كلية العلوم و التقنيات بطنجة
Faculté des Sciences et Techniques de Tanger

University Abdelmalek Essaâdi

Faculty of Sciences and Technologies Tangier

Content-Based Image Retrieval System

Using Shape and Texture Features

Course:
Multimedia mining and indexing
Master Semester 3

Professor:
Dr. M'hamed AIT KBIR

Academic Year:
2025-2026

Submitted by:
Mouhsin Boumargoud

November 15, 2025

Contents

Abstract

This report presents a content-based image retrieval (CBIR) system that extracts and compares images based on shape and texture features. The system implements Fourier descriptors and direction histograms for shape analysis, and Gabor filters with Tamura features for texture characterization. Euclidean distance is used to measure similarity between feature vectors, enabling efficient retrieval of visually similar images.

1 Methodology

1.1 System Architecture

The CBIR system consists of two main programs:

1. **Feature Extraction Program (a_extract_features.py):** Processes all images in a folder, extracts feature vectors, and saves them as JSON files.
2. **Image Search Program (b_image_search.py):** Takes a query image and retrieves the 6 most similar images based on feature vector distances.

1.2 Workflow

The processing pipeline follows these steps:

1. **Image Loading:** Read images using OpenCV
2. **Feature Extraction:**
   - Shape features: Contour detection, Fourier descriptors, direction histogram
   - Texture features: Gabor filter responses, Tamura features
3. **Feature Storage:** Save features as JSON files with same name as image
4. **Similarity Search:** Compute distances between query and database images
5. **Result Display:** Show top-6 similar images sorted by distance

2 Theoretical Foundation

2.1 Shape Features

2.1.1 Contour Extraction

The largest contour is extracted using:

```
contours, _ = cv2.findContours(binary,
                 cv2.RETR_EXTERNAL,
                 cv2.CHAIN_APPROX_NONE)
largest_contour = max(contours, key=len)
```

2.1.2 Fourier Descriptors

Contour points are represented as a complex function:

$$z(t) = x(t) + jy(t) \tag{1}$$

The Discrete Fourier Transform (DFT) is applied:

$$Z(k) = \sum_{t=0}^{N-1} z(t)e^{-j2\pi kt/N} \tag{2}$$

Properties of Fourier descriptors:

- **Translation invariance:** Set $Z(0) = 0$
- **Scale invariance:** Normalize by $|Z(1)|$
- **Rotation invariance:** Use magnitude only: $|Z(k)|$

The first 20 coefficients are retained as features.

2.1.3 Direction Histogram

Contour directions are quantized into 8 bins (45° each):

$$\theta_i = \arctan 2(\Delta y_i, \Delta x_i) \tag{3}$$

The normalized histogram captures the distribution of edge orientations.

2.2 Texture Features

2.2.1 Gabor Filters

Gabor filters combine spatial and frequency domain analysis:

$$g(x, y; \theta, \lambda) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda}\right) \tag{4}$$

where:

$$x' = x \cos\theta + y \sin\theta$$
$$y' = -x \sin\theta + y \cos\theta$$

Parameters:

- Orientations: $\theta \in \{0, 45, 90, 135\}$
- Wavelengths: $\lambda \in \{3, 5, 7\}$ pixels
- 12 filters total (4 orientations $\times$ 3 scales)

For each filter, mean and standard deviation of the response are computed, yielding 24 features.

2.2.2 Tamura Features

Three key Tamura features are extracted:

1. Coarseness (Granularity):

$$F_{crs} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} 2^{k^*(i,j)} \tag{5}$$

where k^* is the window size that maximizes differences between non-overlapping neighborhoods.

2. Contrast:

$$F_{con} = \frac{\sigma}{\alpha_4^{1/4}} \tag{6}$$

where σ is standard deviation and α_4 is kurtosis.

