Assignment 2, Part 1:

Basics of Image Analysis and Rendering

1.1 Digital Image: Concept and Types

A **digital image** is a numerical representation of a visual object composed of pixels. Each pixel carries information about color and brightness.

Types of digital images:

- **Binary Images**: Pixels are either black or white.
- **Grayscale Images**: Pixels have shades of gray (0–255 intensity levels).
- **Color Images**: Pixels use multiple channels, commonly RGB (Red, Green, Blue).
- **Indexed Images**: Pixels store color indices referring to a palette.
- **Multispectral/Hyperspectral Images**: Capture data across various electromagnetic wavelengths.

1.2 Color Spaces

A **color space** organizes colors in a standardized way to facilitate reproduction, editing, and display.

Examples:

- **RGB** (Red, Green, Blue) — common in displays.
- **CMYK** (Cyan, Magenta, Yellow, Black) — used in printing.
- **HSV** (Hue, Saturation, Value) — preferred for intuitive color adjustments.
- **YCbCr** — used for video compression and broadcasting.

1.3 Gray-Scale Codification

Gray-scale codification simplifies an image into shades of gray, where each pixel holds a single intensity value. Typically, 8-bit encoding is used, enabling 256 levels of brightness (0 = black, 255 = white).

1.4 Workflow of Image Analysis

The general **workflow** consists of:

1. **Image Acquisition**: Capture the image using devices like cameras or scanners.
2. **Preprocessing**: Improve image quality (e.g., denoising, normalization).
3. **Segmentation**: Separate the image into meaningful parts or regions.
4. **Feature Extraction**: Identify essential features (shapes, edges, textures).
5. **Analysis and Interpretation**: Apply algorithms to extract information and recognize patterns.

1.5 Levels of Image Processing/Analysis

- **Low-Level Processing**: Direct manipulation of pixel values (e.g., noise reduction, edge detection).
- **Mid-Level Processing**: Structure the image into regions or objects (e.g., segmentation, feature extraction).
- **High-Level Processing**: Interpret structured data to derive meaning (e.g., object recognition, behavior analysis).

1.6 Rendering in Computer Graphics

Rendering is the process of generating a 2D image from a 3D model by simulating lighting and material properties.

Significance: It is essential for creating realistic visuals in fields like gaming, simulations, film, and design.

Rendering Pipeline Stages:

1. **Application Stage**: Sets up the scene and handles animations and visibility.
2. **Geometry Stage**: Transforms 3D models into 2D view through projection.
3. **Rasterization Stage**: Converts vector data (shapes) into pixel fragments.
4. **Fragment Stage**: Applies textures, lighting, and shading.
5. **Output Merger**: Assembles fragments into the final image for display.

1.7 Real-Time Rendering vs. Offline Rendering

Aspect	Real-Time Rendering	Offline Rendering
Goal	Speed (30–60 fps)	Quality (minutes per frame)
Usage	Games, VR, simulations	Films, animations, design
Example Engines	Unreal Engine, Unity	Pixar's RenderMan, V-Ray

---

1.8 Rasterization

Rasterization is the technique of converting 3D objects into a flat, 2D pixel grid.

Role in Rendering:

- Determines which pixels are inside geometric shapes.
- Interpolates attributes (like depth, texture, lighting).
- Produces the final image efficiently for real-time applications.