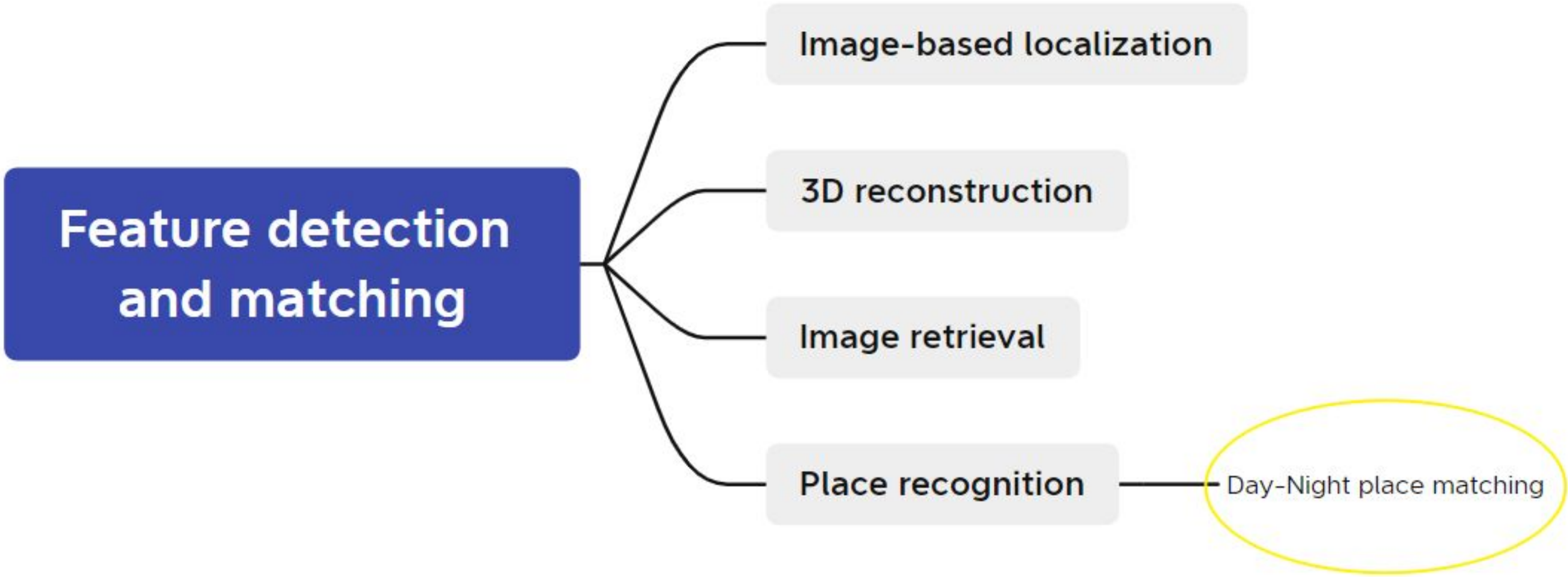
# Nighttime Place Recognition

Yang Wu, Mingjie Zeng, Shuwen Liu

# Outline

- Problem
- Method
- Experiments
- Results

# Problem


Feature detection and matching
Image-based localization
3D reconstruction
Image retrieval
Place recognition
Day-Night place matching

# Problem/2

Main Challenge:

- Feature detectors are severely affected by illumination changes between day and night, including many popular feature detectors such as DoG, Hessian, and the popular SIFT descriptor.
- The number of repeatable keypoints is only an upper bound of match points since the descriptors extracted from the regions might not match[1].

