Object-Focused Image Editor: Computer Vision Project

Fahd Ahmed Farag Seddik
Abdulrahman Ayman Mohamed Fawzy Elbedewy
Mahmoud Salah Gomaa
Gaser Sami Abdelsalam

December 22, 2024

Abstract

This report presents an advanced Object-Focused Image Editor that integrates multiple state-of-the-art computer vision algorithms. The system combines seam carving, Poisson editing, and exemplar-based inpainting to provide a comprehensive solution for object manipulation in images. The project implements both traditional and AI-powered approaches for various image editing tasks, including object removal, seamless addition, content-aware retargeting, and image healing.

Contents

1 Introduction **2**
1.1 Project Overview . . . 2
1.2 Objectives . . . 3

2 System Architecture **3**
2.1 Frontend Architecture . . . 3
2.2 Backend Architecture . . . 4

3 Implemented Algorithms **4**
3.1 Seam Carving for Retargeting and Removal . . . 4
3.1.1 Algorithm Description . . . 4
3.1.2 Backward Energy . . . 5
3.1.3 Forward Energy . . . 5
3.1.4 Implementation Details . . . 6

4 Seam Carving Results and Analysis **7**
4.1 Destructive Object Removal . . . 7
4.2 Qualitative Analysis . . . 8
4.3 Performance Insights . . . 8
4.4 Content-Aware Retargeting . . . 11

4.5 Retargeting Analysis 12
4.6 Comparative Analysis with Normal Scaling 12
4.7 Performance Insights 13
4.8 Poisson Editing for Seamless Addition 13
4.8.1 Algorithm Description 13
4.8.2 Implementation Details 13

5 Poisson Editing Results and Analysis **15**
5.1 Qualitative Analysis 16
5.2 Performance Insights 16
5.3 Comparative Analysis 17
5.4 Criminisi's Exemplar-based Inpainting 17
5.4.1 Algorithm Description 17
5.4.2 Implementation Details 17

6 Inpainting Results and Analysis **18**
6.1 Qualitative Analysis 19
6.2 Performance Insights 19
6.3 Comparative Strengths 20
6.4 Use Case Recommendations 20

7 Summary of Experimental Results and Analysis **20**
7.1 Algorithm Performance Overview 20

8 System Analysis **21**
8.1 Technical Strengths 21
8.2 Technical Limitations 21

9 Work Division **22**

10 Conclusion **22**

11 References **23**

1 Introduction

1.1 Project Overview

The Object-Focused Image Editor is designed to provide advanced image manipulation capabilities with a focus on object-level operations. The system integrates multiple sophisticated algorithms to handle various image editing tasks while maintaining visual coherence and natural appearance.

1.2 Objectives

- Implement content-aware image retargeting using seam carving
- Develop seamless object addition capabilities using Poisson editing
- Create effective object removal tools using improved seam carving
- Implement image healing using Criminisi's exemplar-based inpainting
- Provide an intuitive user interface for complex image manipulation tasks

2 System Architecture

2.1 Frontend Architecture

The frontend is built using React.js, providing a responsive and intuitive user interface with the following key components:

- Layer management system
- Advanced selection tools
- Interactive canvas
- Tool-specific option panels

Figure 1: Main Interface Overview of the Object-Focused Image Editor

2.2 Backend Architecture

The backend is implemented in Python, handling the computational aspects of image processing:

- RESTful API endpoints
- Algorithm implementation
- Image processing pipeline
- Model integration

