2D Feature Tracking Results Analysis

Ada Lazuli

2024-01-01

Part 1 - Detector Performance

The purpose of this section is to review the performance of the various detectors implemented in the project and draw conclusions around performance. The quantity of **keypoints** generated and **duration** of detection will be the only metrics considered in this section.

The first step is to load the data and overview the comma separated values (CSV) contents.

```r
# load the aggregated csv
df <- read.csv("detector_tests.csv")
glimpse(df)
```

```
## Rows: 60
## Columns: 3
## $ detector  <chr> "orb", "orb", "orb", "orb", "orb", "orb", "orb", "orb", "orb~
## $ keypoints <int> 500, 500, 500, 500, 500, 500, 500, 500, 500, 500, 5063, 4952~
## $ duration  <dbl> 20.360200, 3.467500, 14.660900, 6.151090, 5.922500, 6.365720~
```

The CSV has 60 rows, with three columns per row. The first column details the detector used, the second column records the number of key points generated by the detector for an image, and the third column captures the duration of time needed for keypoint generation.

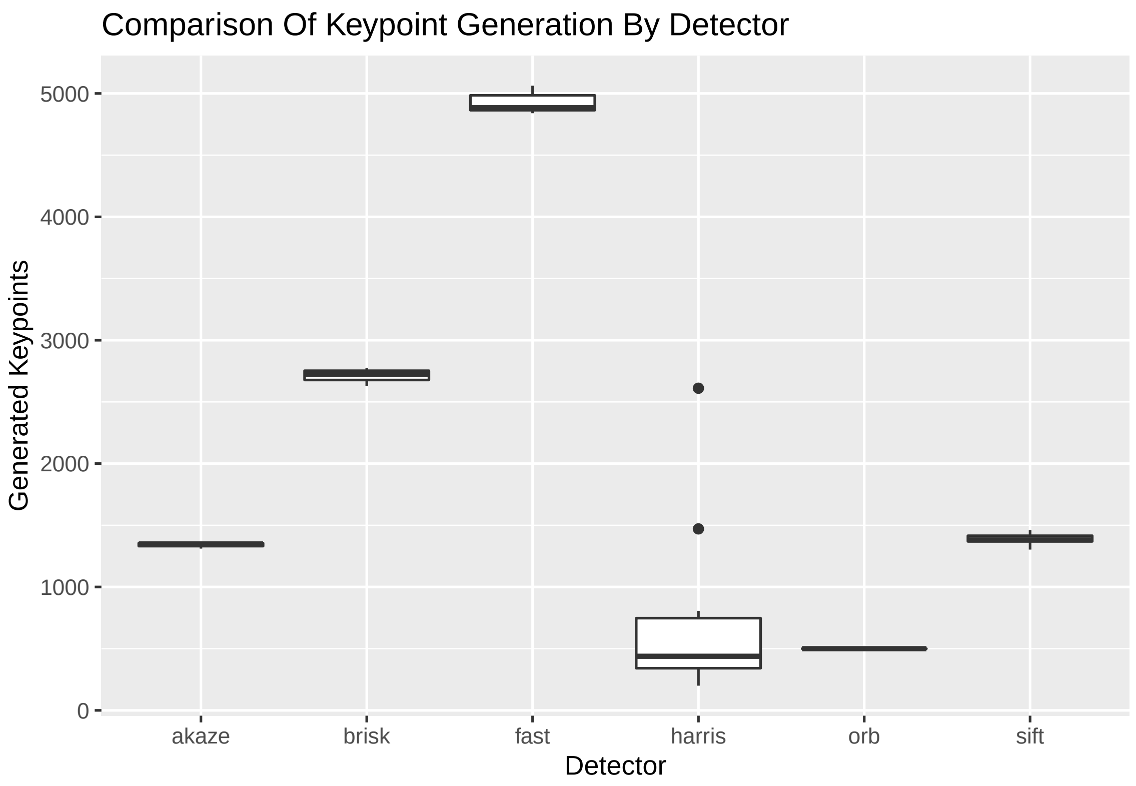
Additionally, the following code block shows that 6 detectors are uniformly represented in the CSV, with each having 10 associated observations.

```r
df %>% count(detector)
```

```
##   detector  n
## 1    akaze 10
## 2    brisk 10
## 3     fast 10
## 4   harris 10
## 5      orb 10
## 6     sift 10
```

