Automatic Virtual Camera View Generation for Lecture Videos

Derek Pang DCYPANG@STANFORD.EDU
Sameer Madan SAMEER27@STANFORD.EDU
Serene Kosaraju SERENEK@STANFORD.EDU
Tarun Vir Singh TARUNVIRSINGH@STANFORD.EDU
CS 229 Project Report, Fall 2010

Abstract

ClassX, an interactive online viewing system developed at Stanford University, currently offers automatic tracking of the lecturer to create a virtual camera view for students. However, a tracking-based camera view does not mimic a human operator naturally. To address this issue, we propose a new learning algorithm to automatically generate a professional virtual camera view by learning the behavior of a human camera operator. Our algorithm is based on a human saliency model which predicts the viewer's center-of-attention and additional features which drive the cinematic decision-making process of the system. Experimental result reveals our system can provide an effective camera view that is almost indistinguishable from a human-generated camera view.

1. Introduction

Growing Internet access, increasing network throughput, improving computer hardware and enhanced video compression are providing a boost to inexpensive online delivery of lecture videos. However, most lecture capture systems depend on human camera operators as well as manual work of post- production and online video publishing, thus resulting in expensive solutions.

The *ClassX* lecture capturing system (Mavlankar et al., 2010) developed at Stanford University offers a solution to these problems by offering an interactive lecture viewing system. Each viewer can interactively choose an arbitrary region-of-interest (RoI) for viewing on the video. Apart from allowing the user to control pan/tilt/zoom, ClassX offers a *tracking* mode. The RoI video streamed in *tracking* mode is generated through automatic cropping and mimics a human camera operator, similar to the approach in (Nagai, 2009). This approach differs from prior work employing a camera that physically moves to track the lecturer or multiple cameras that cover different regions (Rui et al., 2004; Bianchi, 2004; Zhang et al., 2008). However,current *ClassX*'s tracking algorithm always follows the lecturer, and does not produce a natural camera view that is relevant to the viewers.

Heck, Wallick and Gleicher (Heck et al., 2007) have proposed a system called *Virtual Videography* for lecture recordings. It allows a computer system to employ the art of videography and mimic videgrapher-produced video, while unobstrusively recording a video scene. Similar to *ClassX* , the system uses unattended, stationary video cameras to record the event and automatically produce a professional video sequence by simulating various aspects of a production crews. Motivated by the *Virtual Videography* system, we propose a supervised-learning based algorithm to predict a professional-produced camera view for lecture videos. Different from Heck's system, our learning algorithm make use of a human saliency model which predicts the viewer's center-of-attention and other features that motivates cinematic decisions in camera operation.

2. System overview

In this project, we consider a simple virtual camera with two degrees of freedom, horizontal position and a bi-level zoom level. The orientation of the camera is ignored because the position of the virtual camera is flexible to directly capture the scene along a horizontal axis. Therefore, the virtual camera will not capture the scene at an angle and will not introduce additional geometric distortion. Furthermore, since the trajectory of the lecturer is often constrained in the horizontal direction, we can avoid moving the camera vertically. Our camera also only supports a bi-level zoom factor for simplicity.

A system overview is shown in Figure 1. Our system will first obtain an input video recorded from a static HD camera, as well as the locations of any writing boards in the scene. Then, our system will compute the relevant features, such as center-of-attention, lecturer position, to predict the expected camera operation and trajectory based on supervised learning techniques. Finally, we will perform a consistency check and apply Kalman filter to our result to reduce irregularities and noises in our prediction. After the camera view is decided, we can crop the relevant region of the video and encode the video content for delivery.

3. Feature extractions

To successfully operate an automatic camera, the system must learn where to capture the most relevant information to the viewer and how to make certain cinematic decisions, such as panning and zooming. In this section, we propose a set of features that are easily extractable from a lecture video and other system inputs for making robust camera view prediction.

