

Train Your TV

Zhi Li and Borja Peleato

Term Project for CS229 Machine Learning, Stanford University

Abstract

We study the problem of predicting the users' viewing behavior in an Interactive TV application, using soccer match as an example. Based on the information extracted from the video frames and the user's Region of Interest (RoI) trajectory history, we make the prediction of user's RoI in the frames ahead of time. We start with a generic probabilistic model, make simplifications and develop a tractable system. Lastly, we verify its performance through a set of experimental results and a live demo.

Prediction Using Video

We discretize each video frame into blocks and for each block p we compute $\Pr(I_{t+n}(p) | f^{t+n})$, where $I_{t+n}(p)$ is the indicator function of block p being in RoI in frame $(t+n)$. We further make the following simplifications:

$$\begin{aligned} \Pr(I_{t+n}(p) | f^{t+n}) &= \Pr(I_{t+n}(p) | f_{t+n}) \\ &= \Pr(I_{t+n}(p) | \xi_{t+n}(p), p) \end{aligned}$$

The second equation makes the assumption that $I_{t+n}(p)$ depends only on a local patch ξ_{t+n} around p and the actual location p . To compute $\Pr(I_{t+n}(p) | \xi_{t+n}(p), p)$, we select features to reflect $\xi_{t+n}(p)$ and p , and use a logistic regression model. The features we select include: $DIST_BALL(p)$, $MOV(p)$, $NUM_PLAYERS(p)$ and $LOCAL_LABEL(p)$.

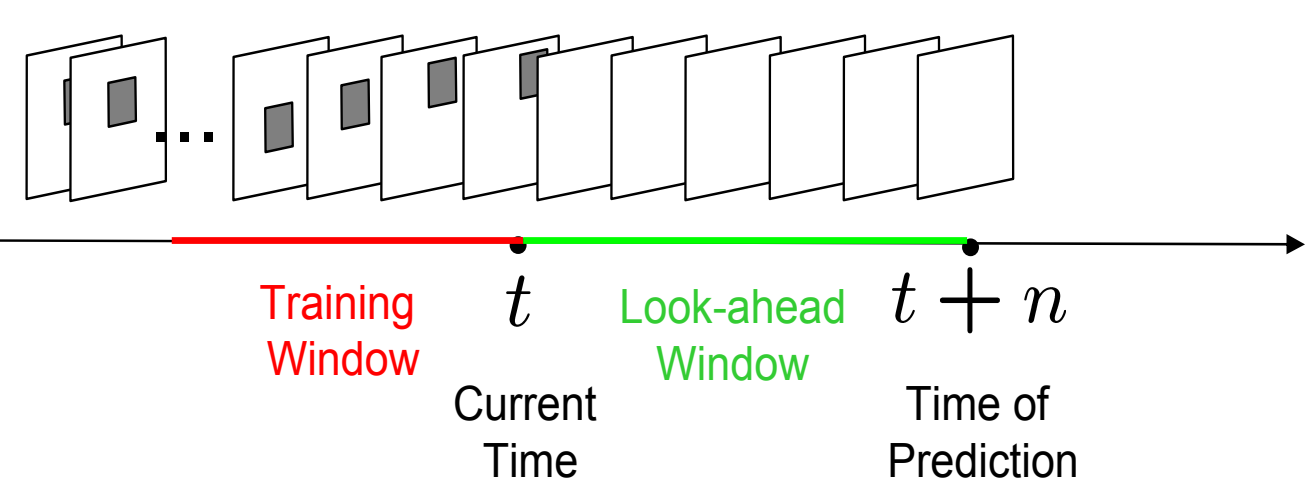
Prediction Using Trajectory

The available information is the user's recent RoI trajectory and his behavior from watching previous matches $\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_N$. We believe that there exist "typical trajectories" that users very commonly follow which would be difficult to capture under a single set of parameters. Hence, we divide the trajectories into clusters and perform an independent regression within each cluster.

Parameters and cluster centroids are computed in an offline training phase and stored to be used at prediction time. While the user is watching the match, his behavior is periodically classified into one of the clusters and his future RoI center is predicted using the corresponding coefficients. His RoI window is modeled as a Gaussian convolved with the current window.

