Adaptive Bayesian Recognition in Tracking Rigid Objects

Yuri Boykov* Daniel P. Huttenlocher

Computer Science Department
Cornell University
Ithaca, NY, 14850

Abstract

We present a framework for tracking rigid objects based on an adaptive Bayesian recognition technique that incorporates dependencies between object features. At each frame we find a maximum a posteriori (MAP) estimate of the object parameters that include positioning and configuration of non-occluded features. This estimate may be rejected based on its quality. Our careful selection of data points in each frame allows temporal fusion via Kalman filtering. Despite "unimodality" of our tracking scheme, we demonstrate fairly robust results in highly cluttered aerial scenes. Our technique forms a natural feedback loop between the recognition method and the filter that helps to explain such robustness. We study this loop and derive a number of interesting properties. First, the effective threshold for recognition in each frame is adaptive. It depends on the current level of noise in the system. This allows the system to identify partially occluded or distorted objects as long as the predicted locations are accurate. But requires a very good match if there is uncertainty as to the object location. Second, the search area for the recognition method is automatically pruned based on the current system uncertainty, yielding an efficient overall method.

1 Introduction

Kalman filtering has been used in a number of approaches to visual tracking (e.g.,[3, 10, 4, 1, 7]). As discussed in [6, 7], the general limitation of such methods comes from their “unimodality”. Kalman filter is based on the assumption that all distributions remain gaussian. In practice, for many linear or linearizable systems this assumption is often reasonable as far as the noise in the system dynamics is concerned. The main problem is the data likelihood function which can easily be non-gaussian and multimodal in cluttered scenes. In fact, the problem heavily depends on what one means by “data”. The main original application for Kalman filter was tracking ballistic objects where the source of data were direct measurements from a radar. Unfortunately, images do not contain direct measurements of objects. The process of selecting data, or observations, becomes critical for visual tracking based on Kalman filter.

For example, in the section on tracking contours via Kalman filter in [7] the data is extracted from a single “feature curve” (r_f). This “feature curve” is obtained from the image in a more or less ad-hoc manner. Thus, it is not surprising that a unimodal tracker can easily lock on to some clutter in the image once a wrong “feature curve” is selected. One way to avoid this problem is the condensation algorithm [6, 7] that can handle arbitrary multimodal distributions via quite elegant sampling scheme. In this case, many curves are sampled in the image. All persistent matches are tracked and all occasional matches dissipate. However, the method becomes not very practical for objects described by state vectors of high dimensions. In this case it will have to draw too many samples in order to converge. Moreover, this algorithm can use only a limited set of matching techniques. Since image is tested at a large number of sampled curves, one can hardly afford anything more than independent matching of features, nodes, and etc.

Unimodal Kalman filtering still offers the advantage of a closed form temporal fusion and we address its limitations by choosing “data” more carefully. A maximum a posteriori (MAP) recognition method is used to find the best location of the object in each frame. This method incorporates dependencies between the object features and, as shown in [2], offers better robustness to clutter and partial occlusion than other matching techniques. The recognition scheme reports the best location as an observation (or “data”) if the quality of match there is sufficiently good. Otherwise the object is declared missing in this frame. Kalman filtering is then used to estimate the current state (or

*Current Affiliation: Computer Science Department, University of Western Ontario, Canada. (yuri@csd.uwo.ca) trajectory) of the object from a sequence of observed "data" points. Our approach couples Kalman filtering with a Bayesian recognition technique in such a manner that the recognition parameters are naturally estimated based on the object state estimates and the level of noise in the filter. The resulting adaptively conservative recognition scheme is reluctant to report false matches that may come from image clutter, distorted object, or a combination of both. Thus, our choice of "data" might be suitable for a unimodal filter. Our experimental results support this claim.

Note that a number of earlier papers on Kalman based tracking (e.g. [9, 1]) also acknowledged the problem of false matches and developed various criteria for rejecting an observation, such as there being too large a deviation from the predicted location. In contrast, in our approach the recognition method makes a decision at each time frame as to whether or not the object was observed in the image. This decision is based on the quality of the MAP estimate of location, not some ad hoc criterion.

