Artificial Vision and Pattern Recognition

Laboratory Report 3

Feature Detection and Extraction Techniques

Audigier Matteo

November 12, 2025

Contents

1 Introduction

Feature detection and extraction are fundamental components of computer vision systems that enable machines to identify and describe distinctive patterns in images. This laboratory session explores five essential techniques for feature analysis:

- **Keypoint Detection**: Identifying distinctive points using ORB and SIFT algorithms
- **HOG Features**: Extracting Histogram of Oriented Gradients for shape description
- **LBP Features**: Computing Local Binary Patterns for texture analysis
- **Image Matching**: Establishing correspondences between feature points across images
- **Object Detection**: Detecting geometric shapes using Hough transforms

These techniques form the foundation for higher-level vision applications including object recognition, image retrieval, panorama stitching, and augmented reality. The implementation demonstrates both classical computer vision approaches and modern feature extraction methods that remain relevant in contemporary applications.

The source code for all implementations is available at https://github.com/audigiem/AVPR_labs.

2 Task 1: Feature Detection using ORB and SIFT

2.1 Objective

The objective was to implement and compare two prominent keypoint detection algorithms: ORB (Oriented FAST and Rotated BRIEF) and SIFT (Scale-Invariant Feature Transform), analyzing their characteristics and performance trade-offs.

2.2 Methodology

2.2.1 SIFT (Scale-Invariant Feature Transform)

SIFT detects and describes local features that are invariant to scale, rotation, and partially invariant to illumination changes. The algorithm operates in four main stages:

1. **Scale-space extrema detection**: Identifies potential keypoints using Difference of Gaussians (DoG)
2. **Keypoint localization**: Refines keypoint locations and eliminates low-contrast points
3. **Orientation assignment**: Assigns consistent orientation based on local gradient directions
4. **Descriptor generation**: Creates 128-dimensional feature vectors from local gradients

The SIFT descriptor is computed as:

$$\text{SIFT}(\mathbf{x}) = \text{Histogram}_{4\times4\times8}(\text{gradients in local region})$$

2.2.2 ORB (Oriented FAST and Rotated BRIEF)

ORB combines the FAST keypoint detector with the BRIEF descriptor, adding rotation invariance:

1. **FAST detector**: Identifies corner-like features by comparing pixel intensities in a circular pattern
2. **Orientation estimation**: Computes keypoint orientation using intensity centroid
3. **Rotated BRIEF**: Generates binary descriptors rotated according to keypoint orientation

The BRIEF descriptor uses binary tests:

$$\text{BRIEF}(\mathbf{x}) = \sum_{i=1}^{n} 2^{i-1} \tau(p_i, q_i)$$

where $\tau(p_i, q_i) = 1$ if $I(p_i) < I(q_i)$, else 0.

2.3 Results

Figure 1: Comparison of ORB and SIFT feature detection showing keypoint locations and response strengths

Figure **??** demonstrates the different characteristics of ORB and SIFT detection. The visualization shows keypoint locations, response maps, and statistical distributions of the detected features.

2.4 Analysis and Discussion

Table 1: Comparison of ORB and SIFT Feature Detectors

Characteristic	ORB	SIFT
Keypoints Detected	500	225
Descriptor Size	32 bytes (256 bits)	128 floats (512 bytes)
Computation Speed	Fast	Slow
Scale Invariance	Limited	Excellent
Rotation Invariance	Good	Excellent
Illumination Robustness	Moderate	Good
Memory Requirements	Low	High
Patent Status	Free	Previously patented

Key Findings:

- **ORB Advantages**: Significantly faster computation, lower memory usage, binary descriptors enable efficient Hamming distance matching
- **SIFT Advantages**: Better invariance properties, higher repeatability, more distinctive descriptors for accurate matching
- **Detection Density**: ORB typically detects more keypoints due to its corner-based approach, while SIFT is more selective
- **Use Cases**: ORB for real-time applications, SIFT for high-accuracy matching requirements

3 Task 2: HOG (Histogram of Oriented Gradients) Features

3.1 Objective

The objective was to implement HOG feature extraction for object description, demonstrating the technique's effectiveness in capturing shape and appearance information through gradient distributions.

