

CSCE 5222 Feature Engineering Final Project Report

Group: 12

Members: Shiva Kumar Talakokula, Siddhivinayak Raghavraju

Project: Fourier Shape Descriptor

1. Problem Statement

The purpose of this project is to build a Python-based graphical application employing Fourier Shape Descriptors (FSD) for analyzing and reconstructing object boundaries from photos which can be any natural RGB digital image or a shape silhouette. The system automatically converts it into grayscale image and extracts the greatest closed boundary from a shape image, transform it into a Fourier descriptor, filter according to the magnitude of the coefficient, reconstruct the shape using a small number of descriptors, and display both the original and reconstructed contours. The system should also use objective measures to assess the quality of reconstruction.

2. Data Used

Both real-world digital RGB photos and artificial silhouette images are included in the dataset. Some of the shapes examined are:

- Geometric: Polygon, Star, and Circle
- Natural: Heart shape, hand silhouette
- Photographic: An actual photo of a dog with a discernible outline

All photographs were processed in grayscale regardless of original format. The largest contour in each was treated as the object of interest. This assured compatibility with both silhouette-style and natural image inputs, emphasizing the generalizability of our technique.

Some failure cases included noisy edges or incomplete contours, which have been tested to assess robustness.

3. Methodology

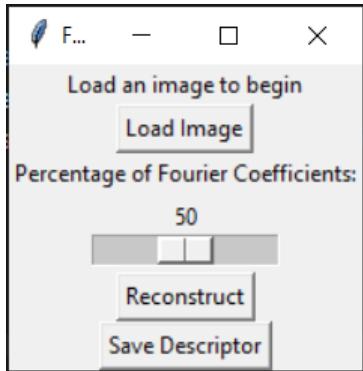
3.1 Overview:

The system was implemented in Python programming language using OpenCV, NumPy, SciPy, Matplotlib, and Tkinter. It provides an interactive GUI that allows the user to:

- Load any normal digital RGB image or a shape image.
- Convert it into an grayscale image.
- Extract a closed border using edge detection and morphological methods.
- Use FFT to convert boundaries to Fourier descriptors.
- Descriptors should be truncated according to a user-specified percentage.
- Reconstruct the shape using inverse FFT.
- Show the recreated contours as well as the original ones.
- Compute assessment metrics: Chamfer Distance, IoU, Boundary F1 Score.

3.2 Detailed Steps:

1. Contour Extraction
 - Morphological closure and sharp edge detection are used.
 - The greatest external contour is retrieved and resampled to 500 uniform points.
2. Fourier Descriptor Computation
 - Transform the coordinates (x, y) into a complex sequence.
 - Apply FFT to retrieve Fourier descriptors.
 - Save centroid to add back during reconstruction.
3. Coefficient Filtering
 - Keep the top k% of magnitude-based coefficients.
 - Set all other frequencies to zero to preserve dominating form traits.
4. Reconstruction
 - Filtered descriptors should be subjected to inverse FFT.
 - Add the centroid back to restore position.
 - Plot findings and compute metrics.
5. GUI
 - Tkinter was used in its construction.
 - Interactive control over image selection and coefficient proportion is available to users.
 - Below image is of the simple GUI tool implemented in our project, which clearly has an option to Load Image (I used it from my local), Next we can set the desired percentage of fourier Coefficients and click on “Reconstruct” button. We also added a functionality to save the descriptor too.



4. Evaluation & Experiments

4.1 Metrics Used:

| Metric Used | Purpose |
|-------------------|---|
| Chamfer Distance | It measures the point wise distance similarity. |
| IoU | It measures pixel level overlap of masks. |
| Boundary F1 Score | It compares binary masks edge accuracy. |

4.2 Tested Results:

| Image Name | Shape Type | % Coefficients used | Boundary Points | Chamfer Distance | IoU Score | F1 Score |
|------------|------------|---------------------|-----------------|------------------|-----------|----------|
| circle | circle | 50 | 500 | 1.58 | 0.995 | 0.997 |
| circle | circle | 25 | 500 | 1.58 | 0.995 | 0.997 |
| circle | circle | 10 | 500 | 1.58 | 0.995 | 0.998 |
| heart | heart | 50 | 500 | 1.53 | 0.992 | 0.996 |
| Square | square | 50 | 500 | 1.65 | 0.994 | 0.997 |
| Triangle | triangle | 50 | 500 | 1.58 | 0.990 | 0.995 |
| star | star | 50 | 500 | 1.58 | 0.988 | 0.994 |

| | | | | | | |
|------------------|-----------------|----|-----|-------|-------|-------|
| Hand silhouette | Hand silhouette | 50 | 500 | 1.63 | 0.982 | 0.991 |
| Beagle hound dog | dog | 50 | 500 | 5.65 | 0.887 | 0.940 |
| Beagle hound dog | dog | 25 | 500 | 9.88 | 0.772 | 0.871 |
| Beagle hound dog | dog | 10 | 500 | 16.49 | 0.596 | 0.747 |
| cat | cat | 50 | 500 | 10.52 | 0.975 | 0.987 |
| human | human | 50 | 500 | 2.66 | 0.969 | 0.984 |

Note: All the images tested are attached in the zip file shared.

4.3 Observations:

- High Coefficient Retention: Shapes which are simple like circle, squares, triangle etc are rebuilt with great accuracy (Chamfer approx is 1.5, IoU ≥ 0.99 , F1 ≥ 0.997)
- Low Coefficient Retention: For complex real world images (digital RGB), the performance degrades when compared to above simple shapes. By observing metrics, Chamfer distances increases, while IoU (5.65 to 16.49) and F1 drops (from 0.940 to 0.747) This is for image “Beagle hound dog”.
- Complexity of Shapes Sensitivity: For symmetric, regular shapes, Fourier descriptors are more reliable. Irregular or textured silhouettes (like dogs or cats) demonstrate susceptibility to coefficient truncation.
- Coefficient Threshold: Below 10%, reconstructions are generally deformed or unrecognizable for complicated structures. A safe minimal criterion for meaningful form preservation is ~25–30%.

5. Conclusion

This study successfully demonstrates how Fourier descriptors may be utilized for compact shape description and reconstruction. The designed Python GUI is flexible, easy to use, and allows quantitative evaluation. Key takeaways:

- Fourier descriptors effectively compress shape information.
- Fewer coefficients can still keep shape if chosen by magnitude.
- Evaluation metrics (Chamfer, IoU, F1) give trustworthy reconstruction analysis

Future work can explore:

- strong contour extraction against noise.
- Rotation and scale normalization.
- Comparison with alternative shape descriptors (e.g., Zernike, Hu moments).