Environment Setup

High-resolution video is available at the server. Both a low-resolution overview video and an enhancement-layer video of the RoI predicted by the system are streamed to the user. Accurate prediction of the RoI will lead to higher video quality (or less distortion) at the user side. We will allow some streaming start-up delay and send some number of frames of the thumbnail video ahead of time. Hence, the inputs to our module are the trajectory of the user's RoI history and the thumbnail video up to the frame of prediction. The performance will be evaluated based on the Euclidean distance between the predicted RoI trajectory and the actual one.



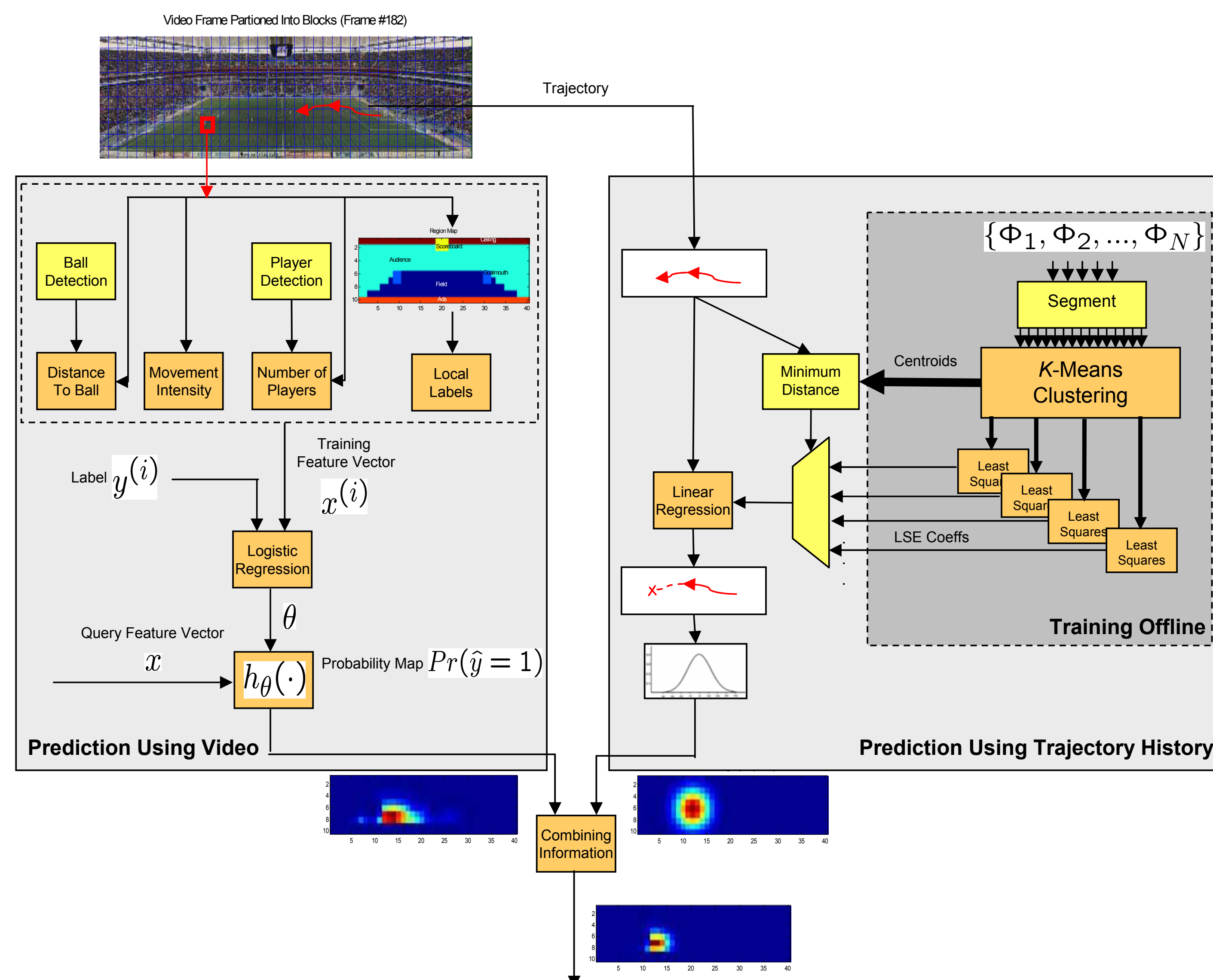
Probabilistic Model

Suppose we have a sequence of overview video frames $\{f_i\}$. On each frame the user can indicate a RoI ϕ_i .