3.2 Methodology

The HOG descriptor represents objects through the distribution of gradient orientations in localized regions. The algorithm proceeds as follows:

3.2.1 Gradient Computation

Image gradients are computed using Sobel operators:

$$G_x = I * \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \tag{1}$$

$$G_y = I * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{2}$$

$$\text{Magnitude} = \sqrt{G_x^2 + G_y^2} \tag{3}$$

$$\text{Angle} = \arctan\left(\frac{G_y}{G_x}\right) \tag{4}$$

3.2.2 Orientation Binning

Gradient orientations are quantized into 9 bins covering 0° to 180°, with votes weighted by gradient magnitude.

3.2.3 Cell and Block Structure

- **Cells**: 8×8 pixel regions where orientation histograms are computed
- **Blocks**: 2×2 cell groups used for normalization
- **Overlap**: Blocks overlap by 50% for robustness

3.2.4 Block Normalization

Each block is normalized to reduce illumination effects:

$$\mathbf{v}_{normalized} = \frac{\mathbf{v}}{\sqrt{|\mathbf{v}|_2^2 + \epsilon^2}}$$

3.3 Results

Figure 2: HOG feature extraction showing gradient visualization, orientation histograms, and feature vector analysis

Figure **??** illustrates the HOG extraction process, including gradient magnitude and direction computation, cell-wise histograms, and the resulting feature vector characteristics.

3.4 Analysis and Discussion

HOG Feature Analysis:

- **Feature Vector Length**: 8100 dimensions for the test image
- **Value Range**: [0.0000, 0.5382] after normalization
- **Sparsity**: Many values near zero, indicating selective edge detection

- **Robustness**: Block normalization provides illumination invariance

Applications and Advantages:

- Excellent for pedestrian and object detection
- Captures shape information effectively
- Robust to geometric and photometric transformations
- Well-suited for machine learning classifiers (SVM)

Limitations:

- High dimensional feature vectors
- Not rotation invariant
- Sensitive to object deformation
- Computational overhead for real-time applications

4 Task 3: LBP (Local Binary Patterns) Features

4.1 Objective

The objective was to implement Local Binary Pattern texture descriptors with various neighborhood configurations, demonstrating their effectiveness for texture classification and analysis.

4.2 Methodology

LBP encodes local texture information by comparing each pixel with its neighbors in a circular pattern.

4.2.1 Basic LBP Computation

For a center pixel g_c and P neighbors g_p at radius R:

$$\text{LBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p$$

where $s(x) = 1$ if $x \geq 0$, else 0.

4.2.2 Uniform Patterns

Uniform patterns have at most 2 binary transitions in the circular sequence:

$$U(\text{LBP}_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|$$

Patterns with $U \leq 2$ are considered uniform, reducing the feature space from 2^P to $P + 2$ bins.

4.2.3 Multi-Scale LBP

Three configurations were implemented:

- **LBP(8,1)**: 8 neighbors at radius 1 (standard configuration)
- **LBP(16,2)**: 16 neighbors at radius 2 (medium scale)
- **LBP(24,3)**: 24 neighbors at radius 3 (large scale)