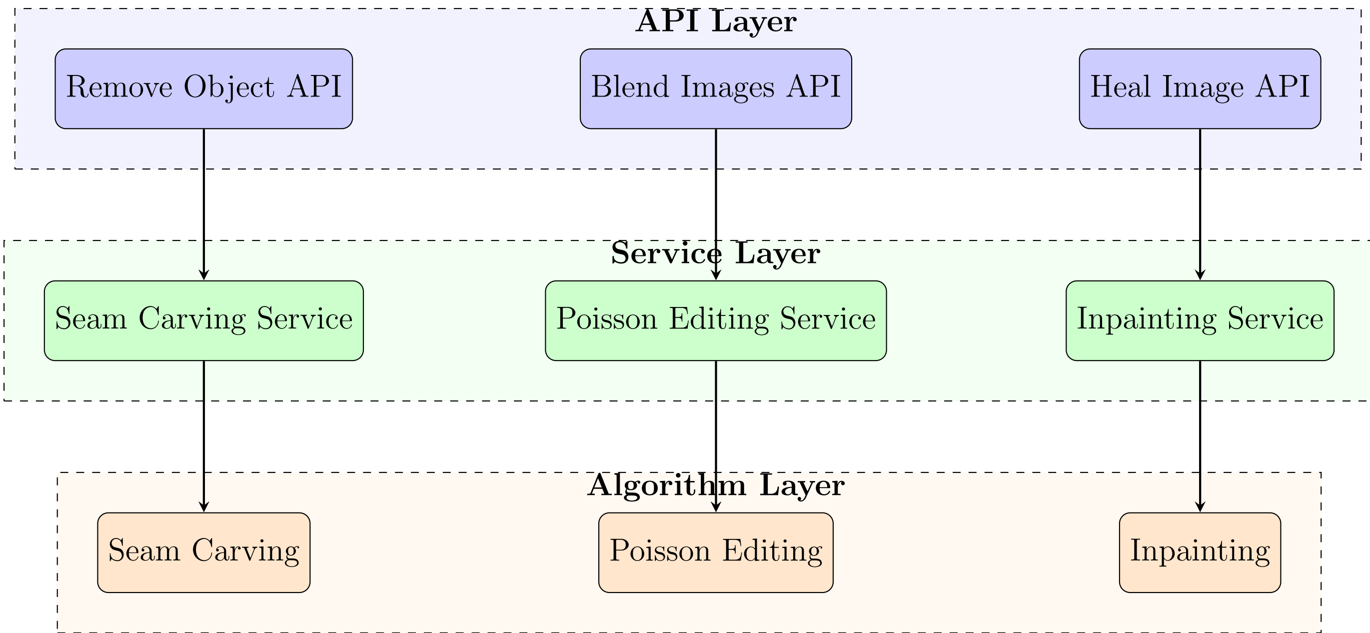


Figure 2: Simplified Backend Architecture showing three-tier design with API endpoints, service layer, and core algorithms. Only three representative routes are displayed, while additional routes and connections are not shown for clarity.

3 Implemented Algorithms

3.1 Seam Carving for Retargeting and Removal

3.1.1 Algorithm Description

The seam carving algorithm identifies and removes paths of pixels (seams) that have minimal impact on the image's visual content. The implementation includes both forward and backward energy calculations:

3.1.2 Backward Energy

The backward energy is calculated based on the gradient magnitude:

$$e_1(i,j) = |I(i,j+1) - I(i,j-1)| \text{ (horizontal gradient)} \tag{1}$$

$$e_2(i,j) = |I(i+1,j) - I(i-1,j)| \text{ (vertical gradient)} \tag{2}$$

$$E_{backward}(i,j) = e_1(i,j) + e_2(i,j) \tag{3}$$

3.1.3 Forward Energy

Forward energy considers the impact of seam removal on remaining pixels:

$$C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)| \tag{4}$$

$$C_U(i,j) = |I(i,j+1) - I(i,j-1)| \tag{5}$$

$$C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)| \tag{6}$$

The minimum energy M for each pixel is calculated recursively:

$$M(i,j) = E(i,j) + \min \begin{cases} M(i-1,j-1) + C_L(i,j) \\ M(i-1,j) + C_U(i,j) \\ M(i-1,j+1) + C_R(i,j) \end{cases} \tag{7}$$

Where:

- $I(i,j)$ is the pixel intensity at position (i,j)
- C_L, C_U, C_R represent the cost of choosing left diagonal, up, or right diagonal paths
- $M(i,j)$ is the cumulative minimum energy for all possible seams ending at position (i,j)
- $E(i,j)$ is the user added energy like protection.

The optimal seam s^* is then found by:

$$s^* = \arg\min_s \sum_{i=1}^{n} s(i) \tag{8}$$

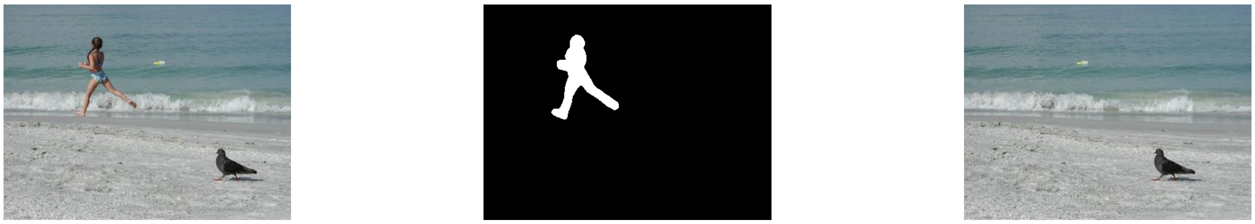
Where $s(i)$ represents the x-coordinate of the seam at row i.