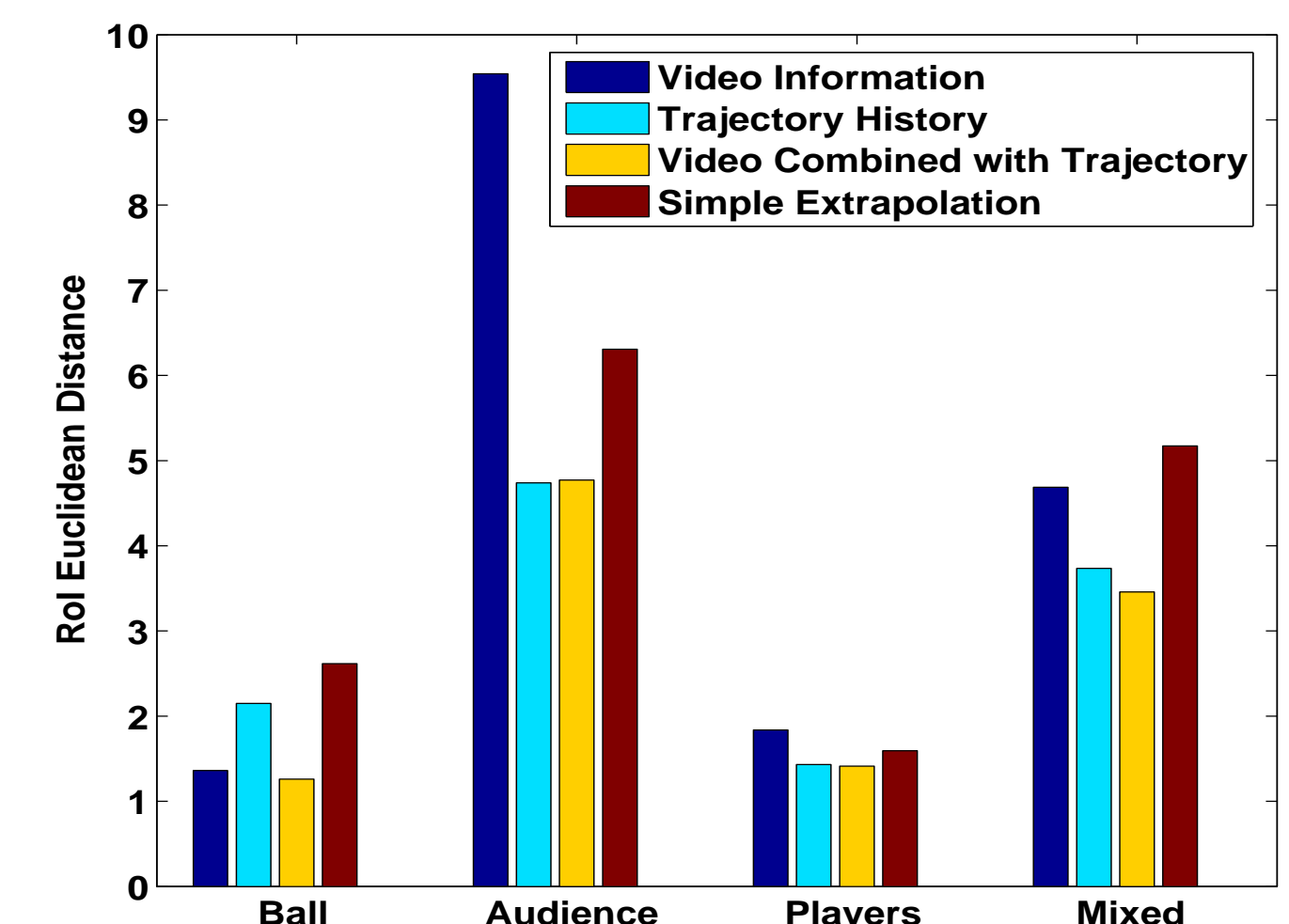
Suppose currently we are at time t , we want to make a prediction of the user's RoI at n steps ahead, i.e., ϕ_{t+n} . The information available for our prediction includes: 1) the overview video up to frame $(t+n)$, i.e., $f^{t+n} = \{f_t, \dots, f_{t+n-2}, f_{t+n-1}, f_{t+n}\}$, and 2) the user's RoI trajectory history up to frame t , i.e., $\phi^t = \{\phi_{t-2}, \phi_{t-1}, \phi_t\}$. Based on all the information available, we make a prediction using:

$$\begin{aligned} \hat{\phi}_{t+n} &= \arg \max_{\phi_{t+n}} p(\phi_{t+n} | f^{t+n}, \phi^t) \\ &= \arg \max_{\phi_{t+n}} p(\phi_{t+n} | f^{t+n}) p(\phi_{t+n} | \phi^t). \end{aligned}$$