3. Directionality:

Computed from gradient direction histogram using Sobel operators:

$$\Delta_H = \frac{\partial I}{\partial x}, \quad \Delta_V = \frac{\partial I}{\partial y} \tag{7}$$

$$|\nabla I| = \sqrt{\Delta_H^2 + \Delta_V^2}, \quad \theta = \arctan\left(\frac{\Delta_V}{\Delta_H}\right) \tag{8}$$

A 16-bin histogram of gradient directions is constructed, and directionality measures the histogram's peakedness.

2.3 Feature Vector Composition

The combined feature vector contains:

- 20 Fourier descriptors
- 8 direction histogram bins
- 24 Gabor features (12 filters × 2 statistics)
- 3 Tamura features (coarseness, contrast, directionality)
- 16 Tamura direction histogram bins

Total: 71 dimensions

2.4 Distance Metrics

Euclidean Distance:

$$d_{euclidean}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{9}$$

Cosine Distance:

$$d_{cosine}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} \tag{10}$$

3 Results

3.1 Shape Dataset (Formes)

The shape dataset contains silhouettes of objects (camels, devices, bells, etc.).

Key Observations:

- Fourier descriptors effectively capture object contour shapes
- Similar objects (e.g., camel-1, camel-2, camel-4) cluster together
- Direction histograms distinguish between horizontal and vertical orientations
- High contrast values (100-108) due to binary nature of silhouettes

Sample Results:
Query: `camel-1.gif`

Rank	Image	Distance
1	camel-2.gif	12.45
2	camel-4.gif	18.73
3	device0-1.gif	45.21
4	device0-2.gif	52.89
5	bell-1.gif	67.34
6	device0-4.gif	71.56

Table 1: Retrieval results for camel-1.gif (shape dataset)

3.2 Texture Dataset (Textures)

The texture dataset contains natural textures with varying patterns and scales.

Key Observations:

- Gabor filters capture oriented texture patterns effectively
- Tamura coarseness ranges from 3.5 to 11.7, reflecting texture granularity
- Tamura contrast ranges from 14.4 to 47.9, showing intensity variation
- Directionality values distinguish isotropic from directional textures

Sample Results:
Query: `Im01.jpg`

Rank	Image	Distance
1	Im02.jpg	3.87
2	Im03.jpg	4.52
3	Im04.jpg	5.19
4	Im05.jpg	6.34
5	Im10.jpg	7.82
6	Im12.jpg	8.91

Table 2: Retrieval results for Im01.jpg (texture dataset)

3.3 Performance Analysis

Feature Dimensionality:

- Shape features: 28 dimensions (20 Fourier + 8 directions)
- Texture features: 43 dimensions (24 Gabor + 19 Tamura)
- Combined: 71 dimensions

Computational Efficiency:

- Feature extraction: ~50-100ms per image
- Distance computation: <1ms per comparison
- Search in 30-image database: <50ms

Retrieval Accuracy:

- Shape dataset: High precision for similar object classes
- Texture dataset: Good discrimination between texture patterns
- False positives mainly occur when textures have similar statistics but different visual appearance

3.4 Feature Importance

Analysis of feature contributions shows:

1. **Fourier descriptors:** Most discriminative for shape (highest variance)
2. **Gabor features:** Critical for texture orientation
3. **Tamura coarseness:** Distinguishes fine vs. coarse textures
4. **Direction histograms:** Complementary to other features

4 Conclusion

The implemented CBIR system successfully combines shape and texture features for image retrieval. Key achievements:

- **Robust feature extraction:** Fourier descriptors provide translation, scale, and rotation invariance
- **Multi-scale analysis:** Gabor filters capture texture at multiple orientations and scales
- **Perceptual features:** Tamura features align with human texture perception
- **Efficient retrieval:** Fast distance computation enables real-time search

Limitations and Future Work:

- Current system uses equal weighting for all features; learned weights could improve accuracy
- Color information is not utilized; HSV/Lab color features could enhance discrimination
- Deep learning features (CNN embeddings) could provide more semantic similarity
- User relevance feedback could refine retrieval results iteratively

References

1. Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1), 1-19.
2. Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6), 460-473.
3. Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), 837-842.
4. Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools.*