An important practical consequence of our approach is that the tracker is able to continue observing an object even when it becomes partly occluded or distorted, as long as the object continues moving as predicted by the Kalman filter. This results from the calculation of an effective matching threshold for the recognition method which is proportional to the system error of the Kalman filter. This error, or uncertainty, is low when the observed locations fit the predictions. If there is substantial uncertainty in the predicted location, however, then the effective matching threshold automatically becomes larger, thus avoiding potential false matches. Note that this is not simply the use of Kalman filtering to predict a location, but rather is the coupling of the Kalman filter predictions into the matching process in order to adaptively set the matching parameters and enable more effective matching. Examples of this capability are presented below.

Finally, in our approach the search space for the recognition method is automatically computed in each frame. This is determined by extrapolating the current estimate of the object's state and the corresponding error of the Kalman filter into the new frame. If the current estimate is accurate enough then the search in the new frame is restricted to a very small area. If the recognition method detects that the object is absent in a given frame then the noise in the system dynamics makes the extrapolated state estimate less certain in the next frame. In this case the search space in that next frame is automatically enlarged. Note that in such cases, the tracker also automatically becomes more conservative at recognizing the object as noted above.

These properties are a consequence of our principled approach to model-based tracking and do not require any additional constructions. We present the results as follows. In Section 2 we describe a general formulation of our Bayesian tracking framework. The properties of our algorithm are analytically derived in Section 2.3. An implementation of a special case of the algorithm is described in Section 3 on experimental results. This implementation is fast enough for real time tracking.

2 The Tracking Framework

Our main goal is to develop a useful and principled approach to tracking a moving object in an image sequence $\{I_t\}$, given a prior model of that object. We consider the problem of tracking in discrete time $t \in \{0, 1, 2, ..., t_{end}\}$ where t can be interpreted as a frame number. The object is represented by a set of features, indexed by integers in the set $M = \{1, 2, \ldots, |M|\}$. Each feature corresponds to some vector M_i in a feature space of the model. Commonly the vectors M_i will simply specify a feature location (x, y) in a fixed coordinate system of the model, although more complex feature spaces fit within the framework. We assume that the object representation M is fixed and does not change from frame to frame. In Section 3.1 we also discuss some possibilities for adaptive object representations.

We would like to estimate an $n \times 1$ vector x_t describing the state of the object at time t. This vector may include location, velocity, acceleration, orientation, angular velocity, scale, or other information about the state of the model in a given coordinate system. Normally this coordinate system is tied to the image frame at time t. As is commonly done in state-based tracking systems, we assume that the dynamics of x_t is given by a linear stochastic equation

$$x_{t+1} = F_t \cdot x_t + w_t \tag{1}$$

where the $n \times n$ matrix F_t is the state transition matrix from t to $t+1$ and the $n \times 1$ vector w_t represents noise. We also assume that w_t is Gaussian with zero mean and a covariance matrix Q_t $(n \times n)$. Equation (1) provides a dynamic model of the object motion in the image sequence. Only the state transition matrix F_t and the covariance matrix Q_t are known at time t. The state x_t is not directly observable. At each frame t we wish to obtain an estimate $\hat{x}_t$ of the state x_t based on currently available observations.

Let y_t denote an $m \times 1$ vector of object parameters directly observable at frame t. Intuitively speaking, this means that y_t can be computed based on information available in a single image I_t. For example, y_t may include location, orientation, and scale of the object but it can not include any second order parameters like velocity and acceleration whose estimation requires information from several frames. We also require that y_t provides sufficient information about the object's positioning in frame t. In addition, we assume

$$y_t = D_t \cdot x_t$$

where D_t is a known $m \times n$ measurement matrix.

In Section 2.1 we describe a Bayesian object recognition framework that uses the object's model M to find a MAP estimate of y_t for a given frame t. Such an estimate $\hat{y}_t$ will be referred to as an observation ("data") at time t. Due to estimation errors we have

$$\hat{y}_t = D_t \cdot x_t + v_t \tag{2}$$

where v_t is an $m \times 1$ measurement noise vector. We assume that v_t is Gaussian with the zero mean and a known covariance matrix R_t $(m \times m)$. Note that the object may disappear from the scene at certain frames due to occlusions or distortions. The recognition framework of Section 2.1 specifically accounts for this possibility and provides an optimal decision test for object occlusion. Thus, new observations $\hat{y}_t$ do not necessarily come with each new frame t.