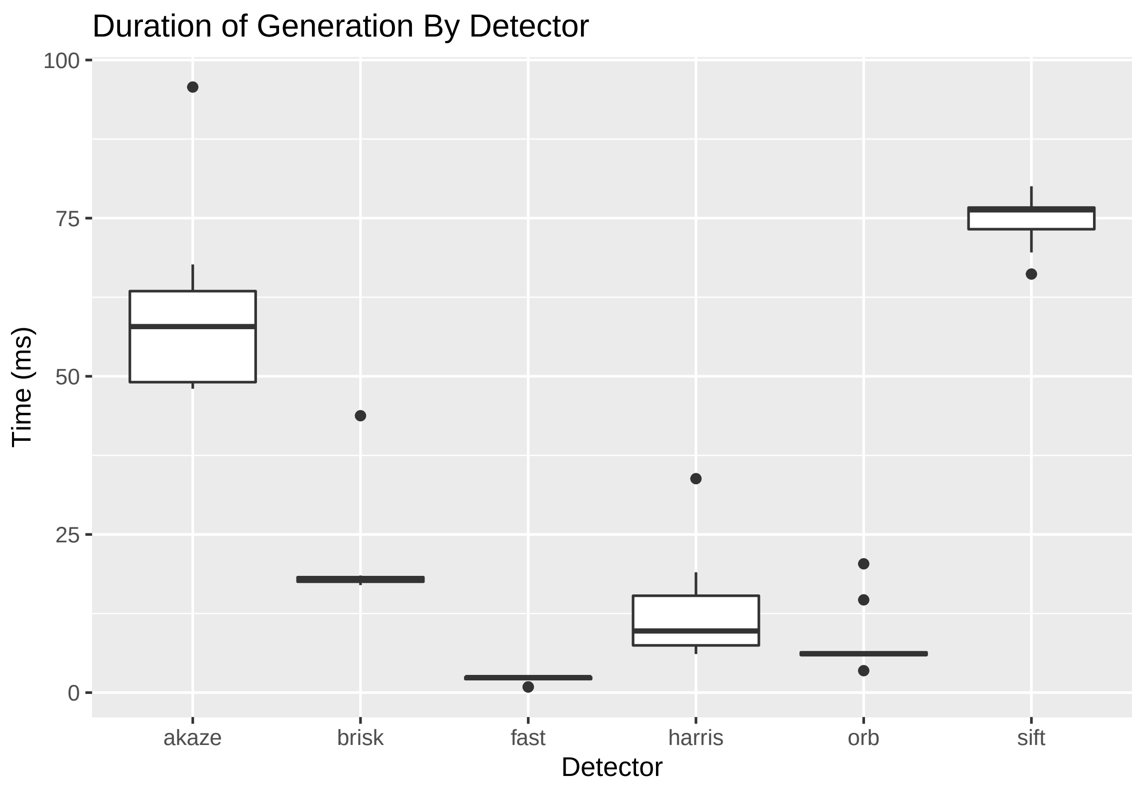
A box and whisker plot was used to review the distribution of keypoint generation by the various detectors for 10 images. The plot shows that most of the detectors were consistent, with *FAST* tending to generate the most keypoints. The *HARRIS* detector was implemented manually and exhibited the highest variability.

```
ggplot(df, aes(x = detector, y = keypoints)) + geom_boxplot() + xlab("Detector") +
  ylab("Generated Keypoints") + ggtitle("Comparison Of Keypoint Generation By Detector")
```



Similarly, a box and whisker plot was used to review the distribution of the time used by the detectors for keypoint generation. The plot showed that the *AKAZE* detector varied the most, *SIFT* generally took the longest, and *FAST* was generally the fastest.

```
ggplot(df, aes(x = detector, y = duration)) + geom_boxplot() + xlab("Detector") +
  ylab("Time (ms)") + ggtitle("Duration of Generation By Detector")
```