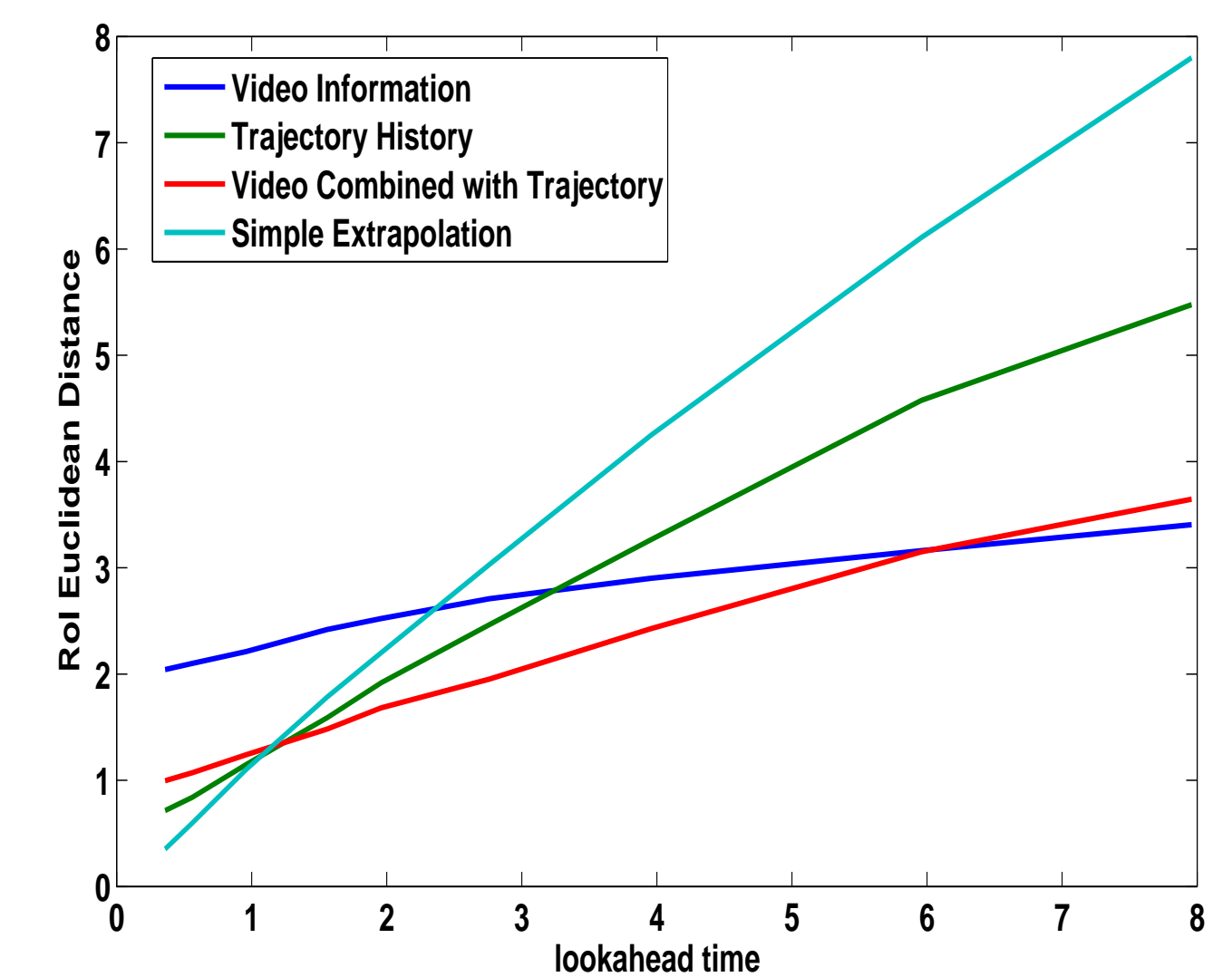
We assume Markovity ($f^{t+n} \leftrightarrow \phi_{t+n} \leftrightarrow \phi^t$) and uniform prior probability $p(\phi_{t+n})$.