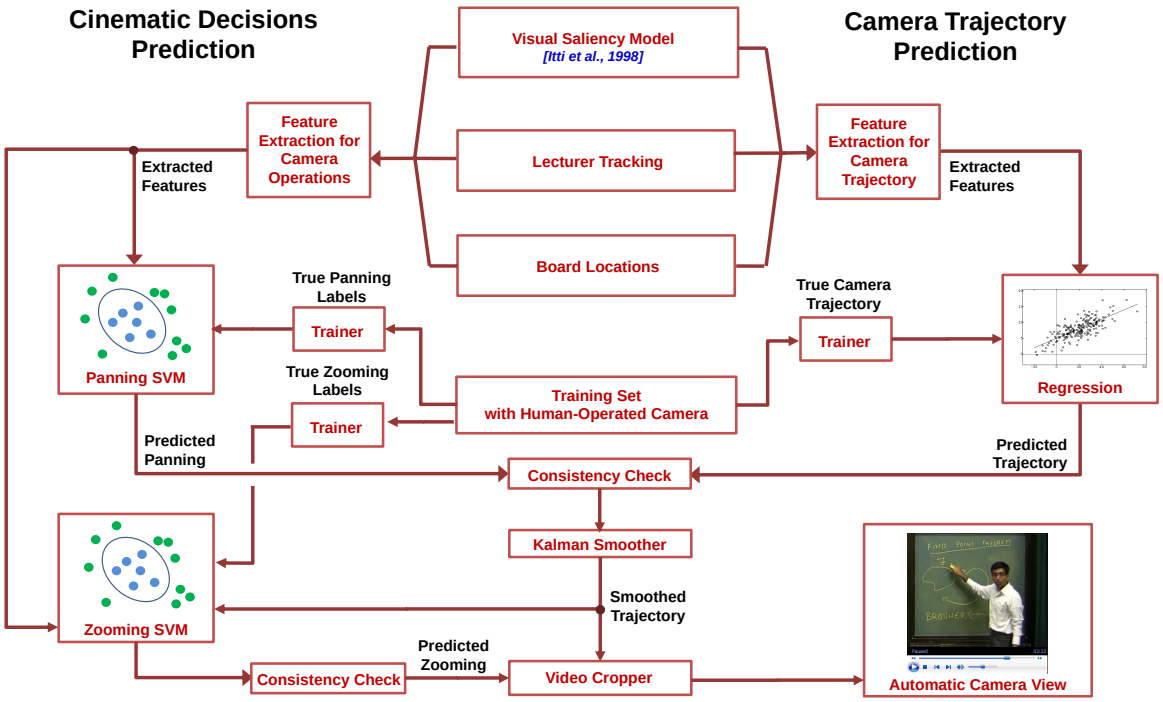

Figure 1. System overview

3.1. Lecturer tracking

One obvious feature to drive the camera's attention for lecture capture is by tracking the location of the lecturer. By having a set of target templates, we utilize conventional background subtraction (Piccardi, 2004) and template matching techniques to track our target. In this case, the position and velocity of the lecturer are the features used for prediction.

3.2. Human saliency model for visual attention

One of the important rules in videography is to direct the viewer's attention toward what is important (Katz, 1991). Without specifying a tracking target, a more generic approach is to predict general viewer's visual attention on a video sequence. A biologically-plausible computational model of human visual attention has been proposed by Itti et al.(Itti et al., 1998). In this project, we simplified Itti's model by only computing motion and color contrast features. Skin color detection is also added to compute the skin color response on a video frame because the presence of people often dominates viewer's center-of-attention in a lecture video. After all features are computed for each frame, we normalize each feature response between 0 and 1 and combine them linearly to form a *saliency map*. A binary threshold is then applied on the *saliency map* to eliminate regions with weak salient response. Finally, the center-of-attention is calculated by taking the median of the locations of the high-response salient regions.

3.3. Features for cinematic decisions

The main challenge to automatically producing effective lecture videos is synthesizing the decision making process of human videographers. Lecturer tracking and human saliency offers features that suggest where the center of "action" in a given scene is. However, they do not consider certain features that influence an operator's cinematic decisions in panning and zooming. For example, a lecturer might abruptly walk away from and into a particular board that he/she is discussing. In this situation, the camera should be focused on the board to avoid abrupt camera movement. Therefore, features from tracking will fail to capture this cinematic decision. Thus, we investigated the following features for making a better panning decision by avoiding the aforementioned problems:

- *The distance between the current center-of-attention and its moving average over multiple intervals* This measurement suppresses abrupt panning changes and promotes long-distance camera panning.
- *Shifts in center-of-attention.* A fast shift in center-of-attention implies panning should be invoked.
- *Long-term turnaround time of a lecturer.* A binary feature that denotes whether the lecturer has returned to a particular writing board given a threshold time.
- *The distance between the center-of-attention and the center of the nearest board.* When the distance is high, camera has a high tendency to pan way from the board

The features for deciding zooming decision also use an analogous set of features:

- *The distance between the lecturer's position and the center of the nearest board.* When the distance is low, the lecturer is likely to be staying in the same board and therefore we should perform a zoom-in operation.

