Fourier Series

1. **What is Fourier Series?**
   - A Fourier series represents the function $f(t)$ as a sum of harmonic (sine and cosine functions) with different frequencies.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right)\right)$$

$$a_0 = \frac{1}{T}\int_{-T/2}^{+T/2} f(t)\ dt$$

$$a_n = \frac{2}{T}\int_{-T/2}^{+T/2} f(t)\cos\left(\frac{2\pi nt}{T}\right) dt$$

$$b_n = \frac{2}{T}\int_{-T/2}^{+T/2} f(t)\sin\left(\frac{2\pi nt}{T}\right) dt$$

   - a_0, a_n, b_n are coefficient that describe how much each harmonic contributes to $f(t)$.

2. **Why Does This Work?**
   - Functions live in a "space", and sine and cosine functions form a complete orthonormal basis for this space. This means any function can be expressed as the combination of these basis functions, just like vectors can be expressed as combination of orthogonal axes.

3. **Linear space of function:**
   - A **vector in Euclidean space** can be represented as a linear combination of orthonormal basis vectors:

$$v = \sum_{i=l} v_i e_i$$

$$where\ v_i\ are\ coefficients\ or\ projections\ of\ v\ onto\ basis\ vecotr$$

   - Extending this idea, a function can be decomposed as a linear combination of basis functions in function space, which is infinite-dimensional.

4. **Piecewise continues functions:**
   - **The function has at most finite number of points in which there are not continues.**
   - A piecewise continues function $f = [a, b] \rightarrow R$ is a function that is continues except for a finite numbers of discontinues.
   - These functions form the vector space V.

5. **Scaler product for functions.**
   - To define the geometry of this function space, a scaler product (inner product) is introduced:

$$< f, g > = \int_b^a f(t)g(t)dt$$

   - This measures the overlap between two functions f, and g.

6. **Norm of a function:**
   - The norm (length) of the function is derived from scaler product of the function:

$$||f|| = \sqrt{< f, f >} = \sqrt{\int_b^a f(t)f(t)dt}$$

   - It quantifies the size and magnitude of the function.

7. **Orthogonal and Orthonormal:**
   - The intuition Behind Fourier Series: suppose f, g are continues, we decide $[a, b]$ in M equal, non-overlapping interval $\frac{b-a}{M}$ with M midpoints $t_1 \dots t_m$:

$$\langle f, g \rangle = \int_a^b f(t)g(t)\ dt \to \lim_{M \to +\infty} \frac{b-a}{M} \sum_{m=1}^{M} f(t_m)g(t_m) \sim \sum_{m=1}^{M} f_m g_m$$

   - Orthogonal and Orthonormal System of Functions: A set of functions $\{\emptyset_n\}$ is orthogonal if for $\forall\ n \neq m\ \in N, < \emptyset_n, \emptyset_m > = 0$. Then the $S_\emptyset = \{\emptyset_0, \dots, \emptyset_n\}$ is orthogonal. If $< \emptyset_n, \emptyset_n > = 1\ \forall\ n\ \in N$, $S_\emptyset$ is orthonormal.

8. **Function decomposition:**
   - Any piecewise continues function can be represented as a linear combination of orthonormal basic functions (Such as sine cosine in Fourier series)

9. **Convergence and Completeness:**
   - The Fourier series converge to $f(t)$ at points where $f(t)$ is continues.
   - At point of discontinuity, the series converges to the averages of left and right limits of the function.
   - Completeness ensures that no information is lost in the representation. If the series doesn't capture all the energy (or norm) of the function, the basis isn't complete.

10. **Theorem 1: Best Approximation:**
    - d_n is an arbitrary real number so f_n is a better approximation for f than d_n. Because the distance between f and f_n is minimized in some way.

$$\forall N > 0 \quad |f - d_N|^2 \geq |f - f_N|^2$$

    - The results hold for any Euclidian space: no other linear combination leads to a better approximation of a vector f in a certain subspace than the one obtained by projecting f along the corresponding basis vectors of that subspace.

11. **Parseval's Equality:**
    - The total energy (or squared norm) of the function equals to the sum of squares of its Fourier coefficients:

$$\|f\|^2 = \sum_{n=1}^{\infty} f_n{}^2$$

    - This shows how the energy of the function is distributed across its harmonics.

12. **Gibs Phenomenon:**
    - When a function has discontinuity, the Fourier series overshoots near the discontinuity, causing oscillations. This overshoot doesn't disappear as more terms are added, but the oscillation's width gets smaller.

- What is linear idependance in an infinite set?
  An infinite linearly independent system, for every finite set that we choose it also should be linearly independent. If the linear combination of that finite set is zero, its because the coefficients are zero. No vector is the linear combination of other vectors.
- What is the best approximation for a function?
  I can't find any other coefficient that makes the distance of function and its approximation less. This best coefficient is the projection of the function along the basis.
- What's Parseval equality in the context of fourier series?
  The Parseval equality determines our approximation is complete or not. If the square norm of the function f(t) is equal to square sum of the coefficients then the approximation is complete and holds true otherwise, if the linear combination of the coefficients is smaller, it means that the approximation of our function isn't complete and we've lost information (energy).
- What's convergence?
- What's Gibbs phonomenan?

Fourier Transforms

1. Introduction:

- The Fourier Transform is a mathematical tool used to analyze and represent signals in terms of their frequency components. It generalizes Fourier Series to functions defined over the entire real line.
- Unlike the Fourier Series which uses discrete coefficients, the Fourier Transform provides a continues frequency-domain representation of signals.
- The Fourier Transform of a time-domain function $f(t)$ is defined as:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

- The Inverse Fourier Transform which reconstructs $f(t)$ from $F(\omega)$, is:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

- Here, $\omega = 2\pi\nu$ is the angular frequency, where ν is the true frequency.

2. Properties of Fourier Transform:

1) Linearity:
   - The Fourier Transform of a linear combination of functions is the same combination of their Fourier Transforms.

$$af_1(t) + bf_2(t) \leftrightarrow aF_1(\omega) + bF_2(\omega)$$

2) Conjugation:
   - If $f(t)$ has a Fourier Transform $F(\omega)$, then the complex conjugate of $f(t)$ corresponds to the conjugate of $F(\omega)$ with a frequency reflection:
   $$f^*(t) \leftrightarrow F^*(-\omega)$$
3) Duality:
   - The Fourier Transform exhibits a symmetry between time and frequency:
   $$f(t) \leftrightarrow F(\omega) \quad \Rightarrow \quad F(t) \leftrightarrow 2\pi f(-\omega)$$
4) Time Shift:
   - Shifting a function in time introduces a phase shift in the frequency domain, with the magnitude unchanged.
   $$f(t - t_0) \leftrightarrow e^{-i\omega t_0} F(\omega)$$
5) Frequency Shift:
   - Multiplying a complex exponential in time shifts the Fourier Transform in frequency.
   $$e^{i\omega_0 t} f(t) \leftrightarrow F(\omega - \omega_0)$$
6) Scaling
   - Scaling in time compresses or extends the frequency domain representation inversely.
   $$f(at) \leftrightarrow \frac{1}{|a|} F\left(\frac{\omega}{a}\right)$$
7) Convolution
   - The convolution of two functions in time domain corresponds to multiplication in frequency domain:
   $$(f * g)(t) \leftrightarrow F(\omega)G(\omega)$$
   - Here:
   $$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$
8) Differentiation:
   - The Fourier Transform of the derivative of $f(t)$ is:
   $$\frac{df(t)}{dt} \leftrightarrow i\omega F(\omega)$$
   - Higher derivatives follow:
   $$\frac{d^n f(t)}{dt^n} \leftrightarrow (i\omega)^n F(\omega)$$
9) Integral:
   - The Fourier Transfer of the integral $f(t)$ is :
   $$\int_{-\infty}^{t} f(\tau) d\tau \leftrightarrow \frac{F(\omega)}{i\omega}, \quad \text{for } F(0) = 0$$
10) Parseval's Theorem:
   - The energy of a function is preserved in both time and frequency domain.
   $$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega$$