Equations (1) and (2) form a linear system of equations for the object dynamics and observations. A Kalman filter can compute the state estimate $\hat{x}_t$ based on available observations $\hat{y}_t$. In Section 2.2 we review some details of Kalman filtering necessary for understanding our tracking algorithm.

A Kalman filter may use the known system dynamics to provide some information about y_t before frame t becomes available. The Bayesian recognition technique of Section 2.1 benefits from this prior information by knowing where to look for the object in frame t. This allows efficient computation of the estimate $\hat{y}_t$ which is then in turn fed back into the Kalman filter for calculating $\hat{x}_t$. Thus, the recognition technique and Kalman filter iteratively exchange information yielding an effective tracking algorithm whose properties are discussed in Section 2.3.

2.1 Bayesian Recognition of Objects

In this section we describe the basic elements of the object recognition technique based on MAP estimation. This method was originally developed in [2] for static images. Here we review it in the context of our tracking problem. We use this recognition technique to extract an observation $\hat{y}_t$ of the object M from a given image I_t.

The vector y_t contains information about the positioning of the object in frame t. Thus, we can define an operation $\oplus$ mapping a model feature M_i and a given positioning y_t to an image feature $M_i \oplus y_t$. The exact definition of the operation $\oplus$ varies for the particular tracking task. For example, in case of translation $\oplus$ represents vector summation. Then, a match of the model M to the image I_t can be described by a pair $\{S, y_t\}$ where $S = \{S_1, S_2, \ldots, S_{|M|}\}$ is a collection of boolean variables. If $S_i = 1$ then the ith feature of the model has a matching feature in I_t and if $S_i = 0$ then it does not. In this case we say it is mismatched. For example, the event

$$\{S_1 = \ldots S_k = 1, \;\; S_{k+1} = \ldots S_{|M|} = 0, \;\; y_t = y\}$$

implies that for $1 \le i \le k$, feature i of M matches some feature in I_t located near $M_i \oplus y_t$, and that the last $(|M| - k)$ features are mismatched.

The quality of a matching configuration S for a given positioning y_t is described by the log-likelihood function

$$\begin{aligned} H_t(S \mid y_t) &= \sum_{i \in M} \left(\ln g_i(I_t | S_i, y_t) - \alpha_i \cdot (1 - S_i) \right) \\ &- \sum_{\{i,j\} \subset M} \beta_{ij} \cdot \delta(S_i \ne S_j). \end{aligned} \tag{3}$$

It is derived in [2] from a point of view of Bayesian statistics. Larger values of $H_t(S \mid y_t)$ correspond to a better quality. Coefficient $\alpha_i \ge 0$ in the first summation in (3) specifies the penalty for not matching model's feature i and $g_i(I_t|S_i, y_t)$ is the likelihood function corresponding to this feature. In section 3.2 we consider a special case of this framework where one particular choice of the likelihood g_i is specified.

Summation over all distinct pairs of features in (3) describes the "smoothness" of configurations S. Coefficients $\beta_{ij} \ge 0$ specify the level of interaction between features i and j of the model M. For example, we can choose large β_{ij} for pairs of features $\{i, j\} \subset M$ which are closely located in the feature space of the model, and small β_{ij} otherwise. In this case, we would encourage configurations S where neighboring features tend to agree in matching or nonmatching.

Let f_t be a prior distribution of positioning y_t assuming that the object is not occluded at frame t. According to [2], the MAP estimate of the match $\{\hat{S}, \hat{y}_t\}$ should maximize over all S and y_t the posterior energy

$$E_t(S, y_t) \;=\; H_t(S \mid y_t) + \ln f_t(y_t) \tag{4}$$

and satisfy the model presence test at frame t

$$H_t(\hat{S} \mid \hat{y}_t) + \ln f(\hat{y}_t) \geq K \tag{5}$$

where K is a fixed threshold. The test (5) takes into account a possibility for object's occlusion. If $\hat{S} = \bar{0}$ or (5) is false then the optimal decision is to report that the model is missing. In this case there is no observation available at frame t.