- *The distance between the predicted trajectory and the center of the nearest board.* Since panning and trajectory prediction are independent of zooming operation, we could make use of the trajectory generated by our learning algorithm. The predicted trajectory also provides a more stable trajectory compared to the center-of-attention and the lecturer's trajectories.
- *Distance of near-future center-of-attention movement.* High distance movement indicates the viewing target is likely to be moving and the camera should be zoomed out.

In addition, we have also studied the effectiveness of the following features for both panning and zooming features:

- *Face detection.* When a lecturer is writing and facing toward the board, the camera should zoom into the board for a better view of his writing. Therefore, we utilized a face detection algorithm proposed by (Viola & Jones, 2004) to detect whether he is facing the board or the camera.
- *Eigen-gestures.* Similar to Eigenface, we apply the principal component analysis (PCA) analysis in our training data and generate a set of eigen-images for recognizing the gestures that are often occurred during panning.

Please note that the same set of features can be applied to different classroom settings and can enable a learning algorithm adaptive to different environment.

4. Cinematic Decision Prediction

4.1. Support Vector Machines for cinematic decisions

The cinematic decisions, specifically panning and bi-level zooming, considered in our virtual camera operations can be generalized as as classification problem. One approach to solving classification problem is support vector machine (SVM). Using our feature set as discussed in Sec. 3.3, a panning SVM and a zooming SVM can output a bi-level decision for the panning and the zooming operation respectively on each video frame. To exploit the possible co-occurence relationships between each feature, we employ a Gaussian kernel with L2-norm distance, where the parameters are selected by a five-fold cross validation. We can also substitute the Gaussian kernel with a linear, polynomial, or a sigmoid kernel, but these kernels do not yield the optimal performance in our application. A multi-level SVM can be used for predicting multi-level zooming operations if necessary.

4.2. Consistency check

The sequence output of our SVM decisions is often noisy and might affect the user's viewing experience. Thus, we perform a consistency check to remove thin troughs and flatten out spikes observed in our decision sequence. The implications of such an action is very different in the case of panning and zooming. In the panning case, it is always better to favor a panning decision for preventing viewing target moving out of the viewing regions. Therefore, all parameters in panning SVM is optimized for higher recall rate. Conversely, in the zooming case, it is always preferable to perform a zoom-out operation to ensure all vital information is inside the viewing region. Therefore, we optimize the precision rate for the zooming SVM.

To rectify our outputs, we use two parameters which define the minimum allowable duration of a spike and a trough to eliminate sporadic changes in camera operations. To optimize the recall rate of panning decisions, we choose a high minimum allowable duration for trough and a low value for spike. In the zooming case, we choose a low allowable duration for trough and a high value for spikes to obtain a higher precision value. This rectification process ensures the flow of our camera operation is consistent.

5. Camera Trajectory Prediction

A professional lecture capturing system must provide a stabilized camera view that captures the user's region of interest. To ensure the same viewing experience in our system, we predict the trajectory of our system's camera view based on the features that indicate both the center of visual attention on a scene and the predicted camera operation that controls the actual camera trajectory.

Using the present and future information of the lecturer's position and the center-of-attention, we formulate a linear regression model to predict where the camera would most likely go to. However, this prediction does not totally reflect the operator's decision on panning in a lecture-setting as discussed earlier.

To accurately predict a natural camera trajectory, we combine our regression prediction with our camera panning prediction discussed in Sec. 4. When a non-panning decision was made, the predicted camera trajectory would move to the nearest board and stop until a panning decision was outputted. When a panning decision arrived, the trajectory would follow the output of the regression model for panning. After the camera trajectory is generated, a Kalman filter can also be applied to reduce the irregularities and noises in our predicted trajectory. The parameters of the Kalman filter can be estimated through Expectation-Maxizmation (EM) algorithm. Please refer to Appendix in our report for a full derivation of the Kalman filter.