3. Special Function and Fourier Transform Pairs

1) Delta function $\left(\delta(t)\right)$:
   - The Dirac delta function is the integral 1 over infinity interval:

$$\delta(t) \leftrightarrow 1, \quad 1 \leftrightarrow 2\pi\delta(\omega)$$

2) Rectangular and Since Functions:
   - A rectangular function in time corresponds to since function in frequency:

$$p_T(t) \leftrightarrow \frac{2\sin(\omega T)}{\omega}$$

   - Conversely:

$$\frac{\sin(\omega_b t)}{\pi t} \leftrightarrow p_{\omega_b}(\omega)$$

3) Gaussian Function:
   - The Fourier Transform of a Gaussian is another Gaussian:

$$f(t) = e^{-\frac{t^2}{2\sigma^2}} \leftrightarrow F(\omega) = \sigma\sqrt{2\pi}e^{-\frac{\omega^2\sigma^2}{2}}$$

   - A Gaussian in time has an inverse variance relationship in frequency $\sigma^2 \leftrightarrow \frac{1}{\sigma^2}$
   - Eigenfunction property: The Gaussian is an eigenfunction of the Fourier Transform operator

4) Triangular Function:
   - The triangular function arises from integration of rectangular function:

$$q_T(t) \leftrightarrow \frac{4\sin^2\left(\frac{\omega T}{2}\right)}{T\omega^2}$$

4. Physical Interpretation

- Module and Phases:
  - The Fourier Transform is represented as:

$$F(\omega) = A(\omega)e^{i\Phi(\omega)}$$

  - where:

$$A(\omega) = \sqrt{R(\omega)^2 + X(\omega)^2}, \quad \Phi(\omega) = \tan^{-1}\left(\frac{X(\omega)}{R(\omega)}\right)$$

  - $A(\omega)$ is the module that captures the magnitude of frequency components it tells you how much of the signal is captured at angular frequency ω. The larger $A(\omega)$ is, the stronger the presence of that frequency in the signal.
  - $\Phi(\omega)$ is phase which describes the alignment. It tells how different frequencies in time are aligned. The phase shows how the frequencies are lined-up for reconstructing the signal.
  - To have purely real Fourier transform, the function must be even.
  - To have a purely imaginary Fourier Transform the function must be odd.
- Time Frequency Duality:
  - A function with finite support in time has infinite support in frequency and vice versa.
  - Uncertainty Principle: The more localized a signal is in one domain, the more spread out it becomes in the other.
  - A finite support means that a function is none zero only over a limited range.

Sampling

1. Introduction

- Sampling is the process of converting continues-time signal $s(t)$ into a discrete-time signal by talking measurements at specific intervals.
- This process enables the analysis and manipulation of signals in digital form.

2. Infinite Impulse Train

- The infinite Impulse Train is the mathematical representation of sampling process:

$$i(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)$$

  - T_s: Sampling Period
  - δ(t): Dirac delta function
- This impulse train serves as a model for periodic sampling, where impulses occur at regular interval T_s

3. Fourier Series Representation of the Impulse Train

- The impulse train can be represented in terms of its Fourier Series:

$$i(t) = \sum_{k=-\infty}^{+\infty} c_k e^{ik\omega_s t}$$

  - Where $\omega_s = \frac{2\pi}{T_s}$ is the angular sampling frequency, and c_k are the Fourier Coefficients. For impulse train:

$$c_k = \frac{1}{T_s}$$

  - Thus the impulse train becomes:

$$i(t) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} e^{ik\omega_s t}$$

4. Fourier Transform of Impulse Train

- Fourier Transform of $i(t)$ is given by:

$$I(\omega) = \frac{2\pi}{T_s} \sum_{k=-\infty}^{+\infty} \delta(\omega - k\omega_s)$$

  - This shows that the frequency domain representation of the impulse train is also periodic, consisting of impulses spaced by ω_s.

5. Sampling a Signal

- When a continues-time signal $s(t)$ is sampled, it is multiplied by the impulse train:

$$s_{\text{amp}}(t) = s(t) \cdot i(t) = \sum_{n=-\infty}^{+\infty} s(nT_s)\delta(t - nT_s)$$

- In the frequency domain, this corresponds to convolution:

$$S_{\text{amp}}(\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} S(\omega - k\omega_s)$$

  - $S(\omega)$: Fourier transform of $s(t)$.
  - $S_{\text{amp}}(\omega)$: Fourier transformed of the sampled signal

- This results shows that the sampling creates shifted copies of the original spectrum $S(\omega)$, spaced by ω_s.

6. Aliasing

- Aliasing occurs when the shifted copies of $S(\omega)$ overlap due to insufficient sampling frequency.
- This distortion arises when the sampling frequency ω_s is less than twice the maximum frequency of the signal ω_b (the Nyquist rate). To avoid aliasing:

$$\omega_s > 2\omega_b$$

    - If aliasing occurs, higher frequencies appear as lower frequencies, leading to loss of information.

7. Shannon Sampling Theorem

- The Shannon Sampling Theorem establishes the conditions for perfect reconstruction of a signal from its samples:
    - The signal must be band limited, meaning $S(\omega) = 0 \; for \; |w| > \omega_b$
    - The sampling frequency must satisfy: $\omega_s > 2\omega_b$
- Under these conditions, the original signal $s(t)$ can be reconstructed using the formula:

$$s(t) = \sum_{n=-\infty}^{+\infty} s_{\text{amp}}(nT_s) \frac{\sin\big(\omega_b(t - nT_s)\big)}{\omega_b(t - nT_s)}$$

    - $s_{\text{amp}}(nT_s)$: Sampled values of the signal.
    - The sinc function $\frac{\sin(\omega_b(t-nT_s))}{\omega_b(t-nT_s)}$ interpolates between sampled points.

8. Mathematical Proof of Sampling Theorem

- The proof relies on expressing the signal $s(t)$ using the inverse Fourier Transform:

$$s(t) = \frac{1}{2\pi}\int_{-\omega_b}^{\omega_b} S(\omega)e^{i\omega t}d$$

    - At sampled point $t_n = \frac{n\pi}{\omega_b}$:

$$s_{\text{amp}}(t_n) = \frac{1}{2\pi}\int_{-\omega_b}^{\omega_b} S(\omega)e^{i\omega t_n}d$$

    - The Fourier expansion of $S(\omega)$ shows that the sampled values $s_{\text{amp}}(t_n)$ are related to the Fourier Coefficients c_n, enabling reconstruction.

9. Nyquist Frequency

- The Nyquist Frequency ω_b is half the sampling frequency: $\omega_b = \frac{\omega_s}{2}$
    - It represents the highest frequency that can be accurately captured without aliasing. The time interval between samples is: $T_s = \frac{\pi}{\omega_b}$

Linear Time Invariant Systems

1. Introduction

LTI systems form the backbone of signal processing, offering a structured way to analyze and manipulate signals.

2. Definition of LTI systems

- An LTI system is a system that satisfies two essential properties:

1) Linearity: A system is linear if the response to a weight sum of inputs is the weight sum of the individual responses, Mathematically:

$$\mathcal{L}\{af_i(t) + bg_i(t)\} = af_o(t) + bg_o(t)$$

2) Time invariance: A system is time-invariant if shifting the input signal results in correspondingly shifted output. Mathematically:

$$\mathcal{L}\{f_i(t-s)\} = f_o(t-s)$$

- These properties simplify the analysis of complex systems by enabling superposition and consistent behavior over time.

3. Characterization of LTI systems

- Impulse Response: The response of an LTI system to an impulse input $\delta(t)$ is called impulse response, denoted as $h(t)$. For a shifted impulse $\delta(t-s)$:

$$\mathcal{L}\{\delta(t-s)\} = h(t-s)$$

- Convolution: the output $f_o(t)$ of an LTI system for any input $f_i(t)$ can be expressed as the convolution of the input signal with the impulse response:

$$f_o(t) = \int_{-\infty}^{\infty} f_i(s)h(t-s)\ ds$$

  - The integral shows that the system's output determined entirely by the input and the impulse response.

4. Properties of LTI systems

- Stability: A system is stable if a bounded input produces a bounded output. This is guaranteed if the impulse response satisfies:

$$\int_{-\infty}^{\infty} |h(t)|\ dt < \infty$$

- Causality: A system is causal if the output at any time depends only on the present and past inputs. For a causal system:

$$h(t) = 0 \quad \text{for } t < 0$$

- This ensures that the system's impulse response is zero before the impulse occurs.