Part 2 - Detector Descriptor Performance

A second CSV was created during experimentation that contains details about various runs using different combinations of detectors and descriptors. Additionally, the keypoint matching algorithm and the images used were held constant throughout the experiments.

```
df <- read.csv("detector_descriptor_experiments.csv")
glimpse(df)
```

```
## Rows: 315
## Columns: 6
## $ detector           <chr> "orb", "orb", "orb", "orb", "orb", "orb", "orb", "~
## $ descriptor         <chr> "brisk", "brisk", "brisk", "brisk", "brisk", "bris~
## $ keypoints          <int> 500, 500, 500, 500, 500, 500, 500, 500, 500, 4952,~
## $ detector.duration   <dbl> 3.529930, 7.709430, 3.157630, 2.886860, 2.900540, ~
## $ descriptor.duration <dbl> 0.580986, 0.570177, 0.611642, 0.615170, 0.664459, ~
## $ matches            <int> 82, 93, 95, 103, 101, 115, 119, 118, 116, 420, 431~
```

The CSV contains 315 rows with 6 columns. Each row captures the following details for each detector-descriptor pairing:

1. The **detector** used for keypoint generation
2. The **descriptor** used for keypoint description
3. The number of **keypoints** generated by the detector
4. The amount of **time** used by the detector
5. The amount of **time** used by the descriptor
6. The number of keypoint **matches** between two images

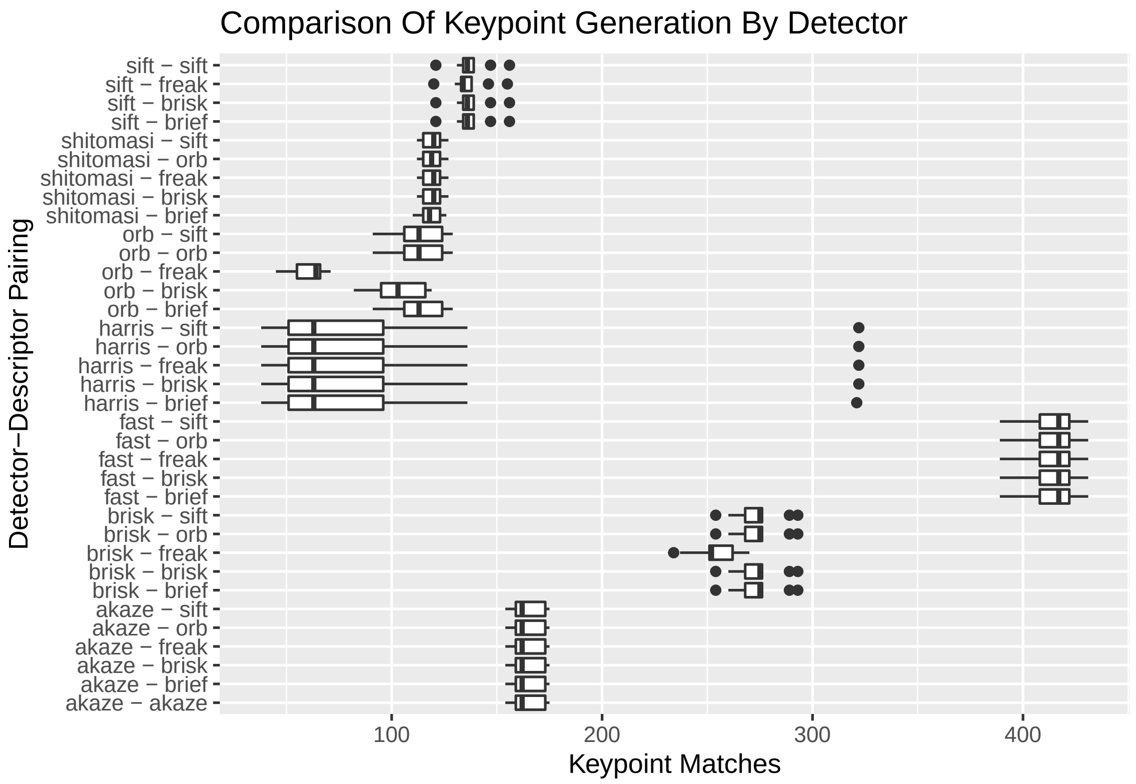
An important aspect to note is that the descriptors are not represented in the data equally. The *AKAZE* descriptor is only compatible with the *AKAZE* detector, hence it is represented the least in the data. Additionally, use of the *ORB* descriptor with the *SIFT* detector resulted in memory overflow errors and could not be recorded, leaving 9 less examples in the data compared to the other descriptors.

```
df %>% count(descriptor)
```

```
##   descriptor  n
## 1      akaze  9
## 2      brief 63
## 3      brisk 63
## 4      freak 63
## 5        orb 54
## 6       sift 63
```