6. Experimental Result

6.1. Experiment setup

In our experiment, we used two board-writing style lectures. Each lecture is approximately 30 minutes long and is composed of 50,000 video frames. For feature selection, parameter estimation, and kernel selection, we used a five-fold cross validation by dividing each video lecture into five continuous segments. For our performance evaluation, we trained our model on a single video lecture and apply the prediction on the other lecture video. We then switched the role of both sets and obtained an average performance of both testing sets. To extract our ground-truth data, we asked a human spectator to view the content and determine the appropriate camera viewpoints using a mouse and a keyboard through a software interface on a PC.

Table 1. SVM performance

	Panning	Zooming
Precision	58%	82%
Recall	69%	64%
Accuracy	72%	71%

Table 2. MSE performance comparison

Method used	MSE	% MSE reduction
Lecturer tracking	28.16×10^7	-
Saliency-based center-of-attention prediction	15.98×10^7	43%
Proposed method without consistency check	12.12×10^7	56%
Proposed method with consistency check	11.79×10^7	58%
Proposed method with ideal panning labels	9.11×10^7	68%

6.2. Feature selections

During our feature selection process, we found that face detection and eigen-gesture features actually harm our prediction performance because they actually do not have a strong correlation with panning and zooming decisions. For example, the lecturer often faces forward in both non-panning and panning situations. Furthermore, if we compare the eigen-gestures of the lecturer in both of these situations, the eigen-gestures exhibit similar characteristics. Therefore, we removed these features from our model.

6.3. Cinematic decision predictions

Table 1 shows the performance of our panning SVM and zooming SVM prediction respectively. As discussed in Sec. 4, the panning SVM is optimized for recall and the zooming SVM is optimized for precision. In general, we found that panning prediction is a harder problem because panning also depends on the contextual information of the lecture that our system cannot capture.

6.4. Camera trajectory prediction

Table 2 reports the mean square error (MSE) between the ground-truth trajectory and the trajectory generated by different methods. We compare each method to the tracking-based trajectory used in the *ClassX* system. If we only use the center-of-attention as our camera trajectory, we can reduce the MSE by 43% compared to tracking-based approach. Using our proposed method, we observed a even larger reduction of 56% and 58% without consistency check and with consistency check respectively. We also show an ideal lower bound for trajectory prediction to be a reduction of 68% if we can perfectly predict the panning decisions.

Fig.2(a) illustrates sample trajectories generated by different methods. Compared to the tracking-based trajectory, both the center-of-attention and our method provided a more stabilized trajectory. Furthermore, our method also offers a better prediction in the camera center and eliminates some of the unnecessary panning. Fig.2(b-e) shows the ground-truth and the predicted labels for panning and zooming decisions. In general, our algorithm correctly predicts large camera panning and provides stable zooming predictions. For qualitative result, we have recorded an example video clip to compare between the ground truth and our proposed method. The video clip can be viewed via http://mars3.stanford.edu/dcypang/cs229/Report/.

Table 3. System complexity

Operation	Implementation	Execution time per frame
Lecturer tracking	C/C++	8 ms
Visual saliency	C	1 ms
Prediction	MATLAB	3 ms
Other	MATLAB	1 ms
Total		13 ms

6.5. System complexity

Table 3 shows the performance of our system tested on the Stanford's *Corn* server with 8-core Opteron 2384 and 32GB RAM. We also developed a low-complexity saliency extraction and lecturer tracking using C/C++. Other operations, such as feature extraction, regression, and SVM prediction, are implemented using Matlab. On average, our proposed system only requires a processing time of 13 ms per frame.

7. Conclusion

In this report, we proposed a supervised learning algorithm that takes advantage of conventional tracking techniques and the state-of-the-art human saliency model to automatically operate a virtual camera for lecture video capture. Our system has resolved the restrictions imposed in the current *ClassX* system. Experimental results reveals our system can provide a indistinguishable camera view from human-generated view and can offer a low-cost, low-complexity solution to the automatic camera view generation for online lecture viewing. For future work, we will extend our model evaluation to multiple classes with different classroom settings. A subjective test should also be conducted for evaluating the quality of the automatically-generated camera views.