- Eigenfunctions and Eigenvalues: For LTI systems, complex exponentials $e^{i\omega t}$ are eigenfunctions. If the input is $f_i(t) = e^{i\omega t}$, the output is:

$$f_o(t) = H(\omega)e^{i\omega t}$$

  - Where the $H(\omega)$ is the eigenvalue and represents the system's response at frequency ω . This forms the basis for frequency-domain analysis.

5. Frequency Domain Analysis

- System Function: The Fourier Transform of impulse response $h(t)$ is called the system function $H(\omega)$:

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}\ dt$$

  - The system function describes how the system scales and shifts the phases of different frequency components.

- Magnitude and phase: $H(\omega)$ can be expressed in polar form:

$$H(\omega) = |H(\omega)|e^{i\phi(\omega)}$$

  - $|H(\omega)|$: determines the attenuation or amplification of the frequency component.
  - $\phi(\omega)$: Describes the phase shift introduced by the system.

- Convolution in Frequency domain: for an input $f_i(t)$ and the out put $f_o(t)$, their Fourier Transforms are related by:

$$F_o(\omega) = F_i(\omega)H(\omega)$$

  - This property simplifies the analysis by converting convolution into multiplication.

6. Example of RC Circuit as an LTI System

- Differential equation: For an RC circuit with input voltage $V_{in}(t)$ and output voltage across the capacitor $V_{out}(t)$:

$$\frac{dV_{\mathrm{out}}(t)}{dt} + \frac{1}{\tau}V_{\mathrm{out}}(t) = \frac{1}{\tau}V_{\mathrm{in}}(t)$$

  - Where $\tau = RC$ is the time constant.

- Impulse Response: The impulse response of the RC circuit is:

$$h(t) = \frac{1}{\tau}e^{-t/\tau}U(t)$$

- System Function: The Fourier Transform of $h(t)$ is:

$$H(\omega) = \frac{1}{1 + i\tau\omega}$$

  - This shows that the RC circuit acts as a low-pass filter, attenuating high-frequency components.

7. Conclusion:

- Unit Step Response: Describes the system's response to a step input:

$$a(t) = 1 - e^{-t/\tau}$$

- Impulse Response: Shows the immediate reaction to an impulse input:

$$h(t) = \frac{1}{\tau} e^{-t/\tau} U(t)$$

- Frequency Domain Behavior: The system function $H(\omega)$ determines the attenuation and shift phase for each frequency.
- Filter Characteristics: LTI system like RC circuit are foundational for designing filters (low-pass, high-pass) in signal processing.
- Linearity and time-invariance: These properties enable superposition and consistent behavior over time, simplifying analysis and design.

Kalman Filter

1. Introduction:
   - The Kalman filter is a powerful tool for estimating hidden state of a dynamic system, combining noisy measurements and prior knowledge to produce optimal estimates.
   - The Kalman filter operates recursively, predicting the state of the system at the next time step and updating it based on new measurements. This process minimizes mean square error of the estimates.
2. Basic concepts:
   - State and Measurement models
     - State Transition Model: the hidden state vector s_t, evolves over time according to

       $$s_{t+1} = \Phi s_t + q_t$$

       - Φ: State transition matrix, which defines how the current state influences the next state.
       - $\boldsymbol{q_t}$: process noise, capturing uncertainties in the model with:

         $$\mathrm{E}[\mathrm{q_t}] = 0, \quad \mathrm{E}[\mathrm{q_t q_t^\top}] = \mathrm{Q}$$

     - Measurement model: the measurement vector $\mathbf{m_t}$ is related to the state vector $\boldsymbol{s_t}$

       $$m_t = H s_t + r_t$$

       - H: Measurement matrix, mapping the state to the measurement space
       - $\boldsymbol{r_t}$: Measurement noise, with:

         $$E[\boldsymbol{r_t}] = 0, \quad E[\boldsymbol{r_t r_t^\top}] = R$$

3. Recursive Kalman Filter:
   - Prediction Step: Before incorporating the new measurement, the Kalman filter predicts:
     - State Prediction:

       $$\widehat{\boldsymbol{s}_{t+1}^{-}} = \Phi \hat{\boldsymbol{s}}_t$$

     - Covariance Prediction:

       $$P_{t+1}^{-} = \Phi P_t \Phi^\top + Q$$

   - Update Step: when a new measurement $\boldsymbol{m_t}$ is acquired, the state and covariance are updated:
     - Kalman Gain:

       $$K_t = P_t^- H^\top (H P_t^- H^\top + R)^{-1}$$

       - The Kalman gain balances the confidence in prediction versus new measurements.
     - State Update:

       $$\widehat{s_t} = \widehat{s_{t^-}} + K_t(m_t - H\widehat{s_{t^-}})$$

- The term $\boldsymbol{m_t} - H\widehat{\boldsymbol{s}_{t^-}}$ is the innovation or measurement residual.

o Covariance Update:

$$P_t = (I - K_t H)P_t^-$$

4. Key Derivations:
   - Error Covariance: the covariance of the estimation error $\boldsymbol{P_t}$ is:
     $$P_t = (I - K_t H)P_t^-(I - K_t H)^\top + K_t R K_t^\top$$
     - Minimizing the trace of $\boldsymbol{P_t}$ leads to the optimal Kalman gain $\boldsymbol{K_t}$.
   - Covariance Projection: For the next time step, the predicted covariance accounts for the process noise.
     $$P_{t+1}^- = \Phi P_t \Phi^\top + Q$$
   - Derivation of Kalman Gain: To find $\boldsymbol{K_t}$ minimize the trace of $\boldsymbol{P_t}$:
     $$\mathrm{Tr}[P_t] = \mathrm{Tr}[P_t^-] - 2\mathrm{Tr}[K_t H P_t^-] + \mathrm{Tr}[K_t(HP_t^- H^\top + R)K_t^\top]$$
     - Taking derivative:
       $$\frac{\partial \mathrm{Tr}[P_t]}{\partial K_t} = -2HP_t^- + 2K_t(HP_t^- H^\top + R)$$
     - Solving give:
       $$K_t = P_t^- H^\top (HP_t^- H^\top + R)^{-1}$$
5. Example: Estimating a Function and Its Derivative
   - State Model: consider a function f_t and its derivative f_t', with the state vector:
     $$s_t = \begin{matrix} f_t \\ f_t' \end{matrix}$$
     - State Transition:
       $$\Phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} q_1^2 & 0 \\ 0 & q_2^2 \end{bmatrix}$$
     - Measurement:
       $$m_t = f_t + r_t, \quad H = [1\ 0], \quad R = \sigma_r^2$$
   - Filter Steps:
     - Prediction:
       - $f_t = f_t\ +\ f_t'$
       - $f_{t+1}' = f_t'$
       - Covariance: $\mathrm{P}_{t+1}^- = \Phi \mathrm{P}_t \Phi^\top + \mathrm{Q}$
     - Update:
       - Kalman Gain: $\mathbf{K}_t = \begin{bmatrix} p_{11,t}^- \\ p_{21,t}^- \end{bmatrix} \frac{1}{p_{11,t}^- + R}$
       - State Update: $\begin{bmatrix} f_t \\ f_t' \end{bmatrix} = \begin{bmatrix} f_t^- \\ f_t'^- \end{bmatrix} + \mathbf{K}_t(m_t - f_t^-)$
6. Practical Insight
   - Recursive Nature: The Kalman filter efficiently handles dynamic systems by updating estimates step-by-step.
   - Noise Management
     - Process noise (Q) accounts for model inaccuracies.
     - Measurement noise (R) represents sensor limitations.
   - Applications
     - Tracking objects
     - Estimating functions and derivatives in noisy environments.

