Experiment Design in Computer Science
Report II

Rintaro Sato
Student ID: 202120597

June 25, 2021

1 Motivation

My research interests lie in Computer Vision. In the previous Report I, I compared run times of the programs between C++ and Python. I implemented these programs using OpenCV, which is a popular library for those who study Computer Vision. In this Report II, I will dive into the algorithm with C++ and OpenCV more deeply.

I would like to compare the speeds of 3 different algorithm, BRISK, AKAZE and ORB, which are the local feature detectors and descriptors. These algorithm are commonly used in many applications. One best example is that they are exploited to match feature points between 2 images. When looking from a real-time application(e.g., Visual SLAM) point of view, it is inevitable that using faster algorithm to match corresponding feature points between 2 images, especially under the situation where only limited computational resources are available. Fig 1 shows the output image of the program that I have implemented in this report.

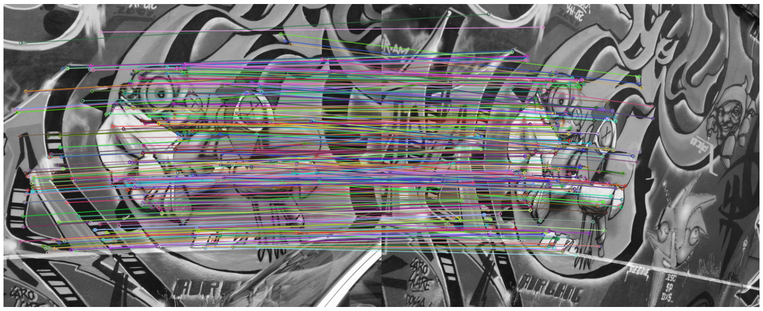

Fig 1: Matching Feature Point between 2 images using AKAZE

2 Research Question

With this motivation, I came up with the following questions.

- Is there any difference in the average run time of the 3 different algorithm(AKAZE, ORB and BRISK) to detect and match feature points, given paired-images?
- In case there is, how significantly is it different against one another?
- What is the best algorithm to use in terms of run time?

In this report, I designed the following experiment to answer these questions. I collected data and performed ANOVA test and multiple comparison. I expect AKAZE, ORB and BRISK could result in more or less fast and similar run time because these algorithm were originally developed for a practical use.

3 Calculation Minimum Sample Size

- desired significance level : $\alpha = 0.05$
- desired power : $1 - \beta = 1 - 0.20 = 0.80$
- minimal interesting effect size: $\delta^* = 5$

Since I set $\alpha = 0.05$, the desired confidence interval size is 1 - $\alpha = 0.95$. I would like to detect whether any two means present differences of magnitude $\delta^* = 5$. Instead of using calculation formulas for the ANOVA only, I performed t-test power calculation for multiple comparison. I used the standard deviation σ of each algorithm for t-test power calculation.

Listing 1: ”Calculation sample size”

```
     One-sample t test power calculation

              n = 92.0752
          delta = 5
             sd = 16.94398
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

To satisfy the desired parameters(α, β, δ^*), required sample size was bigger than $n = 92.0752$. Thus, I decided to obtain $n = 100$ run times in each algorithm.

4 Data Collection

I wrote down the programs using AKAZE, ORB and BRISK. I run each program 100 times and time each program's execution one by one with my shell script. The number of matching feature points calculated by each algorithm was different. Therefore, to compare algorithm fairly, I calculated the run time which spent on 50 inliers of matching feature point in each program. The equation is as follows.

$$run\ time\ per\ 50\ inliers = (50 * run\ time\ of\ program)\ /\ the\ number\ of\ inliers \tag{1}$$

Please note that the unit of run time throughout this experiment is millisecond(10^{-3} seconds). I paid attention to how to conduct the experiment so that the measurements are made under homogeneous conditions (e.g., same compiler, same computer, etc).

The fixed parameter along the entire experiment is threshold($= 2.5$) for identifying inliers with homography check when feature matching. Homography matrix is used for discriminating inliers from outliers. Furthermore, The given values of homography matrix is also constant.

5 Data Visualization

I visualized how run time in each algorithm(group) is distributed with the package *ggplot* in *R*(Fig 2). You can see the run time of AKAZE is widely distributed compaired to ORB and BRISK. It looks like there is no outliers in collected data.

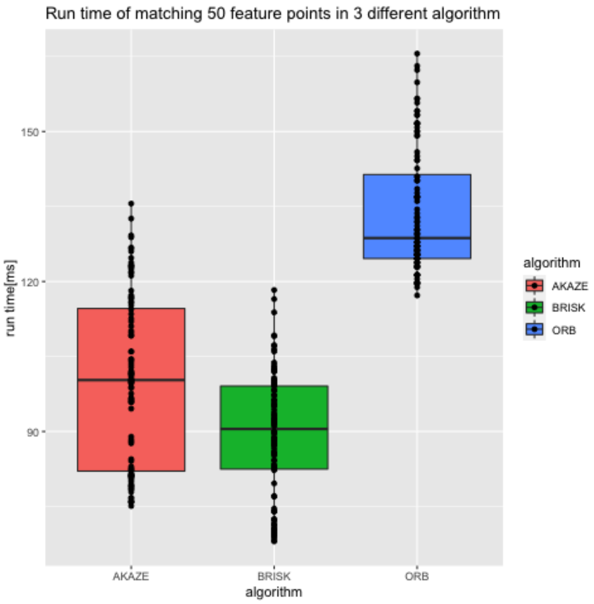

Fig 2: Box plot of run time in AKAZE, BRISK, ORB

6 ANOVA test

The null hypothesis H_0 and the alternative hypothesis H_1 in ANOVA test were defined as follows.

- $H_0 : \tau_i = 0, \forall i \in \{1, 2, 3\}$
- $H_1 : \exists \tau_i \neq 0$

where τ_i represents the effect of the i-th level.

Listing 2: ”ANOVA result”