The matches generated by each detector-descriptor pair were analyzed using a box and whisker plot. *HARRIS* and *FAST* had the highest variability and *FAST* had a tendency for the highest number of keypoint matches. Moreover, the plot showed that the detector played the largest role in performance, compared to the descriptor.

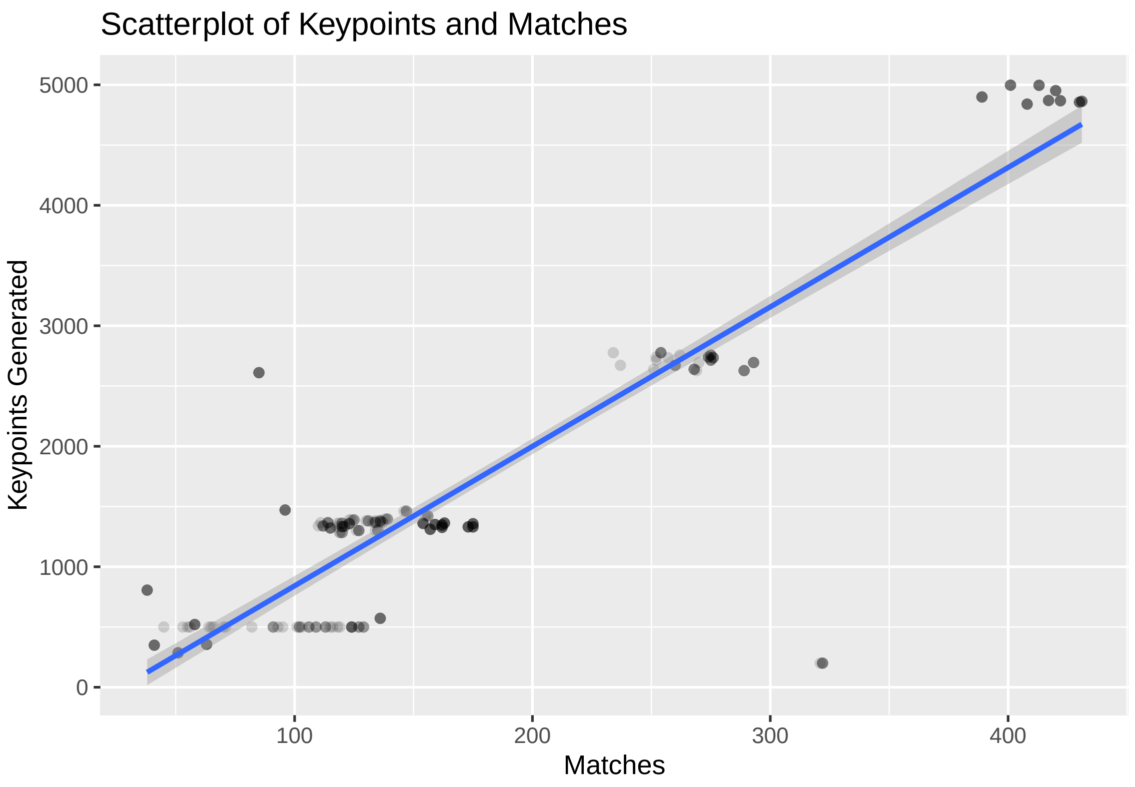
```
df <- df %>% mutate(pair = paste(detector, '-', descriptor))
ggplot(df, aes(x = pair, y = matches)) + geom_boxplot() +
  xlab("Detector-Descriptor Pairing") + ylab("Keypoint Matches") +
  ggtitle("Comparison Of Keypoint Generation By Detector") + coord_flip()
```



A scatter plot was used to further explore the relationship between the number of keypoints and the number of matches. The plot showed that the number of matches increased as the the number of keypoints increased. Therefore, detectors that generate more keypoints, such as *FAST* would naturally have more matches.