8. Acknowledgement

We would like to acknowledge the support of the ClassX team, namely Sherif Halawa, Ngai-Man Cheung, Mina Makar and Prof. Bernd Girod, for offering their helps and providing the lecture video contents for our experiment. We thank Huizhong Chen and Fan Wang for providing the face detection software. Lastly, we would like to thank Prof. Andrew Ng and the teaching assistants for giving us a learning opportunity to apply machine learning in this project.

References

Bianchi, Michael. Automatic video production of lectures using an intelligent and aware environment. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, MUM '04, pp. 117–123, New

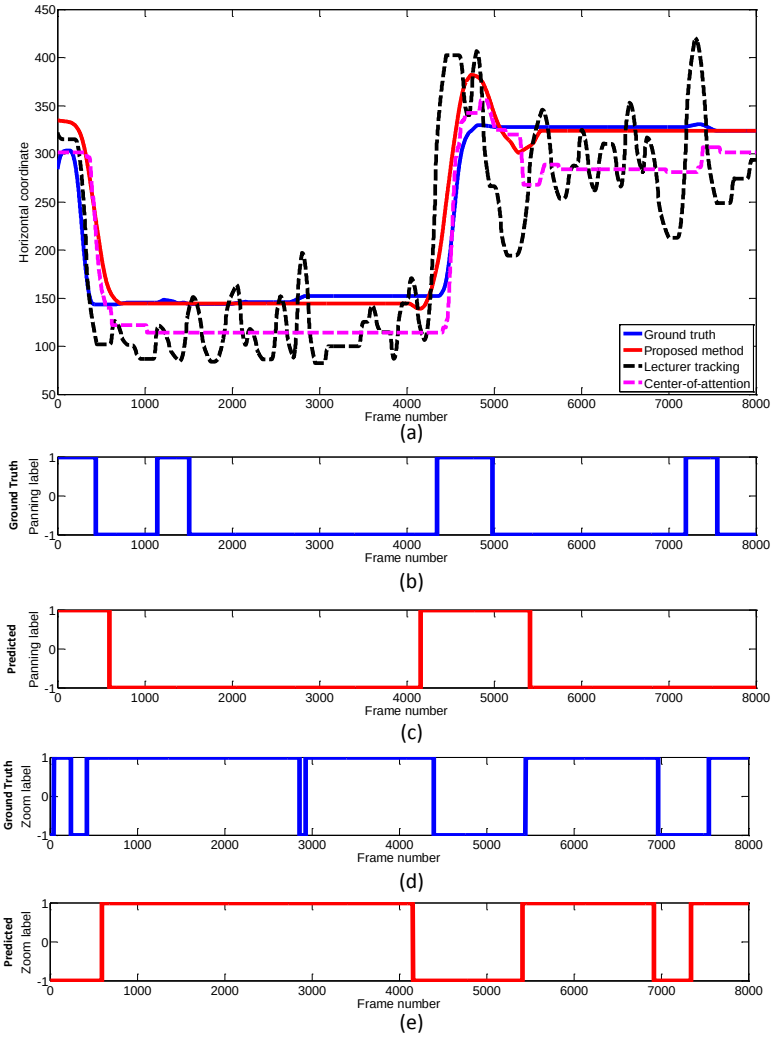

Figure 2. (a) Comparison of trajectories given by the ground-truth, lecturer tracking, center-of-attention and our proposed method. (b) Ground-truth panning labels (c) Predicted panning labels (d) Ground-truth zooming labels (e) Predicted zooming labels

York, NY, USA, 2004. ACM.

Heck, Rachel, Wallick, Michael, and Gleicher, Michael. Virtual videography. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3, February 2007.

Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254 –1259, November 1998.

Katz, Steven. Film Directing Shot by Shot. 1991.

Koch, C. and Ullman, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Human neurobiology*, 4(4):219–227, 1985.

Mavlankar, A., Agrawal, P., Pang, D., Halawa, S., Cheung, N., and Girod, B. An interactive region-of-interest video streaming system for online lecture viewing. *Special Session on Advanced Interactive Multimedia Streaming, Proc. of 18th International Packet Video Workshop*, 2010.