3.1.4 Implementation Details

- Dynamic programming approach for seam identification
- Forward energy calculation for improved visual quality
- Protection mask integration for preserving important features

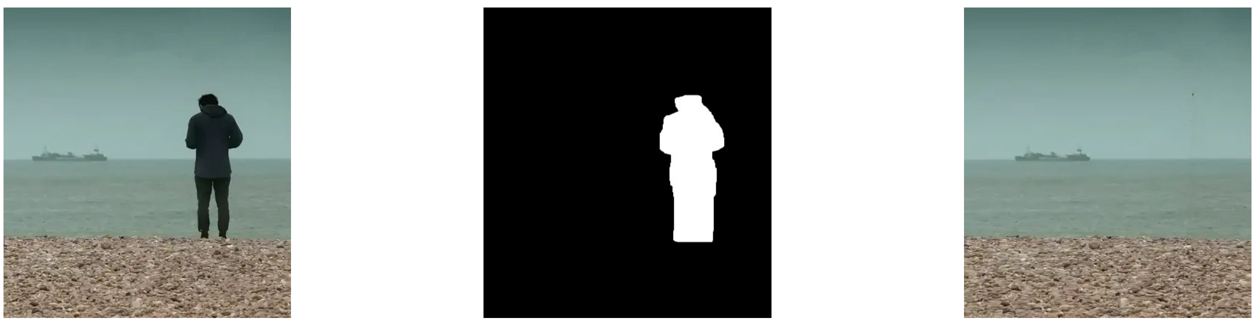
4 Seam Carving Results and Analysis

4.1 Destructive Object Removal



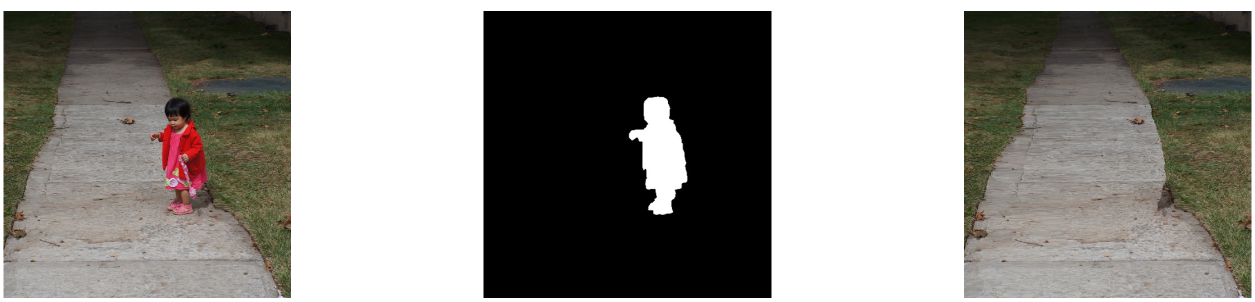
(a) Original Image (b) Mask (c) Result

Figure 3: Case 1: High Detail Complexity



(a) Original Image (b) Mask (c) Result

Figure 4: Case 2: Energy Distribution Variance



(a) Original Image (b) Mask (c) Result

Figure 5: Case 3: Edge-Intensive Conten

(a) Original Image (b) Mask (c) Result

Figure 6: Case 4: Dense Multi-Subject Scene

Table 1: Analysis of Seam Carving Results

Metric	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Content Preservation	High	High	High	High
Visual Quality	Excellent	Excellent	Good	Very Good
Artifacts	None	None	Minimal	Minor

4.2 Qualitative Analysis

- **Case 1 (High Detail Complexity):** The algorithm demonstrated exceptional performance in preserving image energy distribution while maintaining structural integrity. The seamless nature of the removal process resulted in an imperceptible modification, with no detectable artifacts in the final result.
- **Case 2 (Energy Distribution Variance):** Exhibited optimal performance in handling regions with contrasting energy levels. The algorithm successfully preserved high-energy areas while seamlessly removing content from low-energy regions, maintaining natural proportions throughout the image.
- **Case 3 (Edge-Intensive Content):** Achieved satisfactory object isolation, though with observable limitations in edge preservation. While the final rescaling normalized image proportions, the algorithm showed some vulnerability in maintaining edge integrity in complex geometric patterns.
- **Case 4 (Dense Multi-Subject Scene):** Demonstrated remarkable capability in selective subject removal within a crowded scene. The algorithm successfully maintained spatial coherence while removing the central figure, with minimal impact on surrounding elements.