The main difficulty for implementing this object recognition technique is maximization of the posterior energy function in (4) over all S and y_t. In the most general case this problem can be solved exactly and efficiently using a combination of hierarchical pruning and graph cut techniques in [5]. In several special cases (see [2]) the energy maximization can be done analytically which further accelerates the running time of the algorithm. For example, if the level of interaction between "neighboring" features is a positive constant and if non-neighboring features do not interact directly then [2] derive *Spatially Coherent Matching* (SCM) technique. In section 3 we discuss implementational details and experimental results corresponding to the special case of our tracking algorithm where the recognition part is done using SCM.

2.2 Review of Kalman Filtering

A Kalman filter is a system of simple recurrence equations allowing convenient iterative computation of the optimal (minimum variance) estimate $\hat{x}_t$ of x_t given available observations. It also computes the covariance matrix of the current estimate errors. In this section we specify the Kalman filter equations pertaining to our tracking technique. A detailed description of Kalman filtering can be found for example in [8].

We assume that the system and measurement noise terms $\{w_0, v_0, w_1, v_1, w_2, v_2, \ldots\}$ from Section 2 are all independent of each other. This assumption is not entirely unreasonable for tracking. Besides, noise independence can be relaxed in several ways (see [8]) which we do not discuss here mainly to save space.

Let $\hat{x}_{t-1}$ be a known estimate of state at time $t-1$ and let P_{t-1} be the corresponding $n \times n$ error covariance matrix. Then with the system dynamics and measurements as in (1) and (2) a Kalman filter has the following recurrence equations for computing $\hat{x}_t$ and P_t in the next frame. There are two steps.

The first step extrapolates the estimate $\hat{x}_{t-1}$ and the error covariance P_{t-1} into frame t by computing

$$\hat{x}_{t,t-1} = F_{t-1} \cdot \hat{x}_{t-1} \tag{6}$$

$$P_{t,t-1} = F_{t-1} \cdot P_{t-1} \cdot F_{t-1}^{\top} + Q_{t-1}. \tag{7}$$

Note that we can use these extrapolations to compute the prior distribution f_t of vector $y_t = D_t \cdot x_t$ assuming that the object is not occluded in frame t. This prior summarizes all information available just before frame t is observed. It can be shown that

$$f_t = N(\bar{y}_t \mid Y_t) \tag{8}$$

is a Gaussian distribution where $\bar{y}_t = D_t \cdot \hat{x}_{t,t-1}$ is mean and $Y_t = D_t \cdot P_{t,t-1} \cdot D_t^{\top}$ is a covariance matrix. The recognition technique of Section 2.1 uses the prior $f_t(y_t)$ for efficient computation of the observation $\hat{y}_t$, if the object is not occluded.

The second step depends on whether the new frame t produces a new observation $\hat{y}_t$ or not. If there is no observation at t then

$$\hat{x}_t = \hat{x}_{t,t-1} \quad \text{and} \quad P_t = P_{t,t-1} \tag{9}$$

so that the new estimate and the new error covariance are simply extrapolations from the previous frame. However, if the new observation $\hat{y}_t$ is available then

$$\hat{x}_t = \hat{x}_{t,t-1} + K_t \cdot [\hat{y}_t - \bar{y}_t] \tag{10}$$

$$P_t = P_{t,t-1} - K_t \cdot D_t \cdot P_{t,t-1} \tag{11}$$

where

$$K_t = P_{t,t-1} \cdot D_t^{\top} \cdot [Y_t + R_t]^{-1}$$

is a gain matrix ($n \times m$) of Kalman filter. In this case $\hat{x}_t$ and P_t are different from the extrapolations $\hat{x}_{t,t-1}$ and $P_{t,t-1}$. The correction in (10) is proportional to the deviation of observation $\hat{y}_t$ form prediction $\bar{y}_t$.

Note that equation (9) can be seen as a special case of (10) and (11). In fact, absence of observation may be interpreted as an observation with infinitely large covariance R_t corresponding to a zero gain $K_t = 0$.