7. The notions of model and measurement model, the role played by noise in two cases.
   - Model: how the state s_t changes over time with noise:
   $$\boldsymbol{s_{t+1}} = \Phi \boldsymbol{s_t} + \boldsymbol{q_t}$$
   - The noise represents uncertainties in the model (simplification or the effects of the environment)
   $$E[\mathbf{q_t}] = 0, \quad E[\mathbf{q_t}\mathbf{q_t^\top}] = Q$$
   - Measurement model: describes the measurement m_t relates to the state $\boldsymbol{s_t}$ with measurement noise r_t:
   $$\boldsymbol{m_t} = H\boldsymbol{s_t} + \boldsymbol{r_t}$$
   $$E[\boldsymbol{r_t}] = 0, \quad E[\boldsymbol{r_t}\boldsymbol{r_t^\top}] = R$$
   - The noise accounts for inaccuracies in the measurement such as the sensor error.
   - When calculating Kalman Gain, we must decide which of them has a more accurate result. Do we truest $\boldsymbol{m_t}$ or $\boldsymbol{s_t}$.
8. Building a sample model for tracking objects or estimating a function.
   - In the context of object tracking, our state vector contains four values of
   $$s_t = [x_t, y_t, u_t, v_t]^T$$
     - x_t, y_t: represent coordinates in 2D space.
     - u_t, v_t: represent velocity in each axes.
   - We know that:
   $$x_{t+1} = x_t + u_t \Delta t$$
   $$y_{t+1} = y_t + v_t \Delta t$$
   $$u_t = u_{t+1}$$
   $$v_t = v_{t+1}$$

Wavelets

1. Introduction:
   - Wavelets are mathematical tools used for analyzing signals at multiples resolutions.
   - They are particularly useful in signal processing tasks such as compression, denoising, and multiresolution analysis.
   - The Haar wavelet is a foundational example that illustrates the core concepts of wavelets and filter banks.
2. Signal representation using scaling and wavelet functions:
   - Scaling function $\phi(x)$: it represents low-frequency (smooth or approximation) components of the signal:
   $$\phi(x) = \begin{cases} 1, & if\ 0 \le x < 1 \\ 0, & otherwise \end{cases}$$
     - Scaled and translated versions of $\phi(x)$ are defined as:
     $$\phi_i^j(x) = \phi(2^j x - i), \quad i = 0,1, \dots, 2^j - 1.$$
       - Scaling by 2^i: compresses the function to represent finer resolution.
       - Translation by i: shifts the function to cover different intervals.
   - Wavelet Functions $\psi(x)$: it represents the high-frequency (detail) components of the signal:

$$\begin{cases} 1 & if 0 \leq x < \frac{1}{2}, \\ 1 & if \frac{1}{2} \leq x < 1, \\ 0 & otherwise \end{cases}$$

- Scaled and translated versions of $\psi(x)$ are:

$$\psi_i^j(x) = \psi(2^j x - i), \quad i = 0,1, \ldots, 2^j - 1.$$

3. Multiresolution Analysis:
   - It decomposes a signal into components at different levels of resolution. The relationship between spaces V^j (approximation) and W^j (details) is:

$$V^{j+1} = V^j \oplus W^j.$$

     - V^{j+1} contains finer resolution information.
     - W^j captures the details lost when transitioning from V^{j+1} to V^j.

   - Orthogonality: Scaling functions ϕ_i^j and wavelet functions ψ_i^j are orthogonal:

$$\left\langle \phi_i^j, \psi_k^j \right\rangle = 0, \quad \left\langle \phi_i^j, \phi_k^j \right\rangle = \delta_{ik}, \quad \left\langle \psi_i^j, \psi_k^j \right\rangle = \delta_{ik}.$$

     - This ensures no redundancy in representation.
4. Decomposition Analysis: to decompose a signal into approximations (C^{j-1}) and details (D^{j-1}) at coarser level ($j-1$). Analysis filters A^j and B^j are used:

$$C^{j-1} = A^j C^j, \quad D^{j-1} = B^j C^j.$$

   - For Haar wavelets:

$$A^j = \frac{1}{2}\left(P^j\right)^T, \quad B^j = \frac{1}{2}\left(Q^j\right)^T.$$

     - A^j Extracts scaling coefficients
     - B^j Extracts wavelet coefficients

$$A^3 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, B^3 = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

5. Reconstruction (Synthesis Filters):
   - To reconstruct finer-level coefficients (C^j) from coarser-level approximation and details:

$$C^j = P^j C^{j-1} + Q^j D^{j-1}.$$

     - P^j: Synthesis filter for scaling coefficients
     - Q^j: Synthesis filter for wavelet coefficients
   - Synthesis matrices for Haar wavelets (Example j=3)

$$P^3 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, Q^3 = \begin{bmatrix} -1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & -1 \end{bmatrix}$$

6. Energy Preservation:
   - The energy of the signal is preserved during decomposition and reconstruction:

$$|C^j|^2 = |C^{j-1}|^2 + |D^{j-1}|^2.$$

7. Full Decomposition and Reconstruction
   - Decomposition Steps:
     - Start with C^3.
     - Apply A^3 and B^3 to get C^2 and D^2.
     - Repeat for C^2 to obtain C^1 and D^1.
     - Finally, decompose C^1 into C^0 and D^0.
   - Reconstruction Steps:
     - Combine C^0 and D^0 using P^1 and Q^1 to reconstruct C^1.
     - Repeat the process upward to obtain C^2, then C^3.
8. Key Relationships
   - Analysis and Synthesis Filters:
$$\begin{bmatrix} A^j \\ B^j \end{bmatrix} = [P^j \quad Q^j]^{-1}$$
   - Direct Sum Property:
$$V^{j+1} = V^j \oplus W^j.$$
   - Saling and Wavelet Coefficients:
$$C^{j-1} = A^j C^j, \quad D^{j-1} = B^j C^j.$$

Frames

1. Introduction:
   - Frames generalize concept of orthogonal bases by redundancy in vector representation. They Provide robust, stable, and noise-resilient representation, particularly useful in signal processing and data compression.
   - A frame in R^n is a sequence of vectors $\{f_i\}_{i=1}^{m} (m \geq n)$ that satisfies:
$$A|v|^2 \leq \sum_{i=1}^{m} |\langle f_i, v \rangle|^2 \leq B|v|^2$$
   - Where A and B are called frame bounds, These bounds ensure:
     - Lower Bound (A): Prevents the frame from being degenerated.
     - Upper Bound (B): Ensures stability by limiting redundancy.
     - If $A = B$ the frame is called tight.
     - If $||f_i|| = 1$, the frame is unit-norm.

2. Properties of Frames
   - Redundancy:
     - Frames allow $m > n$, leading to redundancy, which is useful for error resilient and signal reconstruction in the presence of noise.
     - The Frame redundancy ration for a tight is:
$$A = \frac{m}{n}$$
   - Frame Operator ($S = R^n \to R^n$):
$$S(v) = \sum_{i=1}^{m} \langle v, f_i \rangle f_i$$
     - S is symmetric, positive definite, and invertible.
     - The eigenvalues of S, λ_{min} and $\lambda_{min}$are the optimal frame bounds:

$$\lambda_{\min}|v|^2 \leq \sum_{i=1}^{m}|\langle f_i, v\rangle|^2 \leq \lambda_{\max}|v|^2$$

3. Dual Frames
   - A dual frame $\{g_i\}_{i=1}^{m}$ satisfies:

$$v = \sum_{i=1}^{m}\langle v, g_i\rangle f_i = \sum_{i=1}^{m}\langle v, f_i\rangle g_i$$

     - Where $g_i = S^{-1}f_i$
   - Minimal Norm Property: Among all possible frame expansions, the coefficients computed using dual frames have the smallest norm:

$$\sum_{i=1}^{m} a_i^2 = \sum_{i=1}^{m}\langle v, g_i\rangle^2$$

4. Analysis and Synthesis Filters
   - Analysis Filter maps a vector $v \in R^n$ to its projection onto the frame vectors:

$$F(v) = \begin{bmatrix} \langle f_1, v\rangle \\ \vdots \\ \langle f_m, v\rangle \end{bmatrix}$$

     - F is an $m \times n$ matrix with rows f_i^T
   - Synthesis Filter
     - The Synthesis filter reconstruct v from its frame coefficients:

$$\sum_{\{i=1\}}^{m} c_i f_i = F^T c$$

     - F^T is the transpose of the analysis filter matrix
5. Tight Frame
   - A frame is tight if $A = B$, meaning:

$$\sum_{\{i=1\}}^{m}|\langle f_i, v\rangle|^2 = A\,|v|^2$$

   - For tight frames, the frame operator simplifies to:

$$A = AI, S^{-1} = \frac{1}{A}I$$

   - Thus vector reconstruction is straight forward:

$$v = \frac{1}{A}\sum_{\{i=1\}}^{m}\langle v, f_i\rangle f_i$$

6. Parseval's Identity for Frames
   - For tight frames, Parseval's identity generalizes to:

$$|v|^2 = \frac{1}{A}\sum_{i=1}^{m}|\langle f_i, v\rangle|^2$$

     - This relates norm of v to the sum of squared projections onto the frame vectors.