# Experiment 1 - Feature Points

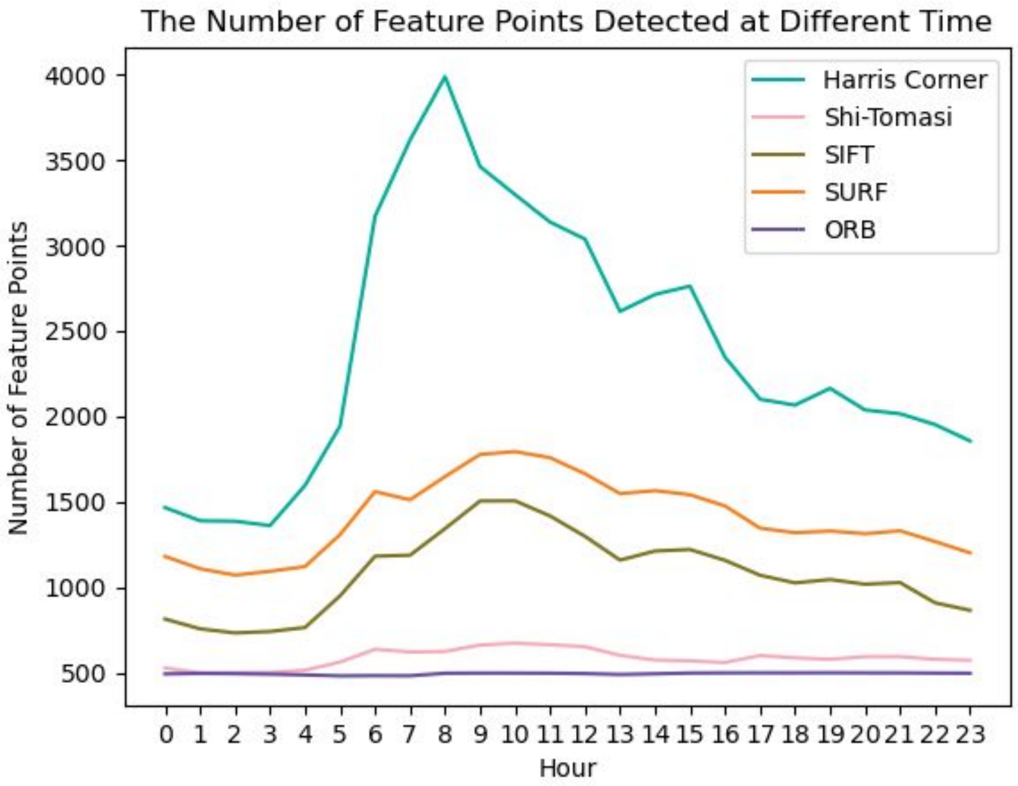
- Finding feature points is an important part of the day-night images matching task

    - Illumination - difference between day&night
    - Number of feature points - upper bound of matching descriptor

# Experiment 1 - Feature Points

- Harris Corner
- Shi-Tomasi

- SIFT
- SURF
- ORB



# Experiment 2 - Pipelines

- SIFT + BF (knnMatch)
- SIFT + FLANN

- SURF + BF (match)
- ORB + BF (match)

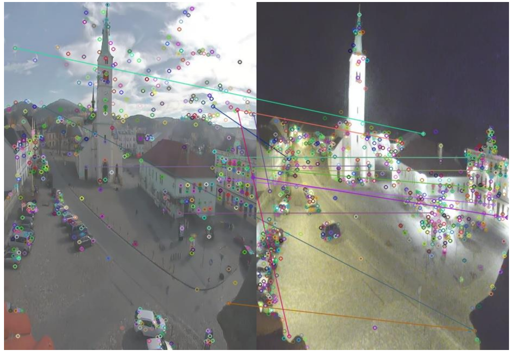
- Corner + SIFT + BF (knnMatch)
- Corner + SIFT + FLANN

# Experiment 2 - Pipelines

Corner_SIFT_BF

Corner_SIFT_FLANN

ORB_BF



SIFT_BF

SIFT_FLANN

SURF_BF

# Method - Pipelines

- Corner + SIFT + BF

  - Keypoints (corner) :
    - goodFeaturesToTrack (return x,y location of the corner)
    - GFTTDetector_create (return KeyPoint object)
  - Descriptor : SIFT
  - Matching : Brute-Force Matcher (knnMatch)
  - Ratio test : by distance

