# OBJECT INSTANCE RECOGNITION - ASSIGNMENT 3

**Mirko Morello**
920601
m.morello11@campus.unimib.it

**Andrea Borghesi**
916202
a.borghesi1@campus.unimib.it

January 3, 2025

## 1 INTRODUCTION

In this task, our objective was to find an elephant toy in an image representing a cluttered desk starting from a template image of the toy. Such task is known as *Object instance recognition* where the objective is to find a specific object in an image.

## 2 TECHNOLOGIES USED: SPEEDED UP ROBUST FEATURES (SURF)

SURF, partly inspired by SIFT, is a robust feature descriptor and detector. It is said to be robust since it is rotation, scale, and partly affine invariant.

## 3 APPROACH

### 3.1 Bounding polygon mask

We started by manually creating a mask by tracing the outline of the elephant composed of 128 points. This mask has been saved in a CSV file once, to then be reused in each run of the experiment to obtain more consistent and comparable results with higher precision than a simple bounding box.
This gives us more precise feedback on how well the algorithm is at identifying the object.

### 3.2 Hyperparameters

The hyperparameters we experimented with belonged to 3 different functions:

- `detectSURFFeatures`
- `matchFeatures`
- `estimateGeometricTransform`

#### 3.2.1 `detectSURFFeatures`

This function can be thought of as the main function, its hyperparameters define how the feature detector will behave, we had a wide range of choices:

- `NumOctaves`, the higher it is, the larger blobs the algorithm will find. The default is 3.
- `MetricThreshold`, its range is between 0 and infinite, the lower the value the higher the number of blobs retrieved. The default is 1000
- `NumScaleLevels`, its range is between 0 and infinite, the higher it is, the more blobs are detected at a finer scale. The default is 3.

We ended up settling with `NumOctaves=12`, `MetricThreshold=100`, `NumScaleLevels=10` that, with our empiric observations, led to the best results. This can be explained because this choice of hyperparameters allows us to retain a great number of blobs

taking a look at the texture of the elephant, we're not looking at finer details but at a slightly bigger scale.

Metric Threshold has been set to a lower value than the default one because we wanted to retain a much higher number of candidate blobs as we noticed that in the rear part of the elephant we didn't match many points, shifting our prediction out of scope. Such a choice would usually introduce a significant amount of noise but, since our template image has been taken over a flat background it did introduce almost any. This allowed us to have a greater amount of candidate keypoints to work with.
NumScaleLevels has been boosted from the default because we wanted to retrieve more points related to the unique texture of the elephant's saddle.
Its counterpart, NumOctaves, is also larger than the

default but its purpose is to identify broader and larger features.

The default parameters find 272 and 1129 blobs for the template and the desk image respectively, while ours find 1614 blobs for the template and 5819 for the desk image.

### 3.2.2 `matchFeatres`

This function finds corresponding interest points between a pair of images using local neighborhoods and the Harris algorithm. Its hyperparameters are:

- `Method`, can be either Exhaustive or Approximate, it specifies how the nearest neighbors between the first and second array of features are found. The default is Exhaustive.

- `MatchThreshold`, the range is between 0 and 100, it represents a percentage of the distance from the perfect match. The default is 10.

- `MaxRatio`, the range is between 0 and 1, it's used to reject ambiguous matches, the higher the value the more matches returned. The default is 0.6.

- `Unique`, can be either True or False, when setted to False multiple features of the first array of features can match to a single feature in the second array. The default is False.

- `Metric`, it's the type of metric used for feature matching, can be either SAD (Sum of absolute differences) or SSD (Sum of squared differences). The default is SSD.

```
Method="Exhaustive", MatchThreshold=30,
MaxRatio=1, Unique=false, Metric="SSD"
```

Our choice of parameters, similarly to the last section, aims to retain a greater number of matched point, at the cost of including false positives. We expect to mitigate this false positive while retaining the true positives in the next section with the geometric transformation. Initially, given the manageable size of the feature set, we opt for an exhaustive pairwise search. To enhance the retention of matches, we also opt for a higher match threshold, determined empirically. Similarly, we set the max ratio parameter to its maximum value to retain ambiguous matches. As depicted in Figure 1, this approach yields a considerable number of matches, including both false positives and potential candidates, surpassing the matches obtained with the default hyperparameters, as illustrated in Figure 2.