7. Frame Construction
   - Basis Extension: A sequence $\{f_1, \dots, f_m\}$ is a frame, it can be extended to a basis for R^m by adding vectors $\{h_1, \dots, h_m\}$ such that:
   $$\begin{bmatrix} f_1 & \cdots & f_m \\ h_1 & \cdots & h_m \end{bmatrix}$$
   - From Frames to Bases: if $\{f_1, \dots, f_m\}$ is a frame, it can be extended to a basis R^m. Frame design often involves projecting a basic of R^m onto R^n.

Images As 2D Signals

1. **Introduction:**
   - A monochrome image: Grey scale image, a 2D array where each element (pixel) represents a single intensity value. The image undergoes two type of discretization:
     - Spatial Sampling (Pixels): the continues real-world is sampled at discrete points to form a digital image. Each sample is a pixel and the finer it is, the more details are preserved.
     - Intensity Quantization: Grey level value. The intensity of each sample point is assigned to a discrete value from a limited set of grey levels. More levels mean smoother intensity transition.

2. **Digitalization: Contains two steps: measurement and conversion (analog to digital).**
   - In Measurement the physical quantity to be represented is measured by an appropriate device that converts it into an electrical continues signal.
   - In the second step the electrical signal is converted into digital signal.

3. **Pixel Representation: The image to be digitalized is overlaid with a grid (sampling grid).**
   - Each element of the grid contains a region of the image and that whole region is approximated by a unique (avg) value.
   - A coarse sampling grid produces an image with fewer details.
     - Image size: $number\ of\ pixels = row\ \times column$

4. **Resolution: Images have fixed size (pixels) and can be visualized at different sizes on various supports (Paper, monitor, …).**
   - The visualization size is controlled by resolution.
   - The resolution depends on the size of the image and the size of the support.
     - The resolution is related on how dense are the elements on the support.

5. **Gray Scale: the values of each pixel is a single sample (only carries intensity information).**
   - It is the result of measuring the intensity of light at each pixel.
     - Black: lowest intensity
     - White: strongest intensity

6. **Color Spaces:**
   a. RGB: Red, Green, Blue is an additive color space. Meaning colors are created by adding different intensities of R, G, and B. This is how digital screens work. RGB is light-based, so the more color you add, the brighter the result (finally white).

b. CMYK: Cyan, Magenta, Yellow, and Black. CMYK is pigment-based (subtractive) and is used in 4-color printing where the inks are layered on top of each other to make colors. The result gets darker (finally black).

7. **Image Histogram:**
   - the histogram of a digital image with intensity levels in image [0, L-1] is a discrete function:
$$h(r_k) = n_k$$
     - r_k: The K-th intensity value of the range.
     - n_k: The number of pixels in image with intensity r_k.
   - While computing histogram, we quantize the range values [0, L-1] into bins. Pixels with similar values go in the same bin.

8. Histogram Processing:
   - an estimate of intensity probability distribution. The area of the normalized histogram is 1.
$$P(r_k) = \frac{n_k}{NM}, 0 \leq r \leq L-1$$
   - Histogram Processing algorithm: Produces transformations on images through their histograms.
     - Global Transformations: Equalization
     - Point-wise Transformations: Translation, Contrast Strech

9. **Histogram Equalization:**
   - If an image has poor contrast, its histogram will be concentrated in a small range (mostly dark or mostly light).
   - Histogram equalization is a technique used to improve global contrast of an image by redistributing its intensity values.
   - It works by spreading out the most frequent intensity values, making the details in darker or lighter areas more visible.

10. **Point Operators:**
    - Simplest kind of transforms. Each output pixel value only depends on the corresponding input pixel value (brightness and contrast adjustment + color correction).
    - Intensity Transformation: define a new level for each level of pixel.
    - Neighborhood Operators: in these operators, an output pixel is obtained starting from a set of neighboring pixels in the input image. The neighborhood is usually squared $(w \times w)$ and w varies.

11. **Geometric Transformation**:
    - Modify the positions of pixels not their values (coordinates $P(p_1, p_2)$) given image I, the effect of transformation H is to move $I(p)\ to\ q = (q_1, q_1)$

12. **Transformations:**
    - Translation: $t = (t_1 + t_2) \rightarrow\ q_1 = (t_1 + p_1), q_2 =\ (t_2 + p_2)$
    - Rotation at angel θ (around the origin): $\begin{cases} q_1 = p_1 cos\theta + p_2 sin\theta \\ q_2 = -p_1 sin\theta + p_2 cos\theta \end{cases}$
    - Scaling : $\begin{cases} q_1 = c.p_1 \\ q_2 = d.p_2 \end{cases}$
    - In digital image processing, Geometric transformation consists of two steps:
      - Spatial transformation of coordinates according to T.
      - An Intensity interpolation to assign intensity values to the spatial Transformation pixels on the discrete grid.

2D Fourier Transform For Images

1. Fourier Transform Formulas:

- 2D Fourier Transform :

$$F(\omega_1, \omega_2) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-i(\omega_1 x + \omega_2 y)}\, dx\, dy$$

- Inverse of Fourier Transform:

$$f(x, y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(\omega_1, \omega_2) e^{i(\omega_1 x + \omega_2 y)}\, d\omega_1\, d\omega_2$$

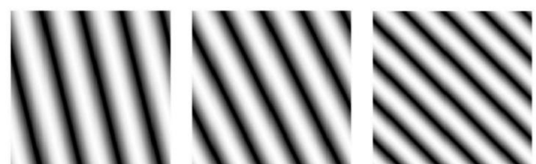
  - ω_1 and ω_2 are the spatial frequencies.
- 2D Fourier Transform is also based on a decomposition into sinusoidal functions that form an orthogonal basis:

$$e^{-i(\omega_1 x + \omega_2 y)} = \cos(\omega_1 x + \omega_2 y) + i \sin(\omega_1 x + \omega_2 y)$$

2. Key points of the plot:

- High Frequencies (ω_1, ω_2) means more oscillations (Densen stripes).
- When $\omega_2 = 0$ the stripes are purely vertical (controlled by ω_1).
- When both $\omega_1, \omega_1 \neq 0$, the patterns become diagonal.
- The angle of pattern is determined by $tran^{-1}(\frac{\omega_1}{\omega_2})$. Meaning that equal values create 45° patterns.

$\omega_1 = 1, \omega_2 = 0$ $\omega_1 = 2, \omega_2 = 0$ $\omega_1 = 4, \omega_2 = 0$

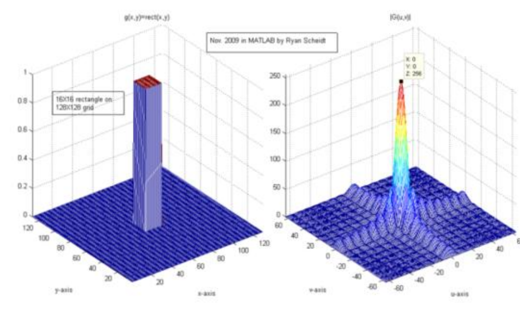


$\omega_1 = 4, \omega_2 = 1$ $\omega_1 = 4, \omega_2 = 2$ $\omega_1 = 4, \omega_2 = 4$

3. Fourier Transform Pairs:

- **Rectangle centered at origin with sides X, Y:**

$$\begin{aligned} F(\omega_1, \omega_2) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-i(\omega_1 x + \omega_2 y)}\, dx\, dy \\ &= \int_{-X/2}^{X/2} e^{-i\omega_1 x}\, dx \int_{-Y/2}^{Y/2} e^{-i\omega_2 y}\, dy \\ &= \frac{1}{2} \mathrm{XY}\, \mathrm{sinc}(\mathrm{X}\omega_1)\, \mathrm{sinc}(\mathrm{Y}\omega_2) \end{aligned}$$