Nagai, Takayuki. Automated lecture recording system with avchd camcorder and microserver. In *Proceedings of the 37th annual ACM SIGUCCS fall conference*, SIGUCCS '09, pp. 47–54, New York, NY, USA, 2009. ACM.

Piccardi, M. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pp. 3099 – 3104 vol.4, October 2004.

Rui, Yong, Gupta, Anoop, Grudin, Jonathan, and He, Liwei. Automating lecture capture and broadcast: technology and videography. *Multimedia Systems*, 10:3–15, 2004.

Treisman, Anne. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

Viola, Paul and Jones, Michael J. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

Zhang, Cha, Rui, Yong, Crawford, Jim, and He, Li-Wei. An automated end-to-end lecture capture and broadcasting system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4:6:1–6:23, February 2008.

A. APPENDIX

A.1. Camera dynamics model

To realize the physical behavior and the natural constraint of a real-world camera system, the trajectory of a camera can be described by a stochastic linear time-invariant (LTI) dynamic model,

$$\begin{aligned} s(i+1) &= As(i) + Bu(i) + w(i), i = 1, ..., N \quad (1) \\ y(i) &= Cs(i) + v(i), i = 1, ..., N, \quad (2) \end{aligned}$$

where $s(i) \in \mathbb{R}^2$ is the the camera trajectory state that comprises its position p and velocity $\dot{p}(i)$, $u(i) \in \mathbb{R}$ is a scalar acceleration $\ddot{p}(i)$ input , $w(i) \in \mathbb{R}^2$ is a random variable that represents the process noise and exogenous effects, $v(i) \in \mathbb{R}$ is a random variable that denotes the observation noise, $A \in \mathbb{R}^{2\times 2}$ is the state transition matrix, $B \in \mathbb{R}^{2\times 1}$ is the input-control matrix, i is the time step of the system indexed by the video frame number and N is the total number of frames in a video sequence. We assume $w(i) \sim N(0, Q)$ and $v(i) \sim N(0, R)$ are zero-mean IID drawn from a multivariate normal distribution with covariance Q and R respectively.

A.2. Kalman filter for camera trajectory prediction

Let $\tilde{y}(i)$ be the predicted trajectory output of our LTI system. Since $\tilde{y}(i)$ is only an approximation to the camera trajectory, $\tilde{y}(i)$ can be regarded as the noisy observation $y(i)$ as shown in (2). We can then apply Kalman filter to predict the true state $s(i)$ of the camera. Since $u(i)$ is an unknown input, we rearrange (1) and incorporate $\ddot{p}(i)$ as part of the state $\tilde{s}(i) \in \mathbb{R}^3$ to be predicted,

$$\tilde{s}(i+1) = \tilde{A}\tilde{s}(i) + \tilde{w}(i), i = 1, ..., N \quad (3)$$

where $\tilde{w}(i) \in \mathbb{R}^3, \tilde{w}(i) \sim N(0, \tilde{Q})$,

$$\tilde{A} = \begin{bmatrix} 1 & 1 & \frac{\Delta t}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\tilde{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

If a more complex dynamic model is needed, $\tilde{A}$ and $\tilde{C}$ can also be estimated by using an EM algorithm. After Kalman filter is applied, $s(i)$ can be computed by the following update step,

$$\begin{aligned} \tilde{s}(i|i-1) &= \tilde{A}\tilde{s}(i-1|i-1), \\ P(i|i-1) &= \tilde{A}P(i|i)\tilde{A^T} + Q, \end{aligned}$$

and the estimation step,

$$\begin{aligned} K(i) &= P(i|i-1)\tilde{C}^T(\tilde{C}P(i|i-1)\tilde{C}^T + R)^{-1}, \\ \tilde{s}(i|i) &= \tilde{s}(i|i-1) + K(i)(\hat{y}(i) - C\tilde{s}(i|i-1)), \\ P(i|i) &= (I - K(i)\tilde{C})P(i|i-1), \end{aligned}$$

and the estimated trajectory is ,

$$\hat{y}(i) = C\tilde{s}(i|i).$$