```
             Df Sum Sq Mean Sq F value Pr(>F)
algorithm     2 106776   53388   270.2 <2e-16 ***
Residuals   297  58675     198
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

P-value was enough small(smaller than $2e^{-16}$) to reject null hypothesis, which means there is at least one level with an effect significantly different from zero.

7 Model Validation

The ANOVA model is based on three assumptions on the behavior of the residuals.

- Normality
- Homoscedasticity, i.e., equality of variances across groups
- Independence

I make sure that each assumption is satisfied by the following statistical tests.

7.1 Normality Assumption

In order to verify the normality assumption, I conducted Shapiro-Wilk test coupled with a normal QQ plot of the residual. In Shapiro-Wilk test, the null hypothesis is the population follows the normal distribution.

Listing 3: ”normality test”

```
Shapiro-Wilk normality test

data:  model$residuals
W = 0.97631, p-value = 7.237e-05
```

P-value was quite small so I ended up rejecting the null hypothesis which the population is normally distributed. However, since I collected relatively large sample($n = 100$), the Central Limit Theorem(CLT) could still guarantee a normally distributed sample mean. Also, as you can see from QQ plot visualization(Fig 3), most of the points fall along a line in the middle of the graph, hence I concluded that normality was verified.

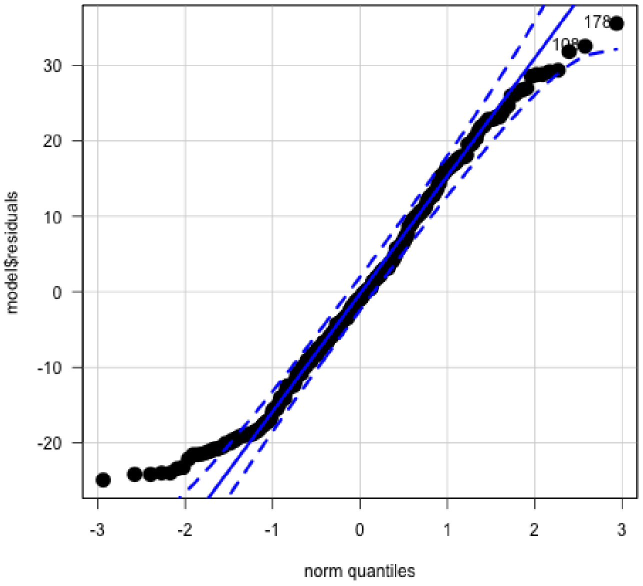

Fig 3: QQ plot of the residuals

7.2 Equality of Variances Assumption

In order to verify homogeneity of variances, I performed Fligner-Killeen (median) test. In Fligner-Killeen test, the null hypothesis is the variances in each of the groups(in this case, algorithm) are the same.

Listing 4: ”homogeneity of variances test”

```
        Fligner-Killeen test of homogeneity of variances

data:  runtime by algorithm
Fligner-Killeen:med chi-squared = 12.733, df = 2, p-value = 0.001718
```

Calculated p-value was around 0.0017. Thus, I rejected the null hypothesis which says the variances in each of the algorithm are the same. However, as you can see from the box plot(Fig 2) and based on the vertical range of residuals(Fig 4), the vairances of the run time in each algorithm are quite simillar. In addition to that, according to the lecture, ANOVA test is known as the robust test to homogeneity of variances.

7.3 Independence Assumption

The independence assumption was guaranteed on the project design phase. I did not execute multiple programs on parallel. Since each program was run one by one, calculated run times do not affect each other.

