3D Reconstruction from a Single 2D Image using Shape-from-Shading and Plotly

Denzel Lupheng

Objective

This project aims to reconstruct a 3D surface from a single 2D image using classical computer vision techniques. The key goals are:

- To infer surface depth using shading cues from a single grayscale image.
- To reconstruct a 3D mesh using integrable surface estimation.
- To replicate the 2D image's texture on the 3D surface using per-vertex coloring.
- To visualize the 3D model interactively using Plotly, making it suitable for academic presentations and exploratory research.

Methodology

1. Preprocessing

- **Contrast Enhancement (CLAHE):** Enhances local contrast and highlights surface features.
- **Edge-Preserving Smoothing (Bilateral Filter):** Reduces noise while retaining shading cues, which are critical for gradient computation.

2. Gradient Estimation

Scharr Operator: Computes partial derivatives in both x and y directions. It provides sharper and more reliable gradient maps than the Sobel operator.

3. Surface Integration

Frankot-Chellappa Algorithm: Converts the estimated gradients into an integrable surface (depth map). The result is normalized and scaled to improve 3D visibility.

4. Shape Detection and Masking

- **Hough Circle Transform:** Detects dominant circular shapes (useful for objects like clocks).
- **Fallback Strategy (Contours):** Uses Canny edge detection and polygon approximation to identify common shapes such as triangles, rectangles, and ellipses.
- **Default Mask:** If no shape is detected, the entire image is used.

5. Texture Replication

Per-Vertex Color Assignment: Each vertex of the 3D mesh is colored based on the corresponding RGB value from the input image. This replicates the appearance of the original 2D image onto the 3D mesh.

6. 3D Visualization

Plotly Mesh3d Renderer: Interactive, browser-friendly, and WebGL-accelerated. Users can rotate, zoom, and explore the surface in real-time. This also enables deployment in notebooks or web-based apps like Colab.

Results and Observations

- The model successfully infers pseudo-depth using shading gradients and displays shape-relief with good accuracy for relatively planar and structured images.
- Circle-based masking isolates the main object and removes noisy backgrounds, improving 3D shape fidelity.
- The texture mapping accurately preserves the original image's appearance.
- The interactive 3D viewer enhances user engagement and allows intuitive understanding of surface geometry.

Visual Outcome

- The resulting mesh closely replicates the input image in color.
- Depth exaggeration factor (`scale = 15`) makes subtle reliefs more visible.
- Mesh triangulation is done per pixel cell for optimal resolution.

Deployment

Local Setup

- Clone the GitHub repository.
- Install required packages:

```
pip install -r requirements.txt
```

- Run the pipeline:

```
python main.py --image_path path/to/image.jpg
```

Google Colab

- Upload your image in the Colab interface.
- Paste the pipeline code into a Colab cell.
- Run and interact with the Plotly-rendered 3D output inline.

Repository Structure

- `main.py`: The main script to execute the pipeline.
- `requirements.txt`: Contains Python dependencies.
- `shape_from_shading/`: Modular components — preprocessing, gradient estimation, integration, shape detection, texturing, and visualization.
- `examples/`: (Optional) Notebook-based demo.

Academic Significance

- Demonstrates an effective integration of classical CV (shape-from-shading) with modern visualization.
- Serves as a pedagogical tool for understanding how local image gradients contribute to surface shape.
- Can be used in CV coursework, graphics modules, or as a visualization engine in research.
- Extensible to include learning-based depth inference or hybrid methods.

Limitations

- Requires relatively smooth and evenly lit grayscale images for accurate shading cues.
- Works best on images with planar surfaces and minimal occlusion.
- Texture mapping is per-vertex and does not account for geometric distortion or occlusions.

Future Work

- Incorporate learning-based monocular depth estimation as an optional back-end.
- Add camera parameter estimation for real-world metric scaling.
- Provide `.obj`/`.ply` export for AR/VR use cases and 3D printing.
- Support for occlusion handling and reflectance modeling.

Conclusion

This project successfully demonstrates a classical yet practical 3D reconstruction pipeline based on shape-from-shading. The combination of Frankot-Chellappa integration, smart preprocessing, and interactive Plotly visualization enables a visually intuitive and technically sound replication of 2D images into textured 3D geometry. The modular structure supports future extensions and makes it ideal for academic learning, demos, or portfolio projects.