2.3 Properties of our Framework

Here we derive some interesting properties of our tracking framework. Many of the properties are determined by the way the prior f_t appears in the posterior energy (4) and in the test (5). Note that the search space for observation $\hat{y}_t$ in each frame t can be significantly restricted as follows. According to (5), we can observe an object only at a positioning y such that

$$\ln f(y) \geq K - H(S \mid y).$$

As can be shown from (3),

$$H(S \mid y) \leq \sum_{i \in M} \ln g_i(I_t | S_i, y_t) \leq \sum_{i \in M} \ln \gamma_i$$

where γ_i is the largest possible value of the likelihood function g_i. In practice, γ_i is usually obvious from the

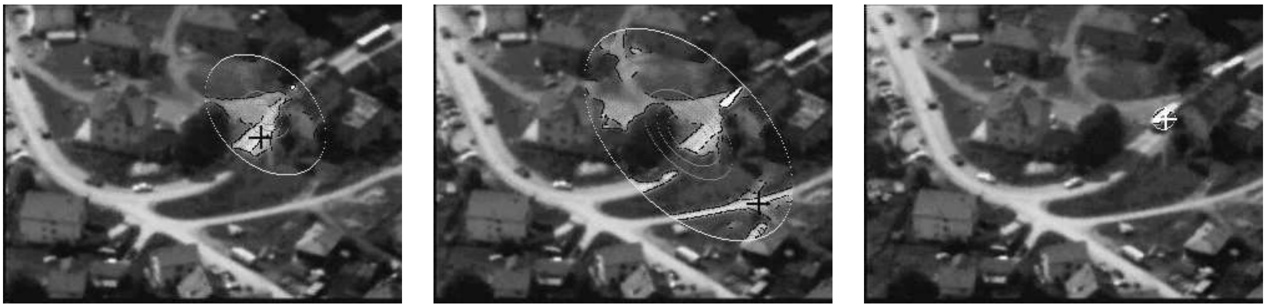


Figure 1: Occlusion example. The lock on to the object is temporarily lost (the left and central frames). The search area and the effective threshold start to grow. Erroneous matches (black crosses) are below the increased effective recognition threshold. Only a really strong match returns the lock (white cross) on to the bus when it finally becomes visible from behind the house. Note that the values of the bad matches in the left two frames are above the reduced effective threshold in the right frame where the lock on to the object is already re-established.

definition of g_i. Then, we can restrict the search for the observation $\hat{y}$ to positioning y in the set

$$\left\{ y \;\middle|\; f(y) \;\geq\; \frac{e^K}{\prod_{i \in M} \gamma_i} \right\}. \tag{12}$$

Note that (8) implies that this search region is an ellipsoid. There are several factors that affect the size of this ellipsoid. From the implementational point of view, γ_i and K are the main parameters. For fixed values of γ_i and threshold K the size of the search space depends primarily on the prior distribution f. The Kalman filter computes f_t for each new frame t. If the predicted estimate of state $x_{t,t-1}$ is accurate enough then the Gaussian prior f_t is sharply concentrated around its center $\bar{y}_t$. In this case the search space for $\hat{y}_t$ is very small. However, if the estimate is less certain then the distribution f_t is spread out and the search space increases. Consider an example in Figure 1. In this case the tracked object (a bus) becomes occluded for a number of frames by a house. Until the object shows up again the search space grows from frame to frame. This happens because observations are not coming and the uncertainty of the estimate increases due to the modeled system noise (see (7), (9)). As soon as the observations of the bus become available, the search space significantly reduces reflecting a decrease in uncertainty of the estimate (see (11)). In Section 3 we provide details of the particular implementation used to generate the data in Figure 1.