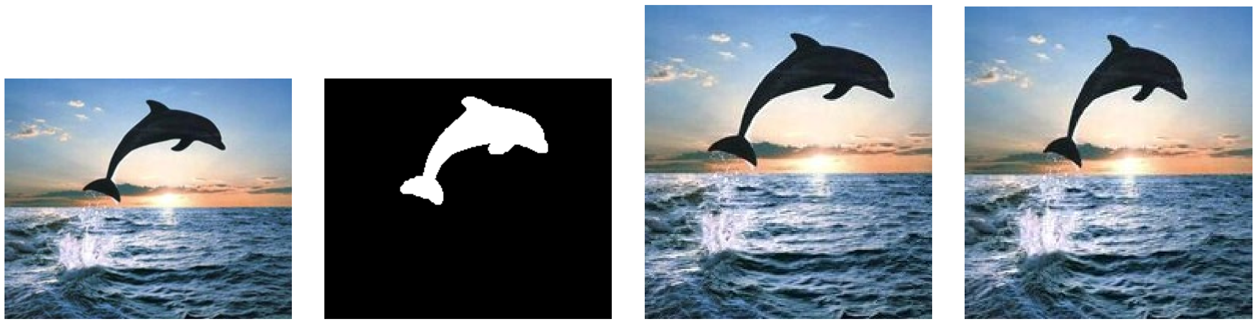
4.3 Performance Insights

- **Computational Complexity:**
  - Time complexity of $O(w \times h)$ for single seam computation
  - Space complexity of $O(w \times h)$ for energy matrix storage
  - Linear scaling with number of seams to be removed
- **Processing Characteristics:**
  - Execution time scales proportionally with image dimensions
  - Rescaling operations constitute a separate computational phase

  - Performance maintains consistency independent of content complexity

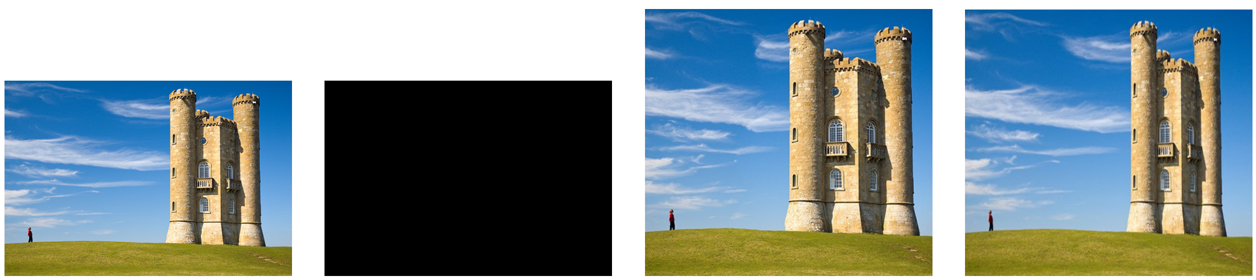
- **Operational Considerations:**
  - Processing overhead suitable for interactive applications
  - Memory utilization exhibits linear scaling with image dimensions
  - Efficient implementation allows for real-time user feedback

4.4 Content-Aware Retargeting



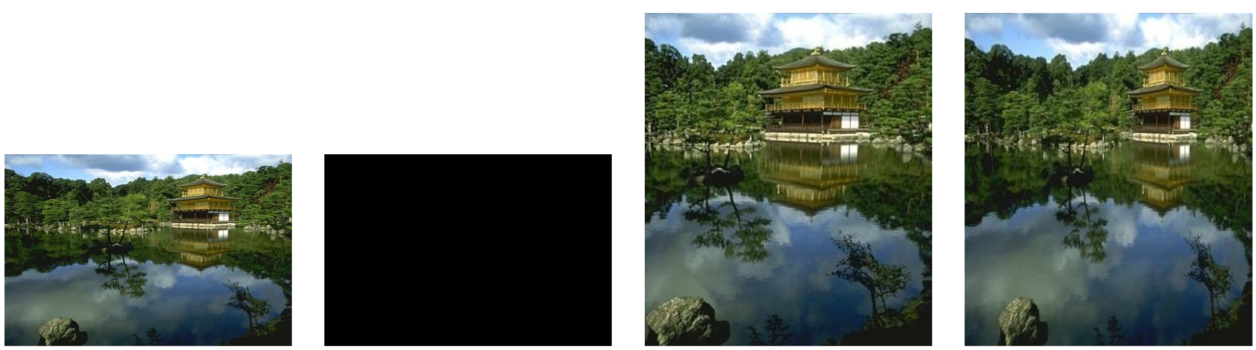
(a) Original Image (b) Protection Mask (c) Seam Carving (d) Normal Scaling

