**Seamlessly Integrating a Person into a Scene**

**Task 1 - Capturing and Preparing the Person's Image**

**Techniques Used:**

1. Image Acquisition via Camera or File

* Image can be captured using a DSLR, webcam, or smartphone.
* In software workflows, image acquisition is done via camera interfaces or file loading.

1. Pre-processing for Quality

* Ensure proper exposure and lighting.
* Maintain a neutral or plain background for easier extraction.
* Capture image in high resolution to retain details during cropping/resizing.

1. (Optional) Face/Body Detection for Alignment

* Detect face or body position to ensure frontal alignment.
* This helps in realistic placement and shadow estimation later.

**Libraries/Tools Used:**

|  |  |
| --- | --- |
| * OpenCV (cv2) : Capture image from webcam, read/save images, visualize |  |
| * Pillow (PIL): Load and process image formats like PNG, JPEG |
| * MediaPipe (optional): Detect and align facial/body landmarks |
| * NumPy: Handle pixel arrays if needed for manipulation |
| * Rembg : Background removal model (U^2-Net-based)   **Task 2 - Analyzing Shadows and Lighting of the Background Image**  **Techniques Used:**   1. **Shadow Detection**  * Color Space Conversion: Convert image to grayscale or HSV/Lab to isolate intensity or illumination. * Thresholding: Apply intensity thresholds to identify darker regions (potential shadows). * Edge Detection: Use Canny edge detection or gradient methods to locate high-contrast shadow edges. * Morphological Operations: Refine detected shadow areas (e.g., erosion, dilation, closing). * Blob/Contour Analysis: Identify continuous shadow regions from thresholded masks.  1. **Shadow Classification**  * Hard Shadows: Identified using high gradient magnitude or edge detection (sharp boundaries). * Soft Shadows: Detected using low-frequency gradients or by subtracting blurred image from original. * Binary Mask Generation: Each type of shadow is stored in a separate binary mask for future use (light estimation, shadow casting).   **Libraries Used:**  OpenCV (cv2): Image loading, grayscale conversion, thresholding, blurring, edge detection, morphological ops.  NumPy: Pixel-level operations, masks, arrays.  Matplotlib (optional): Visualize intermediate results (shadows, gradients, etc.)  **Task 3 - Determining Light Direction**  **Techniques Used:**   1. **Shadow-Based Light Direction Estimation (Primarily for Outdoor Scenes)**  * Shadow Detection: Detect shadows using thresholding and edge detection. * Keypoint Identification: Identify key points like feet and shadow tips using human pose estimation or manual annotation. * Vector Calculation: Compute vector from person's feet to shadow tip. This gives the 2D projection of light direction. * 3D Light Direction Approximation: Use geometry and assumed camera parameters (like height and FOV) to estimate light vector in 3D.  1. **Lighting Estimation (For Indoor or Cloudy Outdoor Scenes)**  * Histogram Analysis: Analyze brightness/intensity variation across the image to determine the brighter direction. * Gradient Direction Estimation: Use image gradient methods (e.g., Sobel operator) to determine dominant light flow. * Surface Orientation (Advanced): Use deep learning-based tools to estimate surface normals and lighting direction. * Vanishing Point/Edge Analysis (Optional): For structured indoor scenes, detect vanishing points to infer light sources (e.g., ceiling lights).   **Libraries Used:**   * OpenCV: Image loading, grayscale conversion, edge detection, histograms, gradients. * NumPy / math: Vector calculations, trigonometry. * MediaPipe / OpenPose (optional): Detect body keypoints like head, foot. * scikit-image: For gradient/texture-based light estimation. * PyTorch / TensorFlow (optional): Advanced models for surface/illumination estimation.   **Task 4 - Coloring and Blending**  **Techniques Used:**  1. Color Matching (Color Transfer/Histogram Matching):  Adjust the colors of the foreground (person) to match the style and tone of the background.  **Methods:**   * **Histogram Matching**: Match the color histogram of the foreground to the background. * **Color Transfer in LAB Color Space**:   + Convert both images to LAB.   + Align the mean and standard deviation of the L, A, B channels.   + Reconstruct the adjusted person image.   2. Relighting / Shading Adjustment:  Modify brightness and shading of the person based on estimated light direction (from Task 3).  **Methods:**   * Use the **light vector** to simulate lighting on the person. * Apply **3D shading models** like Phong or Lambertian (optional). * Approximate using **shadow masks and gradient blending**.   **3. Seamless Cloning / Alpha Blending**  Ensure the edges between the person and the background are smooth.  **Methods:**   * **Poisson Image Editing** (Seamless Cloning): Blends color and gradient at boundaries. * **Feathered Alpha Mask**: Create a smooth boundary around the person using a soft edge mask.   **4. Shadow Rendering and Integration**  Create artificial shadows cast by the person and blend them into the scene.  **Methods:**   * Use **shadow direction vector** from Task 3. * Create a binary mask of the person’s shape. * Apply transformation + blur to simulate soft/hard shadows. * Blend using **multiplicative blending**.   **Libraries Used:**   * OpenCV: Image processing (histogram matching, blending, seamlessClone). * NumPy: Pixel-level matrix operations. * scikit-image: Advanced color and histogram utilities. * Pillow: Image enhancement and alpha manipulation. * cvzone (optional): Easy overlays and blending. * PyTorch / TensorFlow (optional) : Advanced relighting models (e.g., Deep Relighting).   **Task 5 - Generating the Final Output**  **Techniques Used**  **1. Compositing and Seamless Integration**   * What it does: Merge the processed person (foreground) into the background scene. * Technique:   + Image pyramids or Poisson blending for seamless transition at edges.   + Alpha matting for accurate foreground masking.   + Mask refinement to ensure hair/edges are clean.   2. **Color Harmonization**   * What it does: Ensures the color of the person matches the background environment. * Technique:   + Histogram matching   + Style transfer (lightweight)   + Neural harmonization methods (e.g., Deep Image Harmonization)   **3. Lighting and Shadow Re-Application**   * What it does: Aligns the lighting on the person with the environment. * Technique:   + Light direction estimation (from Task 3)   + Directional relighting using shading models or pretrained networks   + Synthetic shadow generation using person’s mask and light vector   **4. Final Rendering**   * What it does**:** Produces a high-resolution image that looks real. * Technique:   + Gaussian smoothing   + Deblurring   + Super-resolution if needed   **Step By Step Task Perform On Image**     |  | | --- | |  | | | |
|  | | |

|  |  |
| --- | --- |
|  |  |
|  |
|  |
|  |
|  |