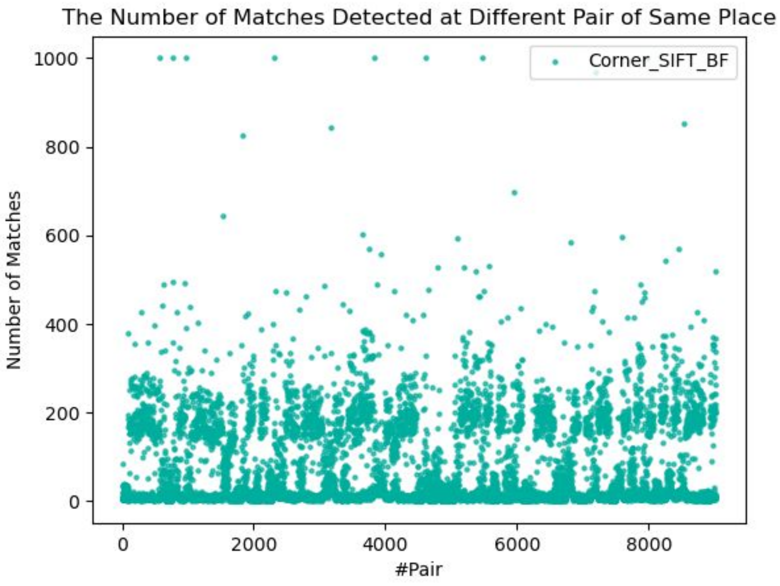
- Preprocessing

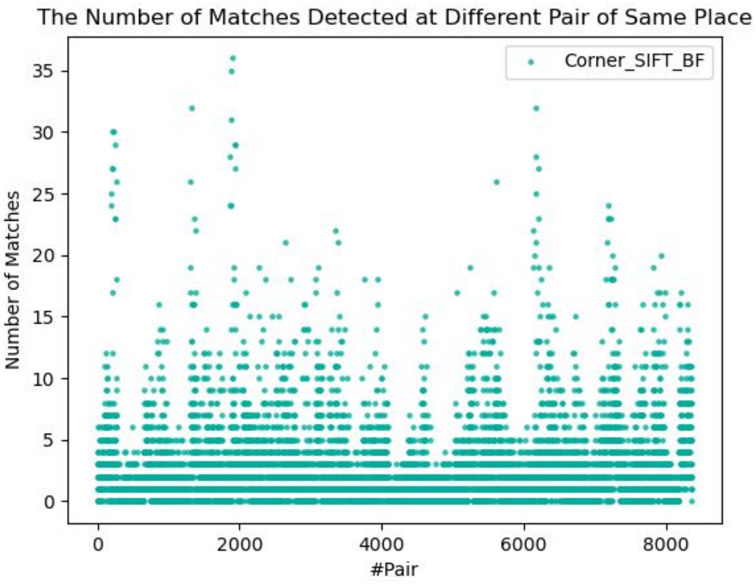
  - Denoise : fastNlMeansDenoising
  - Gamma correction : LUT

# Experiment 3 - How to determine match

- Threshold
    - Number of matches!
- Pipeline of matching
    - Corner (goodFeaturesToTrack) + SIFT + BF
    - Corner (GFTTDetector_create) + SIFT + BF

# Experiment 3.1 - Threshold selection ( GFTTDetector_create )



# Experiment 3.1 - Threshold selection ( GFTTDetector_create )

Good match

```
mean: 79.2372299168975
media: 22.0
percent: 5.0
min: 0.0
max: 1000.0
```

Bad match

```
mean: 2.6788530465949822
media: 2.0
percent: 5.0
min: 0.0
max: 36.0
```

- Use percentile
  - Set percent of good matches: 15% of the matches are smaller than 5
  - Set percent of baad matches: 85% of the matches are smaller than 5

# Experiment 3.1 - Results

- We use the threshold we get test the images in the test folder:

| | Matching Negative | Matching Positive |
|---|---|---|
| Actrual Negtive | 12973 | 1163 |
| Actual Positive | 2853 | 11572 |

| Precision | Recall | Accuracy | F1 |
|---|---|---|---|
| 0.9086768747546132 | 0.8022183708838821 | 0.8593886768670564 | 0.8521354933726067 |

# Experiment 3.2 - Threshold selection ( goodFeaturesToTrack )

In training data, we have images for 15 different places.
We conducted experiments in three groups:
Group1 : Randomly select 500 matched pairs for each place (total paired size 500*15). Then calculate the mean of 500 pairs for each place. For the 15 means, we record the min, mean, 25% quantile, and median as thresholds.