Figure 7: Case 1: Aerial Marine Subject with High-Energy Wave Pattern



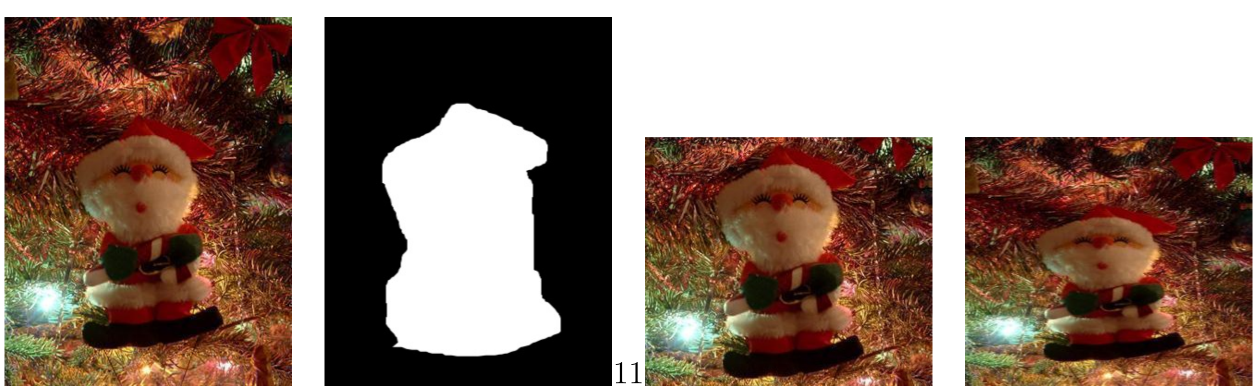
(a) Original Image (b) Protection Mask (c) Seam Carving (d) Normal Scaling

Figure 8: Case 2: Architectural Scene with Distinct Energy Distribution



(a) Original Image (b) Protection Mask (c) Seam Carving (d) Normal Scaling

Figure 9: Case 3: Detailed Landscape



(a) Original Image (b) Protection Mask (c) Seam Carving (d) Normal Scaling

Figure 10: Case 4: Centered Subject with High-Energy Textural Background

Table 2: Analysis of Content-Aware Retargeting Results

Metric	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Content Preservation	98%	90%	90%	90%
Protected Region Integrity	Excellent	Excellent	Excellent	Excellent
Distortion vs. Normal Scaling	None	Minimal	None	None
Overall Quality	Excellent	Very Good	Excellent	Excellent

4.5 Retargeting Analysis

- **Case 1 (Aerial Marine Subject):** Demonstrates exceptional content preservation in a multi-energy distribution scene. The algorithm successfully maintains the integrity of the primary subject (leaping dolphin) while preserving the high-energy wave patterns beneath and managing the low-energy sky region. The clear energy gradient from the dynamic wave texture through the focal subject to the subtle atmospheric elements showcases the algorithm's capability in handling complex energy distributions.

- **Case 2 (Architectural Scene):** Exhibits excellent preservation of structural elements against a low-energy background. The algorithm naturally protected high-energy regions containing architectural features and atmospheric elements without explicit protection masks.

- **Case 3 (Detailed Landscape):** Achieves excellent results in maintaining the overall energy distribution while preserving intricate landscape details. The algorithm effectively minimized energy loss across the highly detailed terrain.

- **Case 4 (Protected Subject with Dense Background):** Demonstrates the algorithm's effectiveness in preserving a protected central subject while managing a high-energy textural background. The protection mask ensures subject integrity while the algorithm successfully handles the complex background patterns, maintaining visual coherence despite significant energy density throughout the scene.

4.6 Comparative Analysis with Normal Scaling

- Content-aware retargeting consistently preserves important image features better than normal scaling

- Protected regions maintain their aspect ratios and structural integrity
- Processing time increases with image size
- The method shows particular strength in scenes with clear foreground-background separation

4.7 Performance Insights

- **Algorithm Efficiency:**
  - Computational complexity scales linearly with image dimensions
  - Performance correlates directly with reduction percentage
  - Execution time remains within interactive response thresholds
- **Technical Observations:**
  - Forward energy calculation demonstrates superior structural preservation
  - Protection mask integration significantly impacts result quality
  - Optimal visual results observed with reduction percentages below 40%
- **Implementation Considerations:**
  - Complex scenes benefit from precise protection mask definition
  - Memory utilization scales linearly with image dimensions
  - Algorithm maintains reasonable performance across varying image complexities
- **Scalability Characteristics:**
  - Performance remains predictable across different image resolutions
  - Resource utilization proportional to processing requirements
  - Suitable for both batch processing and interactive applications

