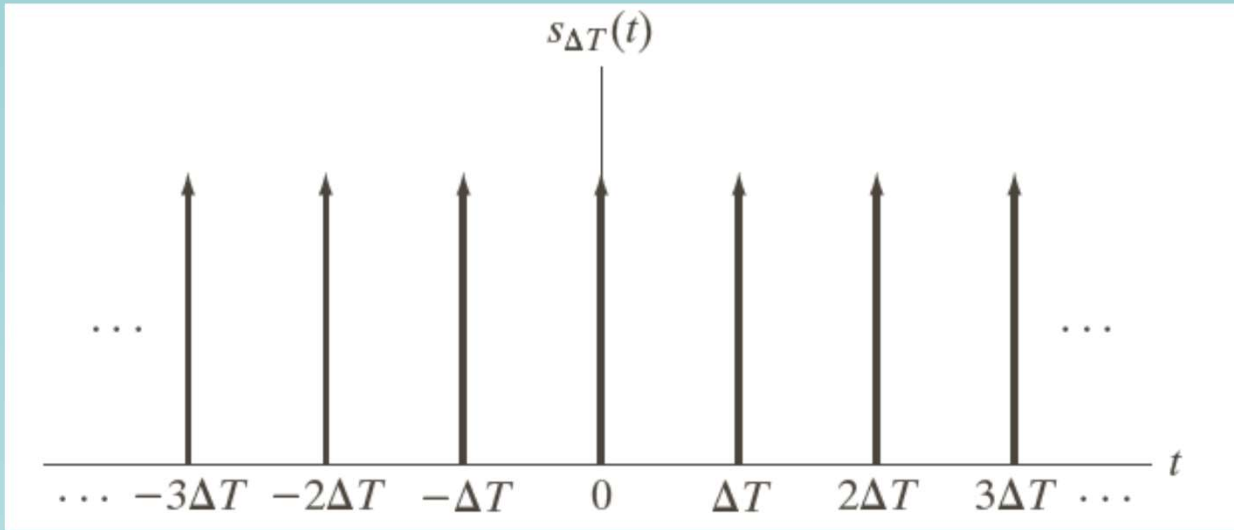


CSE-BUET

Impulse Train: $s_{\Delta T}(t)$



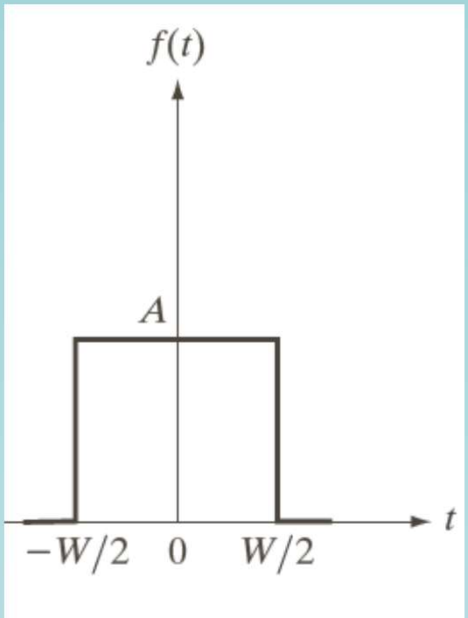
Impulse Train: $s_{\Delta T}(t)$



$$s_{\nabla T}(t) = \sum_{n=-\infty}^{n=\infty} \delta(t - n\nabla T)$$

CSE-BUET

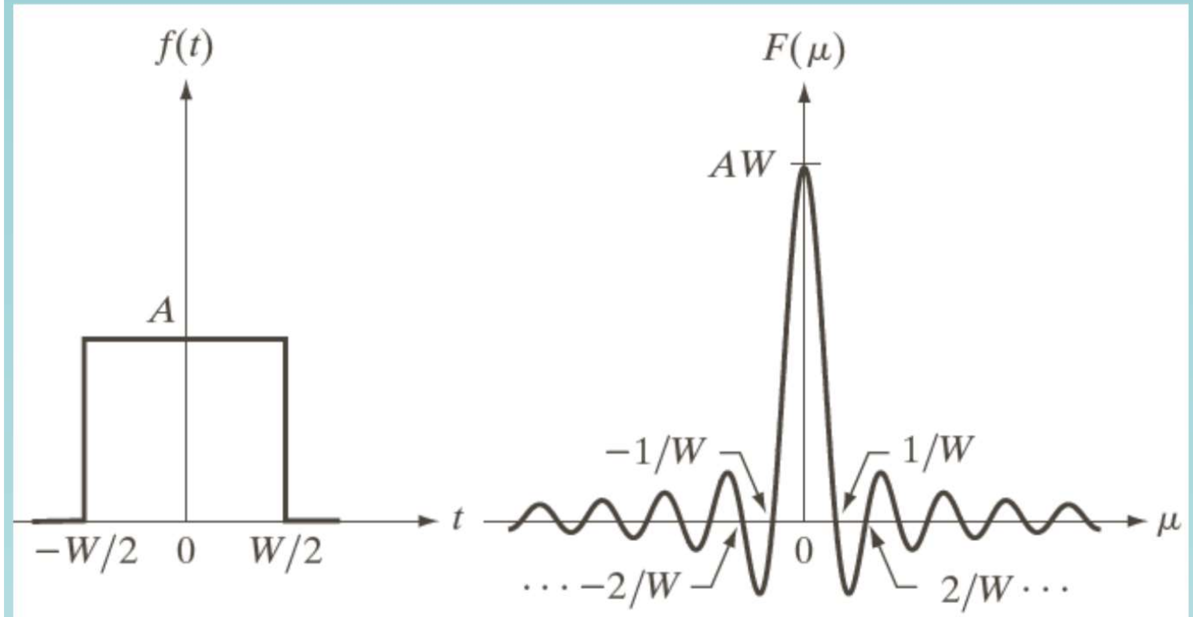
1D Fourier Transform and its Inverse



1D Box/Gate Function

CSE-BUET

1D Fourier Transform and its Inverse

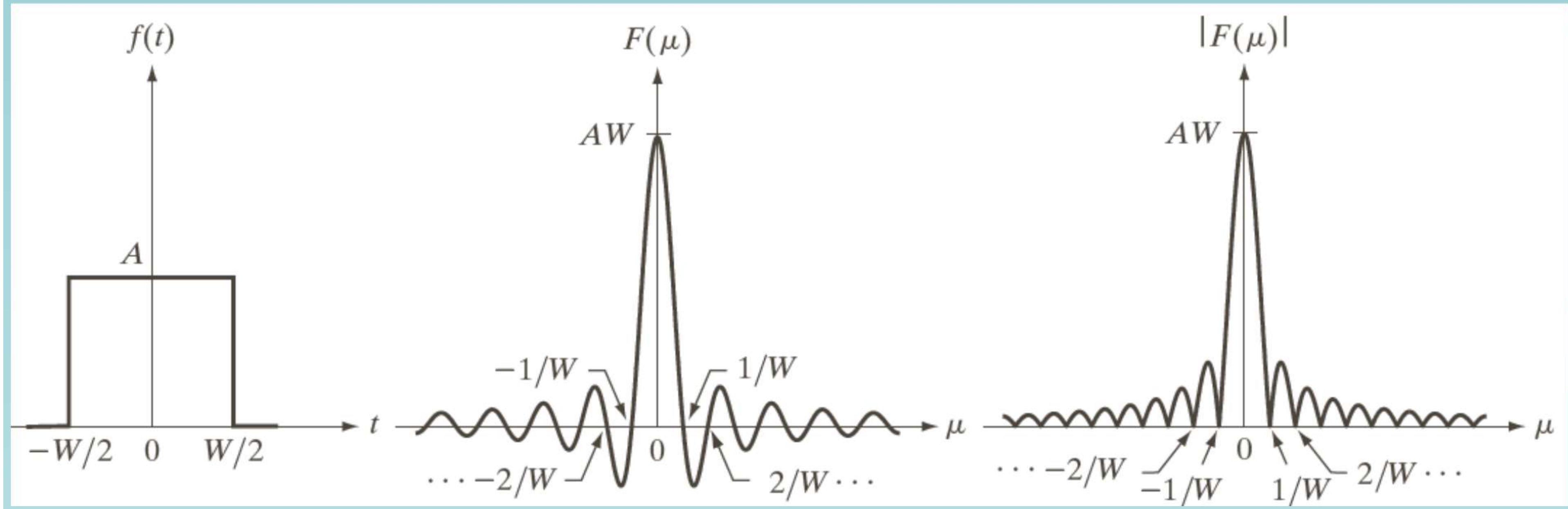


$$F(u) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ut}dt = \int_{-W/2}^{W/2} Ae^{-j2\pi ut}dt = \cdots$$

$$= \frac{A}{j2\pi u}\left[e^{j\pi uW} - e^{-j\pi uW}\right] = AW\frac{\sin(\pi uW)}{\pi uW} = AW\mathrm{sinc}(\pi uW)$$

CSE-BUET

1D Fourier Transform and its Inverse



$$F(u) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi ut} dt = \int_{-W/2}^{W/2} A e^{-j2\pi ut} dt = \cdots$$

$$= \frac{A}{j2\pi u}\left[e^{j\pi uW} - e^{-j\pi uW}\right] = AW \frac{\sin(\pi uW)}{\pi uW} = AW \operatorname{sinc}(\pi uW)$$

CSE-BUET