Project 8 Practice and Application of Image Filtering

Due : 23:59,June 9

Responsible TA: Shuhang Zhang
e-mail: zhangshh2024@shanghaitech.edu.cn

1 Q1: License Plate Localization

1.1 Background

License plate localization is a critical step in automatic license plate recognition (ALPR) systems. Effective pre-processing—particularly image filtering—is essential for improving the robustness and precision of the localization algorithm. Understanding how to choose and combine appropriate filters is therefore an important skill for practitioners in digital image processing.

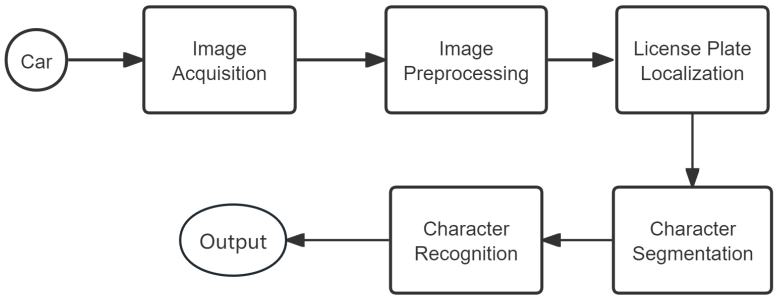

Figure 1: Automatic license plate recognition system pipeline

(a) (b)

Figure 2: License plate localization

1.2 Task

Given a small set of images, you are required to implement a simple license plate localization program.

1. **Pre-processing with Image Filtering:** Design a multi-stage pre-processing pipeline for license plate localization. You should select and combine at least two different image filtering or enhancement methods (e.g., median/mean/Gaussian/bilateral filtering, histogram equalization, morphological operations).
2. **License Plate Localization:** After filtering, apply suitable techniques to locate the license plate region. This may involve connected component analysis or contour extraction.
3. **Visualization:** Show the cropped plate image after localization as fig2

1.3 Discussion

You need to answer these following question in your report:

1. In the pre-processing step, you will design an image filter pipeline. Explain your choices for each filter: What issue does it address? Why is it chosen for license plate localization?
2. Briefly describe the process of the localization step.

You need to show the results:

1. Show the intermediate results of your filters in the pre-processing step (show the example of one image is enough).
2. Show the cropped plate images after localization .

2 Q2: Target Tracking with Kalman Filter

2.1 Background

Real-time tracking of moving objects (such as pedestrians, vehicles, drones, etc.) is a fundamental and critical task in computer vision, autonomous driving, robotics, and other fields. Kalman Filter is a recursive filtering algorithm, which is suitable for state estimation of linear dynamic systems. It can fuse the system motion model and sensor measurements, suppress noise and smooth the trajectory.

(a) (b)

Figure 3: Target Tracking

2.2 Task

You will implement an algorithm of Target Tracking with Kalman Filter. Given a video (a sequence of images), there's a person walking in it. The positions of the person (x, y) are given by Mean-shift algorithm (already implemented), please treat the positions as the '**observed values**' of Kalman filter, and try to predict the '**true values**' of the future positions.

You should take the positions given by Mean Shift as the observed values, implement the Kalman filter to predict the next position of the person[1]. We have provided 2 Matlab Scripts :*extract.m* for extracting object and *kalman_script.m* for target tracking with Kalman filter. You should implement Kalman filter in *kalman_script.m*. For convenience, we have implemented the initialization step of Kalman filter, you are required to implement the updating step. General steps of Kalman filter includes:

1. **State Prediction:** predicting the current state based on the previous estimated state $\hat{x}_{k-1}$, the control input u_{k-1}, and the transition matrix F. B is the control input model.
$$\hat{x}_k^- = F\hat{x}_{k-1} + Bu_{k-1} \tag{1}$$
2. **Error Covariance Prediction:** predicting the error covariance of the state estimate. P_{k-1} is the previous estimate error covariance, and Q is the process noise covariance.
$$P_k^- = FP_{k-1}F^T + Q \tag{2}$$
3. **Kalman Gain:** determining how much the predictions should be corrected based on the new measurement. H is the observation matrix, and R is the measurement noise covariance.
$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{3}$$
4. **State Update:** updating the predicted state $\hat{x}_k^-$ using the actual measurement z_k and the Kalman Gain K_k.
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4}$$
5. **Error Covariance Update:** updating the error covariance of the state estimate after incorporating the measurement.
$$P_k = (I - K_k H)P_k^- \tag{5}$$

The evaluation metric for this task is **position tracking error**

$$E_{position} = \frac{\sqrt{(x' - x)^2 + (y' - y)^2}}{\sqrt{x_c^2 + y_c^2}} \tag{6}$$

where (x, y) is the real position (given by Mean Shift), (x', y') is the predicted position (given by Kalman filter), and (x_c, y_c) is the center of the image.

2.3 Discussion

Show the figure and numerical result of position tracking error . Discuss your results, compare the algorithms in performance, latency, pros and cons, etc.

3 Bonus

In Q2, we have provided a simple initialization step of Kalman filter . In practice , these initialization matrices are important hypermeters in Kalman filter , which also affect the prediction performance. In the bonus part, you are required to customized these initialization parameters according to the video sequence and give your reason. (For example: you could estimate image noise (R) from stationary frames). Compare your results after customized initialization with previous results.

4 Submission Package

Submit a zip file includes:

1. A pdf report
2. A folder of source code with a README.md
3. A folder of images. They should be able to reproduce the numbers in your report.

Note that any missing of above items will lead to at least 20% points deduction.

5 Grading Rules

1. Running the code successfully. (40pts)
2. The performance of your algorithm, evaluating by position tracking error. (20pts)
3. The report of this project. (40pts)
4. Bonus task. (20pts)

References

[1] Dan Yu, Wei Wei, and YuanHui Zhang. Dynamic target tracking with kalman filter as predictor. *Opto-Electron. Eng*, 36:52–56, 2009.