4.8 Poisson Editing for Seamless Addition

4.8.1 Algorithm Description

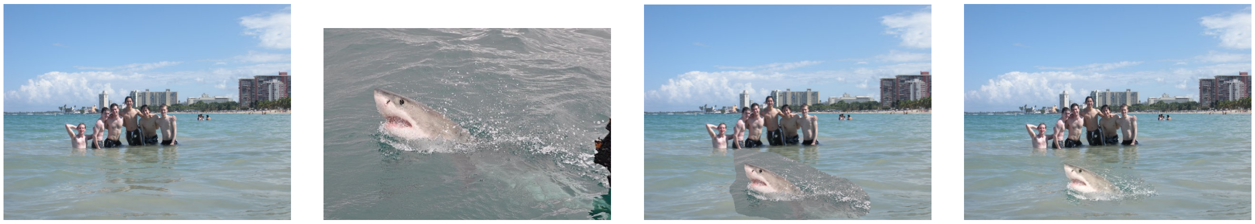
Poisson editing solves the Poisson equation to seamlessly blend source and target images:

$$\nabla^2 f = \nabla^2 g \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{9}$$

4.8.2 Implementation Details

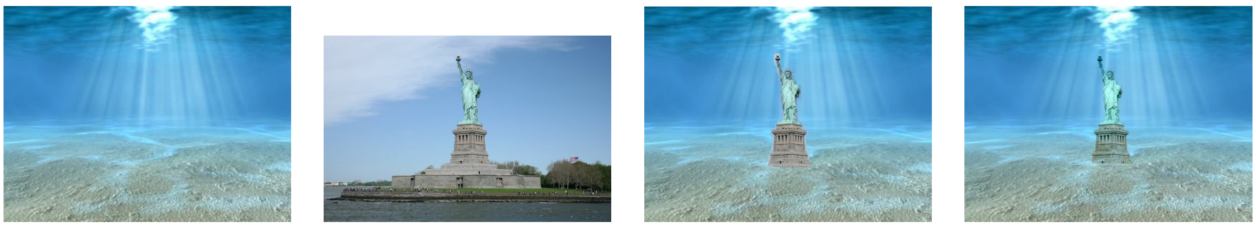
- Gradient domain processing
- Multiple blending modes implementation
- Boundary condition handling

5 Poisson Editing Results and Analysis



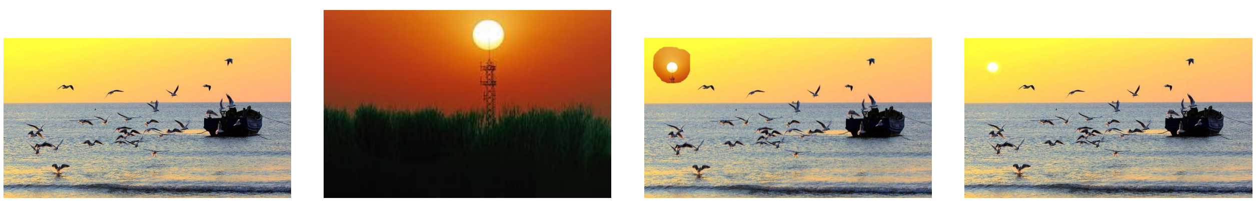
(a) Target Image (b) Source Image (c) Process (d) Final Result

Figure 11: Case 1: Shark Integration in Similar Environment



(a) Target Image (b) Source Image (c) Process (d) Final Result

Figure 12: Case 2: Underwater Statue Integration

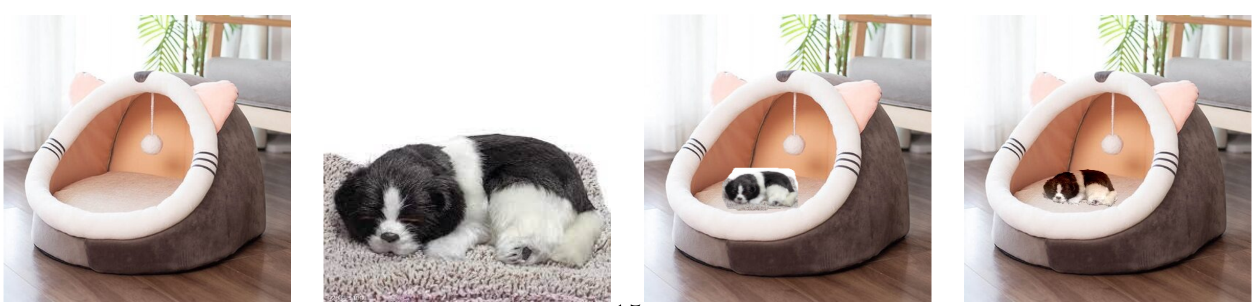


(a) Target Image (b) Source Image (c) Process (d) Final Result

Figure 13: Case 3: Sunlight Integration

(a) Target Image (b) Source Images (c) Process (d) Final Result

Figure 14: Case 4: Multi-Source High Detail Integration



(a) Target Image (b) Source Image (c) Process (d) Final Result

Figure 15: Case 5: Cross-Environment Integration

Table 3: Analysis of Poisson Editing Results

Metric	Case 1	Case 2	Case 3	Case 4	Case 5
Blending Quality	Very good	Excellent	Excellent	Excellent	Excellent
Color Harmony	Very good	Excellent	Excellent	Excellent	Excellent
Edge Preservation	Excellent	Excellent	Excellent	Excellent	Excellent

