LAB 3 REPORT

A PREPRINT

Daniele cecca
Artificial Intelligence for Science and Technology
Milano Bicocca University
Supervised Learning

April 5, 2024

1 Introduction

Object instance detection is a critical task in computer vision, with applications ranging from robotic navigation to image retrieval. The ability to discern a specific object in a cluttered backdrop is challenging due to variable object positioning, scales, and orientations. This study leverages the SIFT algorithm to extract distinct features that are invariant to such changes, thereby enabling the successful identification of a template object—an elephant in this case—within a cluttered desk image.

2 Scale invariant Feature Transform-SIFT

Local, salient regions in an image can be described by point features, which are vectors. They can be used to describe and match images taken from different viewpoints.Features should be invariant to perspective effects and illumination and the same point should have similar vectors independent of pose or viewpoint.

One algorithm that provides us with what we need is the Scale Invariant Feature Transform (SIFT). SIFT constructs a scale space by iteratively filtering the image with a Gaussian and scaling the image down at regular intervals. Adjacent scales are subtracted, yielding Difference of Gaussian (DoG) images. Difference of Gaussian filters are blob detectors ; the interest points (blobs) are detected as extrema in the resulting scale space.

After extracting the interest points, SIFT rotates the descriptor to align with the dominant gradient orientation. Then, gradient histograms are computed for local sub-regions of the descriptor which are concatenated and normalized to form a 128D feature vector (the keypoint descriptor). These operations make the descriptor invariant to rotation and brightness changes.

By now, many algorithms for feature detection and description have been developed, e.g., SIFT, SURF, U-SURF, BRISK, ORB, FAST, and recently deep learning based ones.

3 RANSAC

The RANSAC algorithm is used for estimating the parameters of models in images (i.e., model fitting). The basic idea behind RANSACis to solve the fitting problem many times using randomly selected minimal subsets of the data and choosing the best perform ing fit. To achieve this, RANSAC tries to iteratively identify the data points that correspond to model we are trying to fit.

The RANSAC algorithm aims to address this challenge by identifying the "inliers" and "outliers" in the data. RANSACrandomly selects samples of the data, with the assump tion that if enough samples are chosen, there will be a low probabil ity that at all samples provides a bad fit

4 Methodology

1. **Image Reading and Display:** The elephant template and the cluttered desk scene images are read using MATLAB's image processing toolbox. These images are displayed to provide a visual basis for subsequent detection.
2. **Keypoint Detection:** SIFT features, which are robust against scale and rotation, are detected for both the template and the scene images.

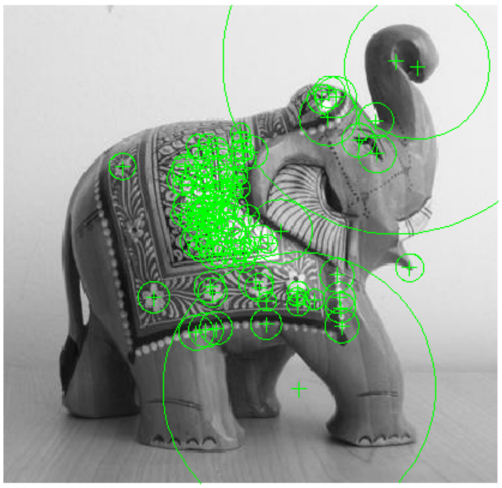

Figure 1: SIFT

3. **Keypoint Description:** Unique descriptors are extracted for each keypoint to facilitate accurate matching.
4. **Feature Matching:** An optimized feature matching process is conducted, allowing for a distinct correlation between the template and the scene features.
5. **Geometric Consistency Check:** An affine transformation is estimated based on the matched features to ascertain the geometric consistency and derive the object's position in the scene. The function excludes outliers using the M-estimator SAmple Consensus (MSAC) algorithm. The MSAC algorithm is a variant of the Random Sample Consensus (RANSAC) algorithm.