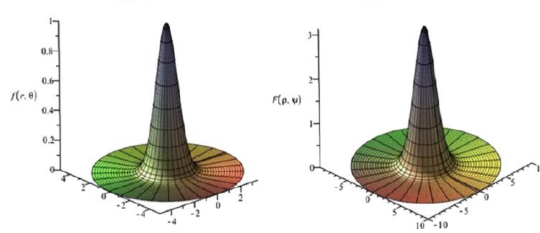


- **Gaussian centered at origin:**

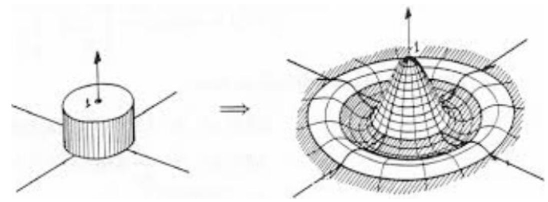
$$f(r) = \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2}, \quad \text{where } r^2 = x^2 + y^2.$$

$$F(u, v) = F(\rho) = e^{-2\pi^2\rho^2\sigma^2}, \quad \text{where } \rho^2 = u^2 + v^2.$$

  - Inverse Scale relation: if a Gaussian is narrow in spatial domain, its FT will be wide in frequency domain (vice versa).
  - Wide Gaussian -> large σ
  - Narrow Gaussian -> small σ



- **Circular disk unit height and radius centered on origin:**



$$f(x,y) = \begin{cases} 1, |r| < a \\ 0, |r| > a \end{cases}$$

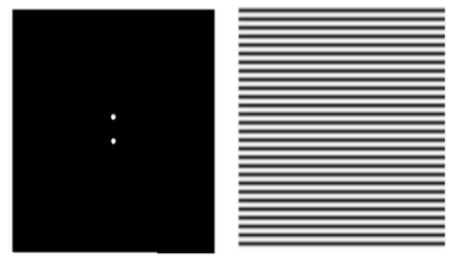
  - Rotational symmetry + a 2D version of Sinc.

- **Delta (Pulse) Function:**

$$f(x,y) = \delta(x,y) = \delta(x)\delta(y)$$

$$F(\omega_1,\omega_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta(x,y)e^{-i(\omega_1 x+\omega_2 y)}\,dx\,dy = 1$$

- **Delta Pairs**



$$f(x,y) = \frac{1}{2}\big(\delta(x,y-a) + \delta(x,y+a)\big)$$

$$F(\omega_1,\omega_2) = \frac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \big(\delta(x,y-a) + \delta(x,y+a)\big)e^{-i(\omega_1 x+\omega_2 y)}\,dx\,dy$$

$$= \frac{1}{2}(e^{-ia\omega_2} + e^{ia\omega_2}) = \cos(a\omega_2)$$

4. Complex Conjugate Symmetry of a Real Function:
   - An image is usually a 2D real-value signal. If $f(x,y)$ is real, then $f(x,y) = f^*(x,y) \rightarrow F^*(w_1,w_2) = F(-w_1,-w_2)$.
     - This means that the spectrum of the Fourier Transform is symmetric. In the other words, there exist negative frequencies which are mirror images of the corresponding position frequencies.
5. Properties:
   - **Rotation in space: we assume g is an in-plane rotation of f by angel θ:**

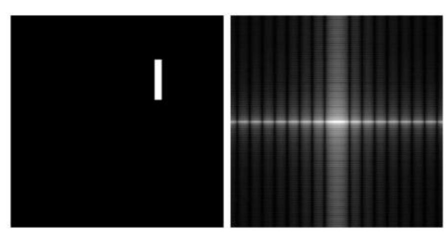
$$(x',y') = R\theta(x,y)$$

$$g(x,y) = f(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$$

     - By linearity properties of FT we have:

$$G(\omega_1,\omega_2) = F(\omega_1\cos\theta - \omega_2\sin\theta, \omega_1\sin\theta + \omega_2\cos\theta)$$

     - The Fourier rotates by same angle of a rotated image
   - **Translation in space: given $g(x-x_0, y-y_0)$ its FT $G(w_1,w_2)$ is related to $F(w_1,w_2)$ by a phase factor**



$$G(\omega_1,\omega_2) = e^{-i(x_0\omega_1 + y_0\omega_2)}F(\omega_1,\omega_2)$$

     - And also for the spectrum:

$$|G(w)| = \left|e^{\{-i\,(x_0\omega_1 + y_0\omega_2)\}}\right||F(\omega_1,\omega_2)| = |F(\omega_1,\omega_2)|$$

     - The spectrum $|F(\omega_1,\omega_2)|$ is insensitive to image translation

- **Shift in frequency: given $G(\boldsymbol{\omega_1} + \boldsymbol{Q_1}, \boldsymbol{\omega_2} + \boldsymbol{Q_2})$ then its IFT is related to $\boldsymbol{f(x, y)}$ by a phase factor:**

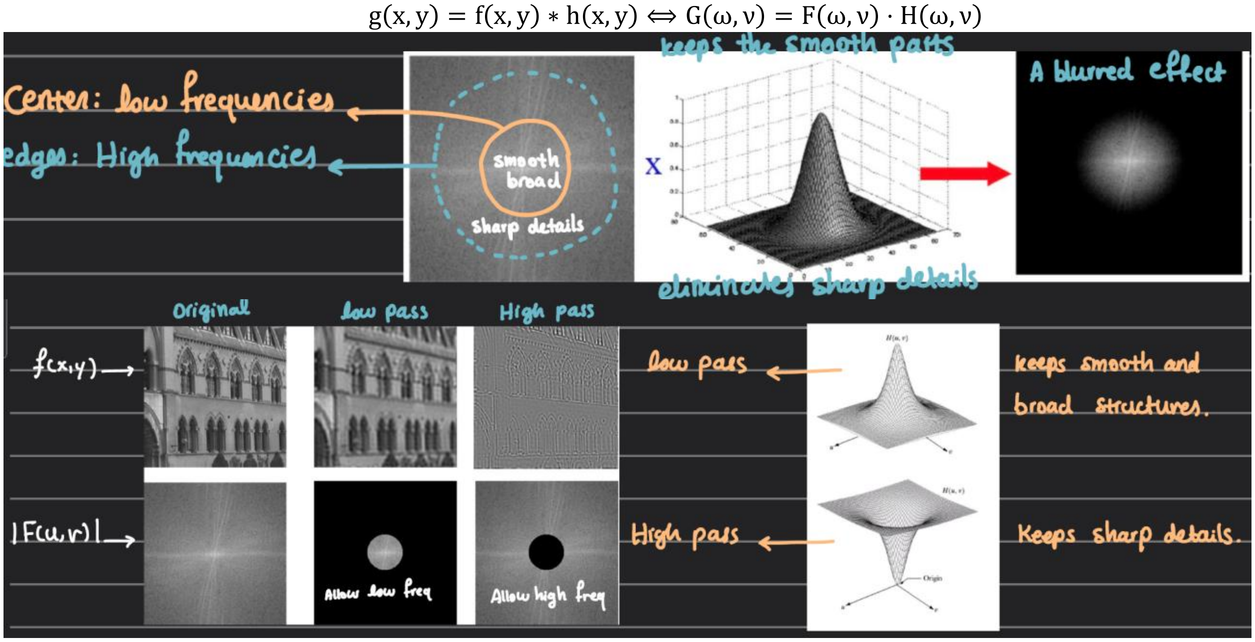
$$g(x, y) = e^{i(\omega_1 x + \omega_2 y)} f(x, y)$$

Fourier Filtering

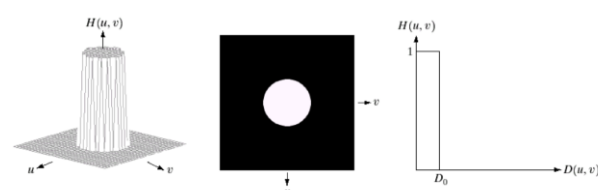
1. **To filter an image in frequency domain:**
   1) Compute DFT of image: $F(u, v)$
   2) Select and appropriate filter H
   3) Multiply $F(u, v)$ by a filter function $H(u, v)$:
   $$G(u, v) \;=\; H(u, v)F(u, v)$$
   4) Compute inverse DFT of $G(u, v)$

2. **Convolution Theorem:**

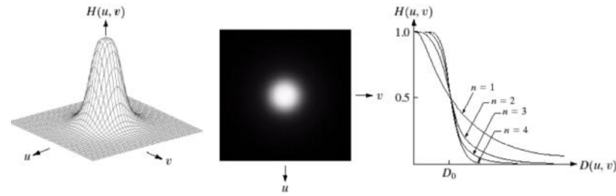
$$\mathrm{g(x, y) = f(x, y) * h(x, y) \Longleftrightarrow G(\omega, \nu) = F(\omega, \nu) \cdot H(\omega, \nu)}$$



- **Image Smoothing: Dropping out high frequency components. By applying low-pass filter. The different low-pass filters are:**
  - Ideal low-pass filter: Simply cut off high frequency components within distance D_0.
    - as D_0 decreases image get more blurry.
    - As D_0 increases image remains similar to the original.