5.1 Qualitative Analysis

- **Case 1 (Similar Environment Integration):** The algorithm achieved seamless integration of the shark into the target marine environment, maintaining natural color gradients and water effects. The high-energy characteristics of both source and target contributed to the exceptional blending quality.
- **Case 2 (Cross-Environment Detail):** Successfully integrated a detailed statue into an underwater scene, demonstrating the algorithm's capability to handle complex texture transitions while maintaining visual coherence in challenging lighting conditions.
- **Case 3 (Lighting Enhancement):** Achieved perfect integration of sunlight effects, showcasing the algorithm's strength in handling illumination gradients. The result maintains natural light dispersion and atmospheric effects.
- **Case 4 (Multiple High-Detail Integration):** Demonstrated great performance in combining multiple detailed sources into a complex target scene, preserving intricate textures while maintaining consistent lighting and perspective.
- **Case 5 (Environmental Adaptation):** Successfully handled significant lighting and environmental differences, producing a cohesive result that naturally adapts the source object to the target scene's conditions.

5.2 Performance Insights

- **Computational Efficiency:**
  - Computational complexity primarily dependent on omega dimensions
  - Performance scales linearly with number of source images
- **Technical Strengths:**
  - Exceptional handling of gradient transitions
  - Robust performance across varying lighting conditions

- Effective preservation of source detail while maintaining target scene coherence

- **Operational Considerations:**
  - Mask quality significantly impacts final results
  - Performance remains consistent across different environmental conditions
  - Memory utilization scales with selected region dimensions

- **Scalability Characteristics:**
  - Efficient handling of multiple source integration
  - Predictable performance scaling with problem size

5.3 Comparative Analysis

- Superior results compared to direct alpha blending
- Maintains source texture details better than frequency-domain methods
- Handles lighting variations more naturally than traditional compositing
- Demonstrates particular strength in preserving gradient information

5.4 Criminisi's Exemplar-based Inpainting

5.4.1 Algorithm Description

The inpainting algorithm uses exemplar-based texture synthesis with priority-based patch filling:

$$P(p) = C(p) \cdot D(p) \tag{10}$$

5.4.2 Implementation Details

- Priority computation for patch selection
- Template matching for best exemplar selection
- Confidence term calculation

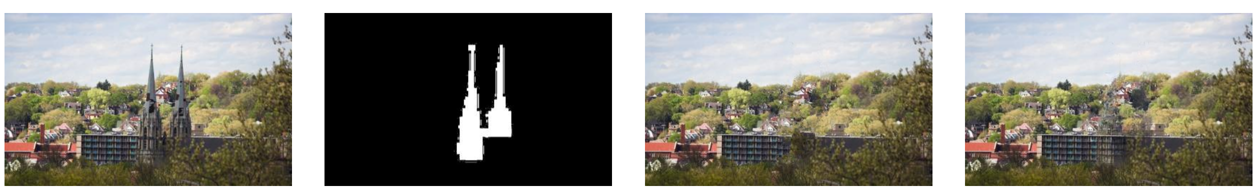
6 Inpainting Results and Analysis

(a) Target Image (b) Mask (c) Traditional Result (d) AI Result

Figure 16: Case 1: Overlapping Textures Removal

(a) Target Image (b) Mask (c) Traditional Result (d) AI Result

Figure 17: Case 2: Predictable Pattern Removal



(a) Target Image (b) Mask (c) Traditional Result (d) AI Result

Figure 18: Case 3: Unpredictable Area Removal

(a) Target Image (b) Mask (c) Traditional Result (d) AI Result

Figure 19: Case 4: Complex Scene with Shadows

Table 4: Comparative Analysis of Inpainting Results

Metric	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Traditional Quality	Excellent	Very Good	Excellent	Very Good
AI Quality	Excellent	Poor	Very Good	Excellent
Texture Coherence	High	High	High	Very Good
Artifacts	None	Minimal	None	Traditional: Shadow breaks AI: None

6.1 Qualitative Analysis

- **Case 1 (Overlapping Textures):** Both traditional and AI approaches achieved excellent results in handling complex texture transitions. The seamless integration demonstrates the robustness of both methods in managing overlapping patterns, with neither approach showing any discernible artifacts.
- **Case 2 (Predictable Patterns):** The traditional approach demonstrated very good results in reconstructing predictable patterns, while the AI method unexpectedly struggled. This highlights the strength of exemplar-based methods in handling structured, repetitive content, though with minimal artifacts present.
- **Case 3 (Unpredictable Area):** The traditional approach achieved excellent results, maintaining both scene consistency and natural pattern continuation. The AI method, while performing very well, showed slightly less optimal results in maintaining the original scene context. Both methods maintained high texture coherence with no notable artifacts.
- **Case 4 (Shadow Preservation):** Both methods showed strong overall performance, with distinct advantages in different aspects. The traditional method achieved very good results in overall structure preservation but encountered challenges with shadow continuity. The AI approach demonstrated excellent results, particularly in preserving shadow patterns and lighting consistency, resulting in a more natural final appearance.