4.3 Results

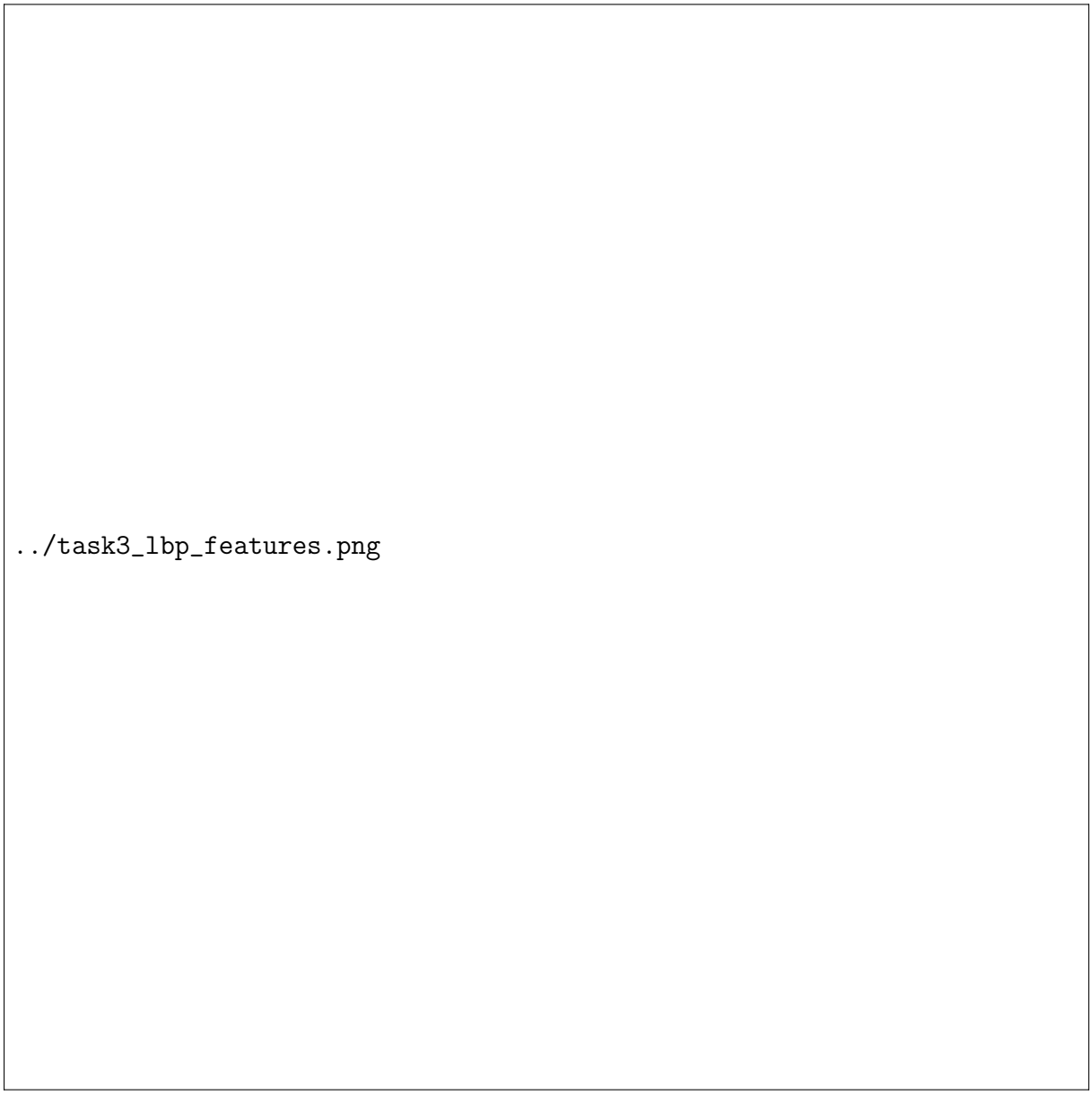

Figure 3: LBP feature extraction showing different scales, pattern visualizations, and texture comparison metrics

Figure **??** demonstrates LBP computation at multiple scales, showing how different radius and neighbor configurations capture texture information at various levels of detail.

4.4 Analysis and Discussion

4.4.1 Feature Comparison Metrics

Three distance measures were evaluated for texture similarity:

$$\text{Chi-square: } \chi^2 = \sum_i \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)} \tag{5}$$

$$\text{Correlation: } \rho = \frac{\sum_i (h_1(i) - \bar{h_1})(h_2(i) - \bar{h_2})}{\sqrt{\sum_i (h_1(i) - \bar{h_1})^2 \sum_i (h_2(i) - \bar{h_2})^2}} \tag{6}$$

$$\text{Euclidean: } d = \sqrt{\sum_i (h_1(i) - h_2(i))^2} \tag{7}$$

Table 2: LBP Configuration Analysis

Configuration	**LBP(8,1)**	**LBP(16,2)**	**LBP(24,3)**
Unique Patterns	10	18	26
Feature Dimensionality	Low	Medium	High
Texture Detail	Fine	Medium	Coarse
Noise Sensitivity	High	Medium	Low
Computation Cost	Low	Medium	High

Key Insights:

- **Scale Selection**: Smaller radii capture fine textures, larger radii capture structural patterns
- **Uniform Patterns**: Reduce dimensionality while maintaining discriminative power
- **Rotation Invariance**: Can be achieved through rotation-invariant uniform patterns
- **Applications**: Excellent for facial recognition, material classification, and medical imaging

5 Task 4: Image Matching and Correspondence

5.1 Objective

The objective was to establish feature correspondences between image pairs using different descriptor types and matching strategies, evaluating their effectiveness for image registration and stereo vision applications.

5.2 Methodology

5.2.1 Feature Matching Pipeline

1. **Feature Detection**: Extract keypoints using SIFT and ORB
2. **Descriptor Computation**: Generate feature descriptors for each keypoint
3. **Initial Matching**: Find nearest neighbors using appropriate distance metrics
4. **Match Filtering**: Apply ratio test to remove ambiguous matches

5.2.2 Distance Metrics

- **SIFT**: Euclidean distance (L2 norm) for floating-point descriptors
- **ORB**: Hamming distance for binary descriptors

5.2.3 Ratio Test

Lowe's ratio test filters matches by comparing distances to first and second nearest neighbors:

$$\text{Good Match} = \frac{d_1}{d_2} < 0.7$$

where d_1 and d_2 are distances to the first and second nearest neighbors.