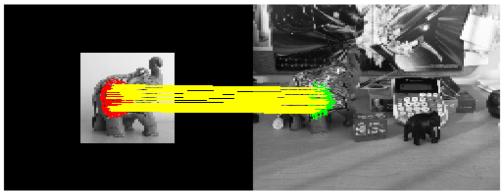

Figure 2: Geometric Consistency Check after matching

6. **Bounding Box Drawing:**

   We have constructed a square bounding box that identifies the elephant. The result is shown in Figure 2.

   The detected keypoints serve to construct a polygonal bounding box around the object identified in the scene, mirroring the shape of the elephant. Our approach involved sorting all the matched points according to their y-coordinates, and these points were then incorporated into the initial bounding box to form a polygon that approximates the outline of the elephant. This method proves effective due to the unique shape of the elephant, which typically features a singular highest point and a single lowest point. If the object presented multiple peaks and troughs, a different sorting strategy, such as ordering by the L2 norm (Euclidean distance from the origin), might have been required to better approximate its shape.

   As shown in Figure 3, this approach was not so effective, so we decided to use the ginput() function. This allows the user to directly click on the image to obtain the coordinates, which are then used to draw the shape of the elephant. These coordinates are subsequently transformed based on the computations done previously. The result is shown in Figure 4.

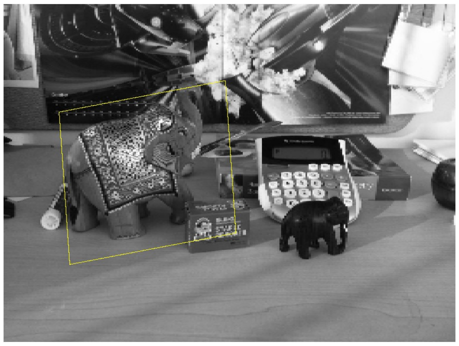

Figure 3: Bounding Box Drawing

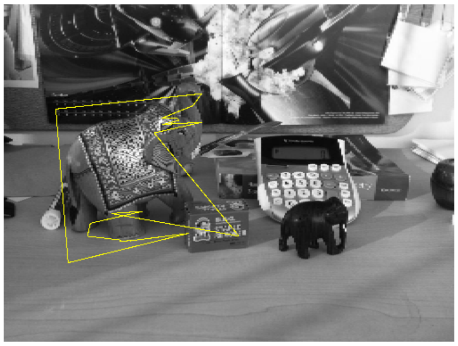

Figure 4: Bounding Box Drawing (Our Approach)

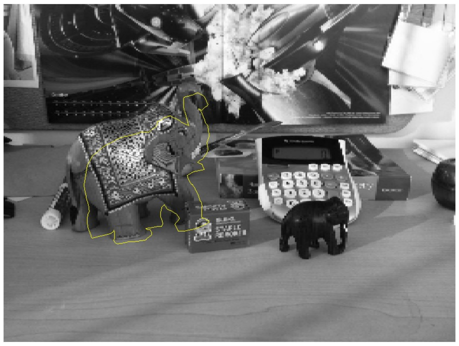

Figure 5: Final result

5 Conclusion

In conclusion, the integration of the Scale Invariant Feature Transform (SIFT) and Random Sample Consensus (RANSAC) algorithms, along with iterative methodological refinements, has been effective in object detection. These techniques have demonstrated efficacy in detecting robust key points and estimating geometric transformations, facilitating accurate feature matching and object localization. To further enhance precision, fine-tuning parameters related to keypoint detection, feature matching, and geometric transformation can be explored. Additionally, the utilization of more specialized techniques holds promise for improving accuracy.

References

[1] Krishna, R. A. N. Jay. *Computer Vision: Foundations and Applications*. Available at: `http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf`. Accessed 4 Apr. 2024.

[2] University of Tubingen Computer Vision Lecture Notes. Available at: `https://drive.google.com/file/d/1J4jA3wAteiChtSAdGgd_2PaWklabBsek/view`. Accessed 4 Apr. 2024.