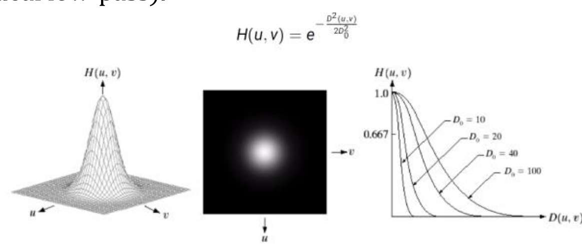
- Butterworth low-pass filter: The transfer function of this filter of order n with cutoff frequency at distance D_0 from the origin is:



$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

  - When n decreases: Gradual smooth transition, wide transition range (Gaussian Like).
  - When n increase: sharper transition (ideal low-pass).

- Gaussian low-pass filter: The transfer function of Gaussian low-pass filter with cut off frequency at distance D_0 from the origin is defined as:



$$\mathrm{H(u,v)} = \mathrm{e}^{-\frac{\mathrm{D^2(u,v)}}{2\mathrm{D_0^2}}}$$

- **Applications:**
  - A low-pass filter is used to connect broken text.
  - Image enhancements: fine details are associated with high frequency components. These filters chop the low ones. High-pass filters are the revers of low pass ones:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

Spatial Features (Convolution and Smoothing)

1. Convolution: The integral of the product of the two functions often one is reversed and shifted.

$$(f * h)(t) = \int_{-\infty}^{+\infty} f(\tau)h(t-\tau)\ d = \int_{-\infty}^{+\infty} f(t-\tau)h(\tau)\ d\tau \quad \text{(by commutativity)}$$

- Discrete convolution: considers two discrete 1D signals $f[n], h[n]$ defined on z:

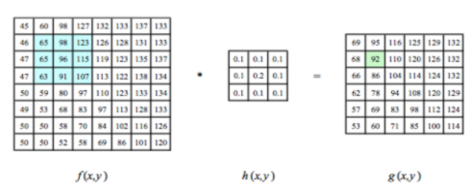
$$(\mathrm{f} * \mathrm{h})[\mathrm{n}] = \sum_{\mathrm{m}=-\infty}^{+\infty} \mathrm{f[m]h[n-m]} = \sum_{\mathrm{m}=-\infty}^{+\infty} \mathrm{f[n-m]h[m]}$$

- Discrete circular convolution: if signals $f[n], h[n]$ have a finite support in Z, $[0, N-1]$ we can consider their periodic extension.

$$(f * h)[n] = \sum_{m=-M/2}^{+M/2} h[m]f[n-m]$$

- In practice we want f *and* h to have different sizes N and M. Because we often assume that the filter h to be a mask, smaller than signal $f(M \ll N)$. And each convolution step acts on a signal neighborhood. So we assume the support of the h to be in $Z[-\frac{m}{2}, \frac{m}{2}]$ and the convolution acts as a central element.

2. 2D discrete Convolution: Consider an image of f and a 2D filter(kernel) h of size $M \times L$. We obtained g the filtered version of f by applying a 2D discrete convolution:



$$g[x,y] = (f * h)[x,y] = \sum_{m=-M/2}^{M/2} \sum_{l=-L/2}^{L/2} h[m,l]f[x-m, y-l]$$

- Each convolution acts on an image neighborhood.

3. Smoothing filter: Applying a smoothing filter corresponds to applying a low-pass filter in Fourier. The main application of low-pass filtering is to reduce noise because the noise is usually is hidden in high-frequencies.
4. Noise Models: Some sort of additive noise
    - Gaussian distribution is a common and effective model.

$$f_r(x, y) = f_i(x, y) + \eta(x, y)$$

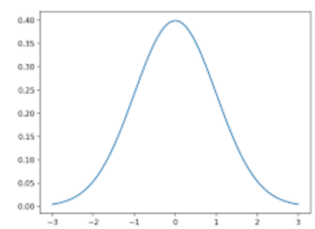
        - f_r: real/observed image
        - f_i: Ideal image (unknown)
        - Noise t

5. Noise Reduction: smoothing filters
    - Average Filter (Simplest choice): derived a by a rectangular function. We replace each pixel by the average of its neighbors and itself.
    - We assume that the neighbors' pixels are similar and the noise is independent from pixel to pixel. Average can be represented by an appropriate kernel.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

    - The larger the size of the kernel, the more severe the smoothing effect.
6. Noise Reduction: Gaussian Filter:
    - The Fourier of a Gaussian is a Gaussian and can be used as a filter in space.
    - Gaussian (zero-mean) distribution in 1D is:

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



    - Gaussian (isotropic case):

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

        - σ: the standard deviation.
7. How to build a small discrete Gaussian kernel mask K_G in space:
    1) Mask finite support: cut out the tails to keep "most" of the area.
    2) Sample W odd points (including x=0) and collect the corresponding values into a Gaussian kernel
    3) Check the sum of values (should be close to 1) and the normalize the values of the kernel to 1

8. Efficiency of separable kernels: if the size of 2D kernel mask is $K \times K$, a single 2D convolution costs $O(K^2)$. But two consecutive 1D filtering operations cost $O(2K)$ and might be more efficient.
    - Separable filters: have the form of $K = vh^T$ and are more efficient:

$$f_R = f * v \rightarrow= f_R * h$$

    - The Gaussian filter and average filter are separable.

Feature detection

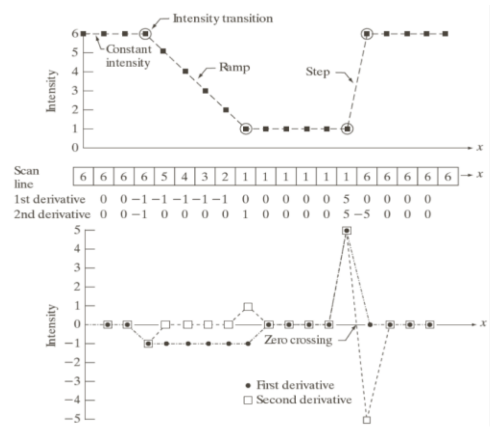
1. Enhancement in system and derivatives
    - They highlight abrupt transition in intensity and correspond to high-pass filters in Fourier. This transitions often define important features in the image like object boundaries.
    - They are used to estimate finite differences (discrete derivatives), in preparation to feature detection in image (edge, corner, texture, …).

- Differentiation amplifies high-frequency components, meaning it enhances not only edges but noise also.

2. Computing Image derivatives
   - 1st derivative: zero in constant areas, not zero on ramps.
   - 2nd derivative: zero in constant areas, zero on ramps with constant slope, not zero on beginning and the end of ramps.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 \mathrm{f}}{\partial \mathrm{x}^2} = \mathrm{f(x+1) + f(x-1) - 2f(x)}$$



- First Derivatives:
  - Image gradient :

$$\nabla f = \mathrm{grad}(f) = [g_x . g_y] = \left[\frac{\partial f}{\partial x} . \frac{\partial f}{\partial y}\right]$$

  - Magnitude of the gradient:

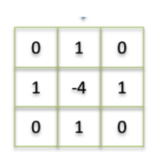
$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$

  - It is not linear so we can't implement it with a kernel.
  - Examples of firs 1st spatial features:

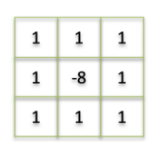
- Second Derivatives:
  - Zero crossings are good indicators of sharp variations.
  - The Laplacian is a 2D isotropic measure of the 2nd spatial derivative of an image.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

  - The Laplacian as a sum of derivatives (which are linear) is linear and can be represented through an appropriate kernel:



Rotation invariant for 90 deg increments



Rotation invariant for 45 deg increments

- Derivatives allow us to highlight signals discontinuities (feature detection)

    - Gradient: edge detection, corners, image sharpening.
    - Laplacian: edge detection (zero crossing algorithm), blob-like features, image sharpening.
3. Derivatives and Noise:
    - In the presence of noise enhancement filter also enhances noise (how to resolve?):
        - Smoothing the image with a low-pass filter.
        - Computing the derivative.