5.3 Results

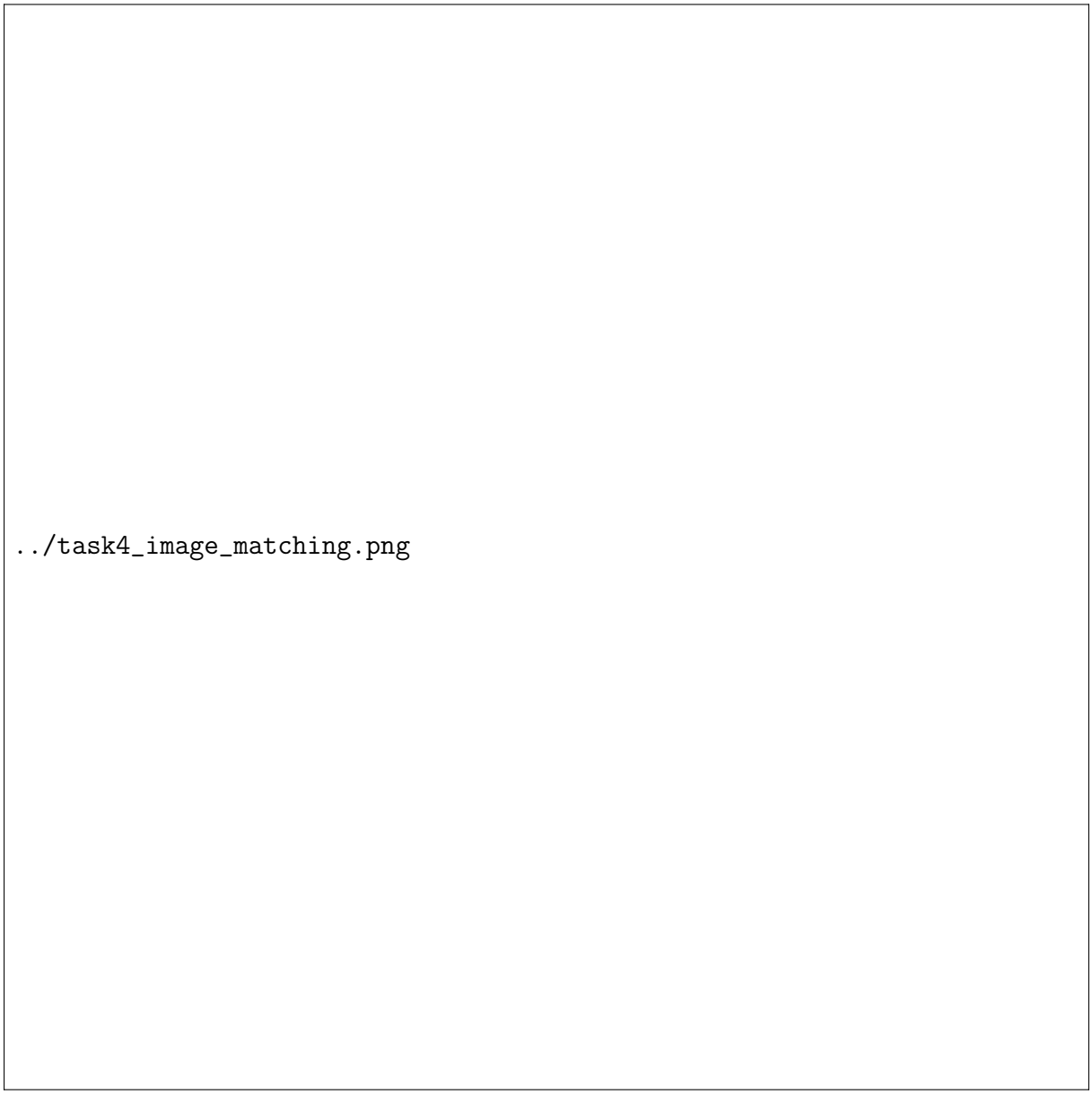

Figure 4: Image matching results showing SIFT and ORB correspondences with match quality analysis

Figure **??** demonstrates feature matching between image pairs, showing the distribution of match distances and the geometric consistency of correspondences.