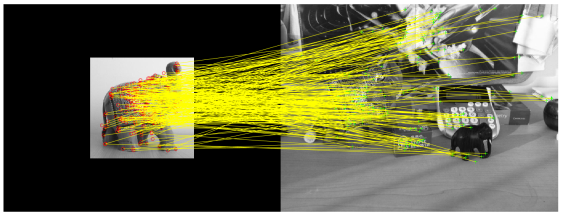

**FIGURE 1:** Matched features with our choice of parameters

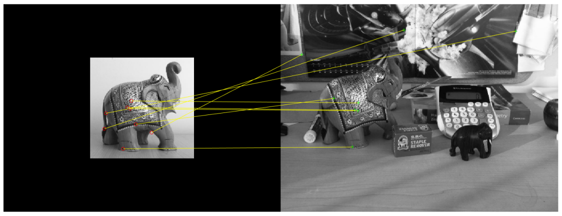

**FIGURE 2:** Matched features with the default parameters

### 3.2.3 `estimateGeometricTransform`

This function tries to estimate the geometric transformation that best aligns the template image (elephant) within the scene image (cluttered desk). To find such a transformation we used RANSAC (RANdom SAmple Consensus), aligning the keypoints while minimizing the error between them.
To get a better result from this we experimented with the following hyperparameters:

- `TransformType`: This defines what type of transformation will be applied to the template when fitting it to the image. We had 3 choices and each had a different degrees of freedom: similarity, affine, and projection.

- `Confidence`: Confidence in finding the maximum number of inliers. The higher this value, the better the robustness of the results but at the expense of additional computations. The default is 99

- `MaxDistance`: The distance from a matched point to its corresponding projected point, obtained through the inverse geometric transform. The default is 1.5.

- `MaxNumTrials`: Maximum number of random trials for finding the inliers. The default is 1000.

We settled for:
`TransformationType = "similarity",`
`Confidence = 0.99, MaxDistance = 30,`
`MaxNumTrials = 10’000.`
At this stage, we aimed for both quality and robustness of results.
Since the analyzed image has the same perspective as the template one the best two types for the transformation matrix are “similarity” and “affine”. Also, since the elephant has a similar rotation in both images, “similarity” seems to provide good enough results. Although an affine transformation can theoretically track exactly the slight rotation, we found the similarity matrix to be the most robust choice and the results are not significantly different.
We set the confidence as high as possible (it is also its default value) since lowering it caused the presence of many false positives, lowering both precision and robustness, and we didn’t have to worry about the expense of the computation.
The maximum distance has been found empirically, that value seemed to be enough to have some play during the trials but also not too high to allow distant inliers to be considered.
Lastly, we increased the number of trials to achieve more robust results as we empirically concluded that the default value of 1000 tended to have a significantly higher variance in the results.
As shown in Figure 3 the retained keypoints with the default hyperparameters after the geometric transformation are much less than the ones retained by our choice of hyperparameters in Figure 4.

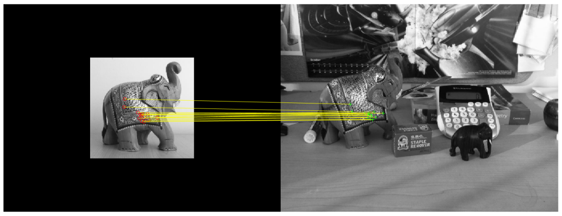

**FIGURE 3:** Retained features with the default parameters

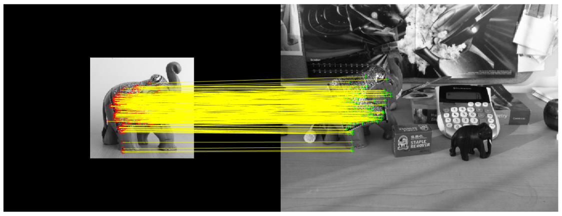

**FIGURE 4:** Retained features with our choice of parameters

The combination of our choice of parameters for the aforementioned functions leads to the result shown in figure 5. The elephant toy is properly found, although the outline does not fit perfectly. This might be due to the repeating patterns in the saddle of the elephant, as this technique of object detection does not perform well in detecting uniform and repeating patterns. Nonetheless, our results correctly identify the object, and such results are consistent over multiple runs of the experiment.

## 4 RESULTS

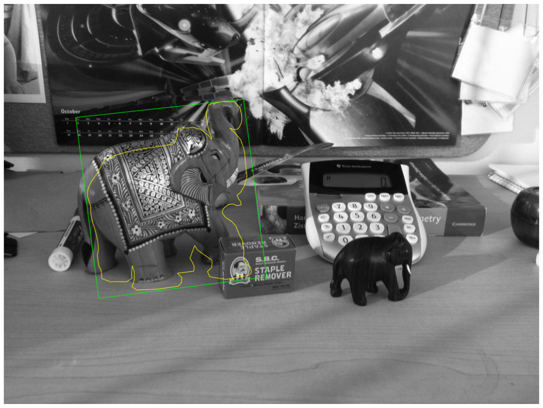

**FIGURE 5:** Localization of the position of the elephant