Object Detection in Images: Looking for the Elephant

Giovanni Mantovani and Andrea Palmieri

April 5, 2024

1 Introduction

The objective of this report is to show the pipeline used for object detection using the Speeded Up Robust Features (SURF) algorithm, then extracting Keypoints Description, followed by Feature Matching and finally Geometry Consistency Check. The objective is to detect an elephant from a template image (Figure 1) in a scene image of a cluttered desk (Figure 2). The bounding box of the elephant in the template image was manually drawn in order to better analyse the performance of each detection pipeline with different parameters.

Figure 1: Template image of the elephant.

Figure 2: Scene image of a cluttered desk

2 Methodology

The methodology employs the SURF algorithm to detect and precisely outline an elephant within a cluttered scene. The process can be divided into several systematic steps:

1. **Feature Detection:** The SURF algorithm was applied to both the target image (elephant) and the scene image to detect keypoints. The 'MetricThreshold' parameter of the function *detectSURFfeatures* was adjusted to ensure a balance between detecting sufficient features and maintaining feature relevance. We experimented with three different setups, using 'MetricThreshold' values of 250, 500, and 1000, to identify the most effective setting for our context.

2. **Feature Matching and Geometric Consistency Check:** Feature descriptors were extracted for the detected keypoints, followed by a matching process to establish correspondences between the images. The 'MatchThreshold' parameter of the function *matchFeatures* was adjusted, with experiments conducted using values of 1 and 2.5, to optimize the matching process. Sequentially, an affine transformation was used to assess the geometric consistency of the matched features, ensuring that only those features that align precisely between the images were retained.

3. **Outline Refinement:** To delineate the elephant's shape within the scene, a detailed outline of 40 edges was manually created around the elephant in the template image, as shown in Figure 3. This outline was then transformed and overlaid onto the scene image to accurately depict the elephant's location and shape.

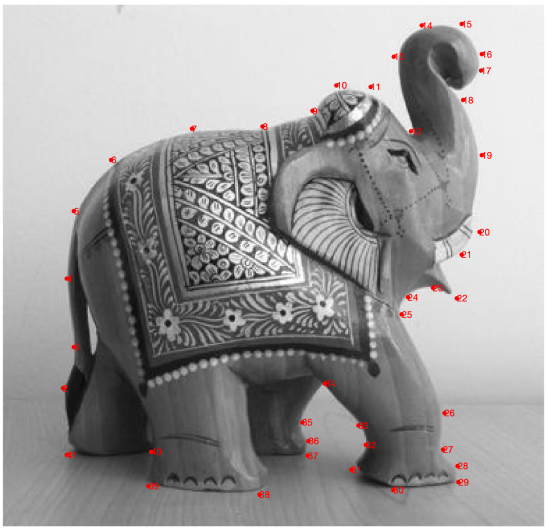

Figure 3: Refined outline of the elephant with optimized parameter settings.

3 Results

In this section, we present the results of the object detection pipeline for different combinations of 'MetricThreshold' and 'MatchThreshold'. They are shown in the following images, showcasing the impact of these parameters on the feature detection and matching process.

Figure 4: Feature detection with 'MetricThreshold' set to 250 and with 'MatchThreshold' set to 1

Figure 5: Feature detection with 'MetricThreshold' set to 250 and with 'MatchThreshold' set to 2.5

Figure 6: Feature detection with 'MetricThreshold' set to 500 and with 'MatchThreshold' set to 1

Figure 7: Feature detection with 'MetricThreshold' set to 500 and with 'MatchThreshold' set to 2.5

Figure 8: Feature detection with 'MetricThreshold' set to 1000 and with 'MatchThreshold' set to 1

Figure 9: Feature detection with 'MetricThreshold' set to 1000 and with 'MatchThreshold' set to 2.5

4 Conclusion

The results show that the object detection pipeline chosen was capable of detecting the elephant in the scene image. However, detection had some issues in dealing with the scene perspective, as the photos are shot from two different angles with respect to the elephant. Nevertheless, the results are satisfying, and they demonstrate the effectiveness of the object detection pipeline in identifying the elephant within the scene image despite the challenges posed by variations in perspective.