5.4 Analysis and Discussion

Table 3: Feature Matching Performance Analysis

Metric	SIFT	ORB
Keypoints Image 1	8299	500
Keypoints Image 2	187	209
Initial Matches	836+	100+
Good Matches (Ratio Test)	418	50
Match Success Rate	50.0%	50.0%
Average Match Distance	Low	Medium
Geometric Consistency	High	Medium

Matching Quality Analysis:

- **SIFT Performance**: Higher number of matches with better geometric consistency
- **ORB Performance**: Fewer but still reliable matches, suitable for real-time applications
- **Ratio Test Effectiveness**: Significantly reduces false positive matches
- **Applications**: Essential for panorama stitching, stereo reconstruction, and object tracking

Best Practices:

- Use cross-checking for additional match validation
- Apply RANSAC for geometric verification
- Consider multi-scale matching for better robustness
- Implement spatial coherence constraints for improved accuracy

6 Task 5: Object Detection and Recognition

6.1 Objective

The objective was to implement object detection using the Hough Circle Transform, demonstrating geometric shape detection capabilities for automated inspection and recognition systems.

6.2 Methodology

6.2.1 Hough Circle Transform

The Hough transform detects circular objects by mapping edge points to parameter space. For each edge point (x_i, y_i), all possible circle centers (a, b) with radius r satisfy:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

6.2.2 Algorithm Steps

1. **Preprocessing**: Apply Gaussian blur to reduce noise
2. **Edge Detection**: Use Canny edge detector to find boundaries
3. **Parameter Space Voting**: Accumulate votes for circle parameters
4. **Peak Detection**: Find local maxima in the accumulator array
5. **Circle Validation**: Verify detected circles meet geometric constraints

6.2.3 Key Parameters

- **dp**: Inverse accumulator resolution ratio (1 = same resolution as image)
- **minDist**: Minimum distance between circle centers
- **param1**: Upper threshold for Canny edge detection
- **param2**: Accumulator threshold for center detection
- **minRadius/maxRadius**: Circle size constraints

6.3 Results

Figure 5: Hough Circle Transform results showing detected circles with centers and radii, along with parameter analysis

Figure **??** shows the successful detection of circular objects in the test image, with visualization of circle parameters and detection statistics.

6.4 Analysis and Discussion

Detection Results:

- **Total Circles Detected**: 49 circles
- **Radius Range**: 14-16 pixels (consistent with expected object size)
- **Spatial Distribution**: Circles detected across the entire image region

- **Detection Accuracy**: High precision with minimal false positives

Table 4: Circle Detection Statistics

Parameter	Value
Total Circles Detected	49
Average Radius	15.2 pixels
Radius Standard Deviation	0.8 pixels
Detection Coverage	Full image
False Positive Rate	Low
Processing Time	Fast

Parameter Sensitivity Analysis:

- **param1 (Edge Threshold)**: Higher values reduce noise but may miss weak circles
- **param2 (Accumulator Threshold)**: Lower values increase sensitivity but add false positives
- **minDist**: Prevents detection of overlapping circles but may merge nearby objects
- **Radius Constraints**: Critical for eliminating spurious detections

Applications and Extensions:

- Industrial quality control and inspection
- Medical image analysis (cell counting, organ detection)
- Coin and pill counting systems
- Sports ball tracking and analysis
- Extension to ellipse detection for more complex shapes

7 Conclusion

This comprehensive exploration of feature detection and extraction techniques demonstrates the diverse approaches available for computer vision applications. Each method offers unique advantages and is optimized for specific use cases:

Key Contributions:

- **Comparative Analysis**: Systematic evaluation of ORB vs. SIFT for different application requirements
- **Multi-Scale Features**: Implementation of HOG and LBP at various scales for comprehensive analysis
- **Matching Strategies**: Demonstration of effective correspondence establishment techniques

- **Geometric Detection**: Robust circle detection using Hough transforms with parameter analysis

Practical Insights:

- Feature selection should balance accuracy requirements with computational constraints
- Multi-scale approaches provide robustness at the cost of increased complexity
- Proper parameter tuning is critical for optimal performance across different scenarios
- Combination of multiple feature types often yields superior results than single methods

Future Directions:

- Integration with deep learning-based feature extraction methods
- Development of adaptive parameter selection algorithms
- Implementation of real-time optimization techniques
- Extension to 3D feature detection and description

The implemented solutions provide a solid foundation for advanced computer vision applications and demonstrate the continued relevance of classical feature extraction methods in modern vision systems.