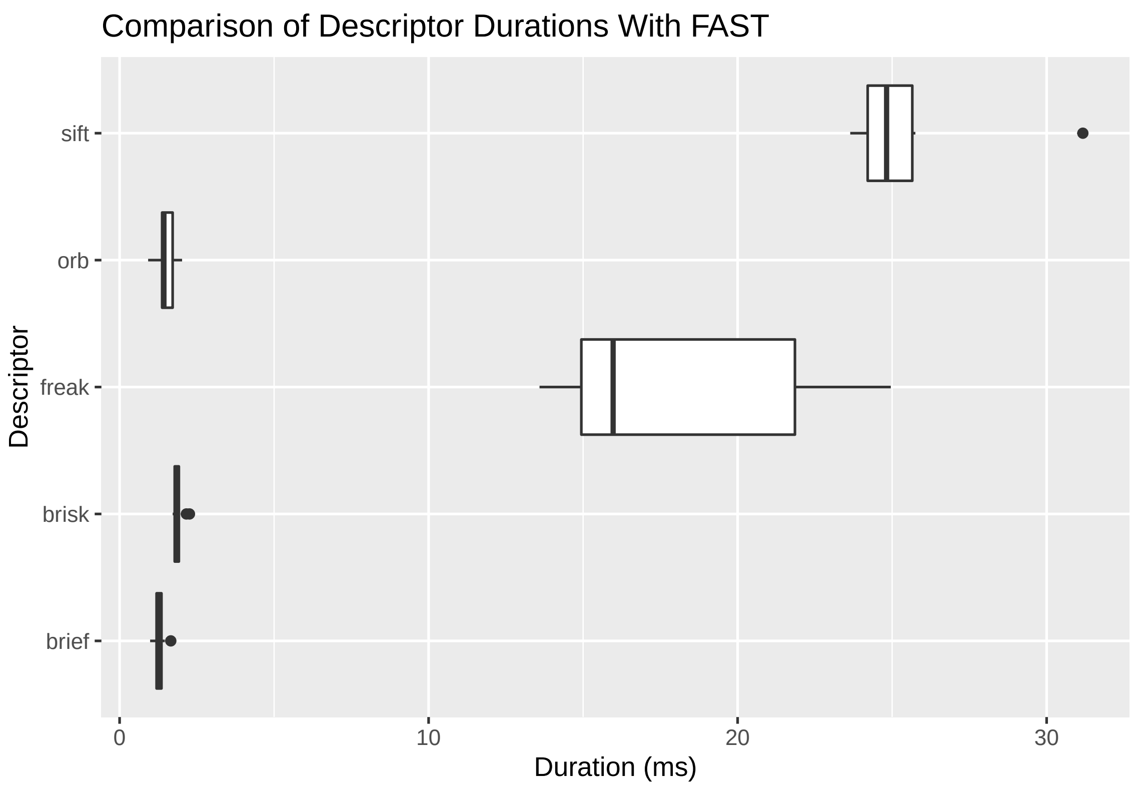
```r
ggplot(df, aes(x = matches, y = keypoints)) + geom_point(alpha = 0.15) +
  xlab("Matches") + ylab("Keypoints Generated") +
  ggtitle("Scatterplot of Keypoints and Matches") + geom_smooth(method='lm')
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Lastly, the duration for each descriptor was reviewed using a box and whisker plot. The plot showed that the *FREAK* descriptor had the highest variability, *SIFT* took the longest, and *BREIF* was the fastest.

```r
ggplot(df %>% filter(detector == 'fast'), aes(x = descriptor, y = descriptor.duration)) +
  geom_boxplot() + xlab("Descriptor") + ylab("Duration (ms)") +
  ggtitle("Comparison of Descriptor Durations With FAST") + coord_flip()
```



Conclusion

After reviewing all of the data collected, the best pairing of detector and descriptor appears to be **FAST** with **BRIEF**. The combination results in the highest keypoint generation, matching, and shortest duration.