Group2 : Randomly select 1000 matched pairs for each place (total paired size 1000*15). Then calculate the mean of 1000 pairs for each place. For the 15 means, we record the min, mean, 25% quantile, and median as thresholds

Group3 : Randomly select 2000 matched pairs for each place (total paired size 2000*15). Then calculate the mean of 2000 pairs for each place. For the 15 means, we record the min, mean, 25% quantile, and median as thresholds.

# Experiment 3.2 - Threshold selection ( goodFeaturesToTrack )

| | Group 1 (500 per place) | Group 1 (1000 per place) | Group 1 (2000 per place) |
|---|---|---|---|
| min | 57 | 60 | 59 |
| mean | 112 | 112 | 113 |
| 25% quantile | 84 | 85 | 86 |
| median | 103 | 101 | 105 |

# Experiment 3.2 - Testing dataset ( goodFeaturesToTrack )

There are images of two different places in given testing data.

Our testing data:  For each place, we select 250 matched image pairs. Then we randomly select 500 unmatched image pairs as negative samples. In total, we have 500 positive samples and 500 negative samples.

Decision algorithm: In our prediction, if the matched number is equal or larger than the threshold, we predict that pair as “matched”. Otherwise, we regard the pair is “unmatched”.

# Experiment 3.2 - Results (Positive class is “matched” in confusion matrix)

Sorted by decreasing accuracy

| threshold | accuracy | recall | precision | F1 |
|---|---|---|---|---|
| 84 | 0.833 | 0.724 | 0.925831 | 0.812570 |
| 85 | 0.826 | 0.710 | 0.924479 | 0.803167 |
| 86 | 0.825 | 0.700 | 0.933333 | 0.800000 |
| 101 | 0.741 | 0.486 | 0.991837 | 0.652349 |
| 103 | 0.733 | 0.470 | 0.991561 | 0.637720 |
| 60 | 0.729 | 0.904 | 0.669630 | 0.769362 |
| 105 | 0.726 | 0.456 | 0.991304 | 0.624658 |
| 59 | 0.721 | 0.904 | 0.661786 | 0.764159 |
| 57 | 0.706 | 0.906 | 0.647143 | 0.755000 |
| 112 | 0.702 | 0.406 | 0.995098 | 0.576705 |
| 113 | 0.699 | 0.400 | 0.995025 | 0.570613 |

Result: In our experiments, we can get highest accuracy 0.833 by setting threshold as 84. Recall is 0.724. Precision is 0.926. F1 score is 0.813.

# Experiment 3.2 - Results (Positive class is “matched” in confusion matrix)

Sorted by increasing threshold

| threshold | accuracy | recall | precision | F1 |
|---|---|---|---|---|
| 57 | 0.706 | 0.906 | 0.647143 | 0.755000 |
| 59 | 0.721 | 0.904 | 0.661786 | 0.764159 |
| 60 | 0.729 | 0.904 | 0.669630 | 0.769362 |
| 84 | 0.833 | 0.724 | 0.925831 | 0.812570 |
| 85 | 0.826 | 0.710 | 0.924479 | 0.803167 |
| 86 | 0.825 | 0.700 | 0.933333 | 0.800000 |
| 101 | 0.741 | 0.486 | 0.991837 | 0.652349 |
| 103 | 0.733 | 0.470 | 0.991561 | 0.637720 |
| 105 | 0.726 | 0.456 | 0.991304 | 0.624658 |
| 112 | 0.702 | 0.406 | 0.995098 | 0.576705 |
| 113 | 0.699 | 0.400 | 0.995025 | 0.570613 |

Analysis:

1. With the threshold increases, we can see that accuracy increases first and then decreases. We get our best result when threshold is 84. So we think the appropriate threshold should be in range (60, 85).
2. With the threshold increases, the recall keeps decreasing. It means that we will more easily to predict the matched images as “matched” when setting a relatively small threshold.
3. On the other hand, the precision increases along with increasing threshold. It means that we can get larger percent of matched images in our predicted “matched”.

# Experiment 3.2 - Future works

- We will conduct more experiments for the threshold in range (60, 85).
- For each group, we will try to use median instead of using mean. Maybe that’s a good way to accelerate the experiments, and get more accurate result.

# Reference

- [1] Zhou, Hao, Torsten Sattler, and David W. Jacobs. "Evaluating local features for day-night matching." European Conference on Computer Vision. Springer, Cham, 2016.

# Q & A Time

Thank you!