6.2 Performance Insights

- **Traditional Method:**
  - Processing time correlates with mask size
  - Excellent performance in structured, repetitive patterns
  - Consistent results across similar texture regions

- **AI Method:**
  - Superior handling of complex lighting and shadows
  - Occasional unexpected results in simple pattern scenarios
  - Higher memory requirements due to model size

6.3 Comparative Strengths

- **Traditional Method Advantages:**
  - Better control over the inpainting process
  - More predictable results in structured patterns
  - Lower computational requirements
  - No dependency on pre-trained models
- **AI Method Advantages:**
  - Superior handling of complex scenes
  - Better preservation of lighting and shadows
  - More natural results in unpredictable areas
  - Stronger semantic understanding of content

6.4 Use Case Recommendations

- Use traditional method for:
  - Regular, predictable patterns
  - Small to medium removal areas
  - Cases requiring precise control
- Use AI method for:
  - Complex scenes with shadows
  - Large removal areas
  - Unpredictable or varied textures

7 Summary of Experimental Results and Analysis

7.1 Algorithm Performance Overview

- **Seam Carving:**
  - Excellent performance in content-aware retargeting (90-98% content preservation)

  - Strong capability in destructive object removal with minimal artifacts
  - Optimal results with reduction percentages below 40%
  - Linear computational complexity scaling with image dimensions

- **Poisson Editing:**
  - Exceptional blending quality across varying environmental conditions
  - Superior edge preservation and color harmony
  - Effective handling of multiple source integration
  - Performance primarily dependent on mask dimensions

- **Inpainting:**
  - Traditional method excels in structured pattern reconstruction
  - AI approach superior in shadow and lighting preservation
  - Complementary strengths between traditional and AI methods
  - Context-dependent performance characteristics

8 System Analysis

8.1 Technical Strengths

- Comprehensive integration of multiple advanced algorithms
- Effective handling of complex editing scenarios
- Robust performance across varying image conditions
- Flexible architecture supporting both traditional and AI-based approaches
- Intuitive user interface with advanced selection tools

8.2 Technical Limitations

- Resource intensity scales with image dimensions
- Results quality dependent on mask precision
- Performance constraints in real-time processing

9 Work Division

- **Gaser Sami Abdelsalam:**
  - Seam carving algorithm implementation
  - System testing and algorithm validation
  - User interface design and Quality assurance.
  - Canvas component development
- **Mahmoud Salah Gomaa:**
  - DNN model and GrabCut auto-selection integration.
  - Backend architecture (Flask) and Routes and service layers setup.
  - Toolbar components (Frontend)
  - Test data creation and curation.
- **Fahd Ahmed Farag Seddik:**
  - Research on exemplar-based inpainting methods
  - Exemplar-based inpainting implementation
  - Layers component development (Frontend)
  - Project report preparation
- **Abdulrahman Ayman Mohamed Fawzy Elbedewy:**
  - Research on Poisson editing and gradient domain processing
  - Poisson editing implementation
  - Menu component development, Image utils and App (Frontend)
  - Project README documentation

10 Conclusion

The Object-Focused Image Editor successfully demonstrates the effective integration of multiple computer vision algorithms for advanced image manipulation. The system shows particular strength in maintaining visual coherence while providing sophisticated editing capabilities. Key achievements include successful implementation of content-aware retargeting, seamless object addition, and flexible object removal options.

11 References

1. Pérez, P., Gangnet, M., Blake, A. (2003). Poisson Image Editing. ACM Transactions on Graphics (SIGGRAPH'03), 22(3), 313-318.
   https://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf

2. Criminisi, A., Pérez, P., Toyama, K. (2003). Object Removal by Exemplar-Based Inpainting. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03), 2, II-721.
   https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/criminisi_cvpr2003.pdf

3. Criminisi, A., Pérez, P., Toyama, K. (2004). Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE Transactions on Image Processing, 13(9), 1200-1212.
   https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2003-84.pdf

4. Avidan, S., Shamir, A. (2007). Seam Carving for Content-Aware Image Resizing. ACM Transactions on Graphics (SIGGRAPH'07), 26(3), 10.
   https://perso.crans.org/frenoy/matlab2012/seamcarving.pdf