    - From the properties of convolution:

$$\frac{\partial^n}{\partial u^n}(k * f) = \frac{\partial^n k}{\partial u^n} * f$$

        - Computing the derivative of the filter
        - Smooth the image with the derivative of the filter.
    - Why is it better?
        - Since the kernels are smaller than the image, we will have fewer computations.
        - Often the derivative of kernels can be pre-calculated and this would give us only one convolution.
4. The Sobel Operator:
    - The Sobel is an edge detection filter that approximates the gradient (1st derivative) of an image. By detecting rapid intensity change. The Sobel operator can be decomposed in 2 components.
    - A weight averaging filter (smoothing component)
        - A vertical 1D filter that applies a weight average along the vertical direction. It gives more weight to the center pixel (2) and less to the neighbor pixels (1).

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

    - A differentiation filter (edge-detection component):
        - This is a horizontal 1D differentiation filter, which detects changes in intensity along the horizontal axis. It calculates the difference between the left and right neighboring pixels.

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

    - Using separable filters, we build the Sobel kernel:

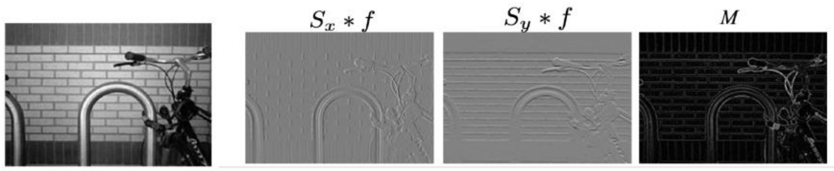
$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

        - X-direction Sobel kernel $\mathrm{g_x} = \mathrm{S_x} * \mathrm{f}$

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Y-direction Sobel kernel $g_y = S_y * f$

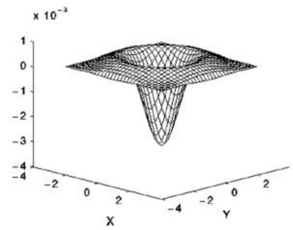
$$S_y = S_x^{\mathrm{T}}$$



- $S_x * f$ enhances vertical lines while $S_y * f$ enhances horizontal lines.

5. LoG (Laplacian of Gaussian) operator:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$



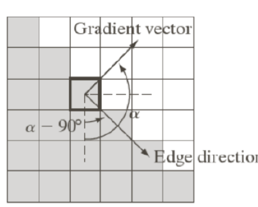
6. Non-linear filter:
   - In contrast to frequency filtering, in the space domain we could design a non-linear filters.
   - A well none example is median filter, used to contrast the salt-pepper noise.
   - Another interesting non-linear filter is bilateral filter (smoothing but edge preserving filter)

Edge Detection

1. **Edges Definition:**
   - Edge Position: Location of edge in image. Typically determined by a change in intensity or gradient magnitude. Position is computed by finding the local maxima in the gradient magnitude of the picture.
   - Edge Normal: The normal points in the direction of the highest intensity change, meaning it is orthogonal to the edge orientation. It is given by gradient vector:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$$

   - Edge Strength: A measure of how prominent the edge is. Usually determined by the gradient magnitude.
     - high gradient M: Strong edges (high contrast).
     - Low gradient M: weak edges (smooth transition).



$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

   - Here we mainly refer to step edges. The ideal step:

$$I(x) = \begin{cases} 0 & x < 0 \\ A & x \geq 0 \end{cases}$$

2. **Criteria for Optimal Edge Detection**
    - Good detection: Minimize the probability of false positive (spurious edges) and false negatives (missed edges).
    - Good localization: The edges detected must be as close as possible to the true edges.
    - Single response constraint: The detector must return one point only for each true edge. Meaning for each actual edge in space, the detector should return one detected edge point. If it produces multiple responses for a single edge, it can lead to thick edges instead of a single clear boundary.

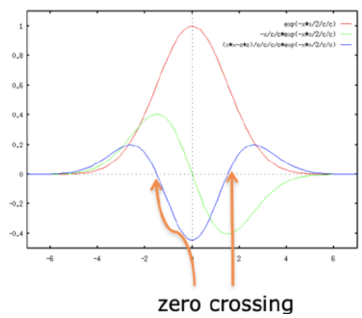
3. **Edge detection by thresholding:**
    - Sobel Algorithm: Sobel as we said is a gradient-based method for edge detection.
    - It highlights regions with high intensity variations by computing the gradient magnitude. It applies a 3×3 convolution kernel (horizontal and vertical edges).
    - To detection strong edges, a threshold T is applied. If the gradient magnitude is above T, it is considered an edge. T can be specified manually or by an algorithm.

4. **Edge detection by zero crossing**
    - Computing the gradient and thresholding is the simplest way. But a more accurate way to compute gradient maxima is to look for zero-crossing of the second derivative:

$$\nabla \cdot J_\sigma(x) = [\nabla^2 G_\sigma](x) * I(x)$$

        - is the Laplacian of the gradient

$$LoG\ mask\ = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$



    - Steps:
        1) Filter the image with LoG mask.
        2) Locate edge elements then the sign between adjacent element changes.
        3) This approach produces edge chains in closed loops (why? The LoG first smoothes the image with Gaussian and then adds Laplacian, since the Gaussian removes the noise, the edges are continuing and well-defined. Also, since images contain enclosed objects, the detected edges form a closed contours around objects or regions of interest.
        4) It is common to use a threshold: Adjacent pixels have different signs and their absolute value exceeds a threshold.

5. Approximating the LoG:
    - The LoG filter can be approximated using the Difference of Gaussian (DoG), which is more effective and computationally simple.
    - The DoG is obtained by subtracting two Gaussian-blurred versions of the same image where each Gaussian has a different standard deviation (σ). This difference closely resembles LoG.

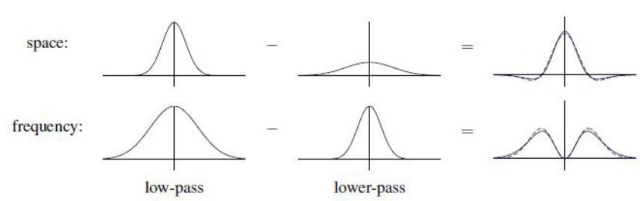


Figure 3.35 The difference of two low-pass filters results in a band-pass filter. The dashed blue lines show the close fit to a half-octave Laplacian of Gaussian.

6. Edge detection: Canny Algorithm
    - It implements an approximation of the optimal step edge detector

- **Edge Enhancement:** Reduce the noise using Gaussian smoothing, then compute Gradient magnitude and directions.
- **Non-Maxima Suppression:** To thin out the edges, the algorithm checks if the current pixel is the local maximum in the direction of gradient. If the pixel's gradient magnitude is not the maximum compared to neighbors along the gradient direction, it is suppressed. (set to zero).
- **Hysteresis Thresholding:** using thresholding, weak and strong edges are identified. Then the weak edges that are connected to strong edges are retained while isolated weak edges are discarded. This removes false edges and ensures continues edges.
- **Computing edges intensity:** $E_s(i,j) = \sqrt{J_x^2(i,j) + J_y^2(i,j)}$
- **Estimating edge normal:** $E_o(i,j) = \arctan\frac{J_y}{J_x}$
- Canny Algorithm Steps:
    1) For each pixel (i,j) given a sampling of direction $(0^\circ, 45^\circ, 90^\circ, \ldots)$, look for direction d in D that best approximates $E_o(i,j)$, if $E_s(i,j)$ is smaller than its neighbors in the direction d, then set the $E_s(i,j)$ to zero.
    2) Given two thresholds $\tau_1 < \tau_2$, for each pixel (i,j) if $E_s(i,j) > \tau_1$:
        I. Starting from $E_s(i,j)$ follow the chains of connected local maxima in both directions perpendicular to the edge normal as long as $E_s(i,j) > \tau_2$.
        II. Mark all visited points and save a list of the locations of all points in the connected contour
- As σ increases, there are fewer and fewer closed chains
- Good features:
    - Patches with large contrast changes are easier to localize.
    - Straight line segments with a single orientation suffer from aperture problem