Basic Computer Vision

Report for Final Project – Marker-based augmented reality

Xuefeng Wei
Institut Polytechnique de Paris
xuefeng.wei@ip-paris.fr

Part I. Theoretical Part

In this section, the focus is on the theory of the relevant algorithms of interest to the authors, which are applied to the project in the subsequent experimental section to compare performance. Firstly, the algorithm related to descriptor grabbing is introduced.

I. SIFT (Scale-invariant feature transform)

A method in computer vision to detect, describe and match local feature points of an image by detecting polar points or feature points in different scale spaces, extracting their position, scale, and rotation invariants, and generating feature descriptors, which are finally used for feature point matching of images. The main steps of the algorithm are: 1. detection of extremes in the scale space 2. precise positioning of keypoints 3. assignment of keypoint principal directions 4. generation of keypoint descriptors. This algorithm is one of the most widely used algorithms.

II. SURF (Speeded-Up Robust Features)

While maintaining the excellent performance characteristics of SIFT operator, this operator solves the disadvantages of high computational complexity and time consuming of SIFT and improves the interest point extraction and its feature vector description, and the computational speed is improved. The main steps of the algorithm are: 1. Construct the Hessian matrix and calculate the eigenvalues α. 2. Construct the Gaussian pyramid. 3. Locate the feature points. 4. Determine the main direction of the feature points. 5. Construct the feature descriptors. Thus, each small region has 4 values, so each feature point is a 16*4=64-dimensional vector, which is less than half compared to sift, which will greatly accelerate the matching speed during feature matching.

III. AKAZE

AKAZE is the accelerated version of KAZE which uses non-linear scale space to find features. Both SIFT and SURF algorithms have an important drawback when constructing the scale space: Gaussian blurring does not preserve object boundary information and smooths to the same level of detail and noise on all scales, affecting the accuracy and uniqueness of localization. For the shortcomings of Gaussian kernel function to construct scale space, bilateral filtering and nonlinear diffusion filtering approaches are proposed to solve the problem. the KAZE algorithm previously proposed by AKAZE authors takes nonlinear diffusion filtering to improve the repeatability and uniqueness compared to SIFT and SURF algorithms.

Next is the theoretical part about descriptor matching

I. Brute Force Matcher

First a feature point is selected in the first image, then a (descriptor) distance test is performed with the feature points in the second image in turn, and finally the feature point with the closest distance is returned.

In this project, the KNNmatch method is used to match feature points, and the method returns the most matched descriptor for the top K features. In order to make the final result better, feature filtering was applied to the project. The distance ratio between the two nearest matches of a considered keypoint is computed and it is a good match when this value is below a threshold, and this value is recommended to be 0.7.
Ultimately, in order to make the matched images translatable to video, we need to calculate the perspective transformation matrix between the matched descriptors, which can be achieved through the function findHomography(), and once the perspective transformation matrix is calculated, the image change can be achieved through the warpPerspective() function. We can specify in this method the RANSAC robust algorithm to try different random subset in order to estimate the homography matrix.

Part II. Experiment Part

In this section, the authors focus on testing the computation time required by several algorithms in different environments and with different objects. The first is descriptor extraction for simple objects in well-lit and dark environments, The test image is shown in Figure 1. The results are shown in Figure2. Then is a comparison of images with rich details with and without lighting, the images are shown in Figure 3 and the results are shown in Figure 4

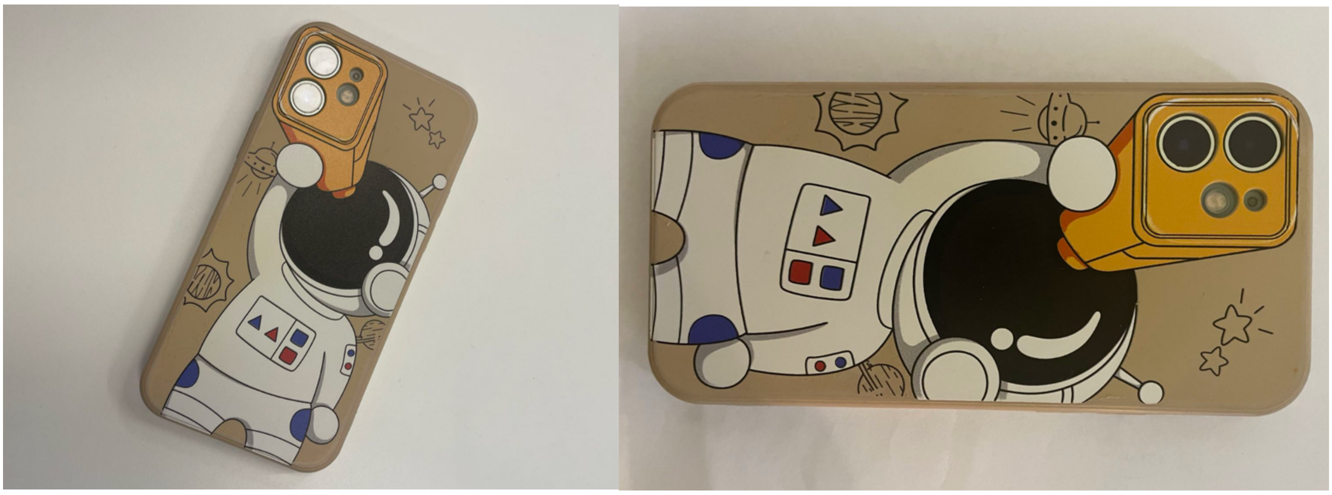

Figure1. Simple Image under lightning and darkness

```
SURF took 0.21047s                SURF took 0.33524s
SIFT took 0.27801s                SIFT took 0.19420s
AKAZE took 0.13738s               AKAZE took 0.12432s

Process finished with exit code 0 Process finished with exit code 0
```

Figure2. Results for the simple image under lightning and darkness

Figure3. Complex Image under lightning and darkness

```
SURF took 0.46109s
SIFT took 0.28062s
AKAZE took 0.13105s

Process finished with exit code 0
```

```
SURF took 0.36911s
SIFT took 0.24674s
AKAZE took 0.15858s

Process finished with exit code 0
```

Figure4. Results for Complex Image under lightning and darkness

From the above experimental comparison of different objects in different environments and backgrounds, AKAZE algorithm can complete the task faster, SURF and SIFT need longer time compared, so I chose AKAZE algorithm in the change project.

PART III. Solution

In this project, the authors' final solution is: the algorithm for capturing image descriptors is AKAZE, and the descriptor matching and filtering algorithms are BFM and KNN, where the filtering of features is performed by setting the ratio to 0.7 and K=2 as hyperparameters to achieve better results. Other hyperparameters are default.

Conclusion

Different objects in different backgrounds and environments have large differences in computational time costs, it is critical to choose a good algorithm to compute image descriptors, feature matching and filtering can bring more stable performance, the authors need to further explore more algorithms for image matching and filtering in the future and test the performance of the algorithm in more environments to get the optimal solution. Finally, a demo video is attached to the project's folder.