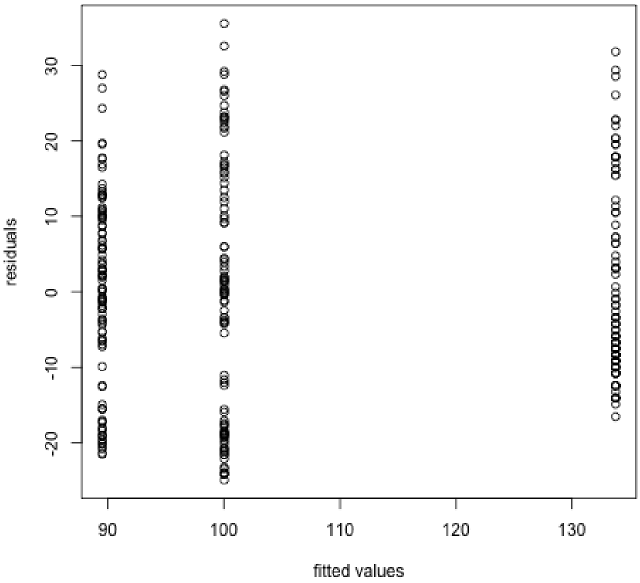

Fig 4: Plot of residuals by fitted values

8 Post-hoc test

For multiple compairson, I performed not only ANOVA test but also Tukey's HSD(honestly significant difference) test. I used Tukey's HSD for post-hoc test to find out which algorithm's means (compared with each other) of run time are different. I also visualized the result(Fig 5).

Listing 5: "Tukey's HSD result"

```
                  diff        lwr       upr p adj
BRISK-AKAZE -10.50856 -15.19079 -5.826326 7e-07
ORB-AKAZE    33.71778  29.03555 38.400014 0e+00
ORB-BRISK    44.22634  39.54411 48.908570 0e+00
```

Based on the difference in BRISK-AKAZE and ORB-BRISK, you can see BRISK is faster compared to other algorithm. You would find that ORB is relatively slower than the other 2 algorithm.

9 Reproductibility

I leave my own environment below for reproductibility. I personally installed OpenCV version 4.5.2 via *Homebrew*, which is the package manager for MacOS. It would be ideal if I could test on other environment(e.g., Windows, Linux, etc) not only MacOS, since the compiler would vary in different OS system(the compiler clang is mainly used on MacOS).

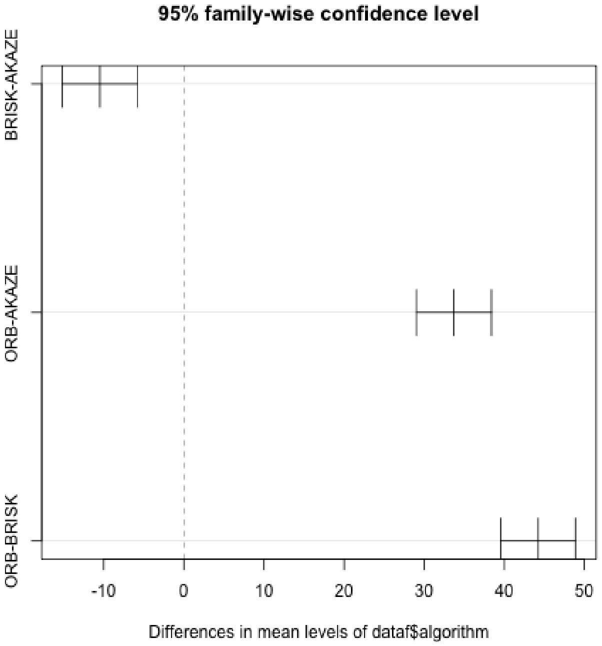

Fig 5: Plot of Tukey's HSD result

Name	Version
OS	MacOS Big Sur version 11.2.2
Processor	1.6GHz Dual-Core Intel Core i5
Memory	4GB 1600 MHz DDR3
Compiler	Apple clang version 11.0.0 (clang-1100.0.33.17)
Library	OpenCV version 4.5.2

Table 1: Development Environment

10 Conclusion

In this report, I compared run times of 3 algorithm, ORB, BRISK and AKAZE. Based on the statistical tests, BRISK is the best approach for feature matching with regard to speed. Please note that this test was conducted only one paired-image. In terms of practical application, I should test various paired-images(e.g., one is normal, the other is rotated image, different scaled image, etc) to investigate the robustness of each algorithm. However, in this case, I need to collect various paired-images for feature matching, which is very time consuming. Conducting an experiment which takes account into the robustness of algorithm, not only run time has left for the future work due to lack of time.