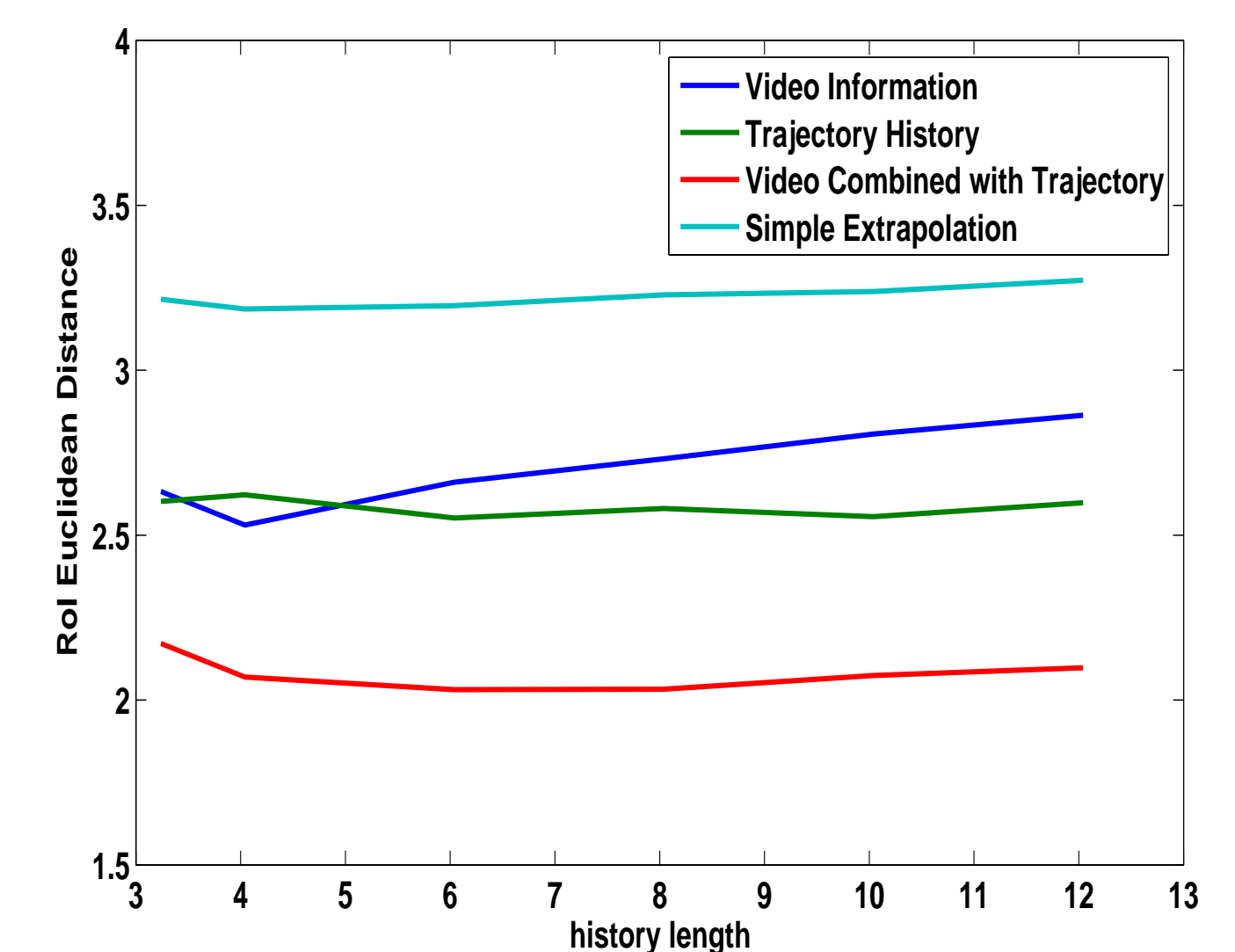
Experimental Results



Distance between predictions and correct RoI centers for different user behaviors and prediction schemes. Users can follow the ball, scan the audience, look at the players or switch randomly. Predictions respectively use video features (good for well localized behaviors), recent trajectory (good for changing behaviors), a smart combination of the previous two, and linear extrapolation (comparison benchmark).



Error vs. lookahead time. For short times, trajectory predictions are better, but as we try to predict further in time, the video features offer a very reliable source of information. We used a lookahead of 3 seconds.



Error vs. training window duration. Longer training times slightly improve the accuracy of trajectory predictions, but at the cost of increased complexity and feature degradation. We decided to base our predictions on the past 4 seconds.