The same example can be used to illustrate another interesting property of our tracking technique. Note that using (8) we can rewrite the test for non-occlusion in the MAP recognition technique, in (5), as

$$H(\hat{S} \mid \hat{y}) - \frac{\Delta\hat{y}^\top \cdot Y_t^{-1} \cdot \Delta\hat{y}}{2} \;\;\geq\;\; K_t^* \tag{13}$$

where $\Delta\hat{y} = (\hat{y} - \bar{y}_t)$ is a deviation of positioning $\hat{y}$ from the expected value $\bar{y}_t$ and

$$K_t^* \;\;=\;\; K + \frac{\ln\left(\det Y_t \cdot (2\pi)^m\right)}{2}$$

is an *effective* decision threshold. The quadratic form in (13) penalizes large deviations $\Delta\hat{y}$. If $\hat{y} = \bar{y}_t$ then there is no penalty and if $\hat{y} \neq \bar{y}_t$ then the penalty depends on an appropriately weighted distance between $\hat{y}$ and $\bar{y}_t$. Note that the penalty for deviation $\Delta\hat{y}$ becomes less important when the state estimate is inaccurate and the error covariance Y_t is "large".

This effective decision threshold K_t^* is adaptively computed for each frame. Thus the tracker becomes more conservative at recognizing the object (the effective threshold K_t^* increases) if the current estimate of state is more uncertain and less conservative when the estimate is more accurate. This follows from a simple fact that the determinant of Y_t works as an indicator of uncertainty.

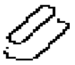
For example, in Figure 1 the value of K_t^* increases in the frames where the object is missing. In this case only a really good match may be reported as an observation. This is a very reasonable strategy minimizing the probability of a wrong match given that the search space is getting large. After the bus shows up from behind the house, we get a match that is strong enough and the tracking algorithm returns to its normal course. The estimate error reduces and the recognition process is less conservative (K_t^* decreases). This is reasonable because the state estimate becomes sufficiently accurate, the search area gets very small, and the penalty for deviation $\Delta\hat{y}$ significantly increases.

Note that if the object is missing for a significantly large number of frames then at a certain point the estimate of state becomes too uncertain. If the determinant of Y_t is so large that the effective threshold K_T^* is greater than the upper bound $\sum_{i \in M} \ln \gamma_i$ of the quality function $H(S \mid y)$ then the presence test (13) will necessarily fail. Thus, the recognition becomes no longer possible. In this case the algorithm stops and the object may be reported as lost. From the point of view of the recognition procedure, this means that any match between the object's model and an image is riskier than a decision that the object is occluded.

Our tracking framework offers additional stability in cases of high clutter in the image and partial occlusion of the object. The recognition technique of Section 2.1 explicitly models dependencies between the model features. This provides a better probabilistic model of partial occlusions of the object and, as was shown in [2], gives a significant improvement in the performance of the recognition method than compared to a number of previous techniques.

3 Experimental results

In this section we present the tracking results generated by our algorithm for a moving bus in a sequence of urban images. The sequence is obtained from a non-stationary aerial camera offering a very wide view of the scene. The quality of the film is quite challenging and the object of interest appears as a very small part of the image. The scene is highly cluttered by other vehicles, houses, and trees. We track the first of the two buses that follow each other on the road. In Figures 1 and 2 we show the results for tracking based on edge features. The model of the bus is a set of its edges extracted from the first image frame, as shown below.



This section is organized as follows. The details of a particular choice of state dynamics and system measurements are discussed in subsection 3.1. In subsection 3.2 we discuss a particular implementation of the recognition part of our algorithm based on Spatially Coherent Matching (SCM). The data is discussed in subsection 3.3.

3.1 System Dynamics and Measurements

For the experiments presented here we assumed that the object's state at each frame t is described by a 4×1 vector x_t whose components are two coordinates of a reference point of the bus, absolute value of velocity and angular orientation of the bus in the coordinate system of the image I_t. We added angular orientation to the object state vector mainly because we wish to estimate it. The orientation estimate helps to compensate for changes in the model appearance when the bus is turning.

In fact, this example shows that our initial assumption about stationarity of the object representation M is not as restrictive as it may seem. The point is that a pair $M_t = \{\hat{x}_t, M\}$ can be used as a form of adaptive object representation. We can add to the state vector x_t certain parameters of the object that may change in time assuming that these parameters can be filtered from what is observed, t.e. $\hat{y}_t$. In some cases features of the object that change in time may be added to x_t and features that do not change in time constitute the stationary part of the model, M. Note, however, that the big assumption here is that $\hat{x}_t$ is estimated by a Kalman filter. Thus, this approach would not work in cases where the system dynamics of the non-stationary part of the model, x_t, can not be reasonably linearized as in (1).

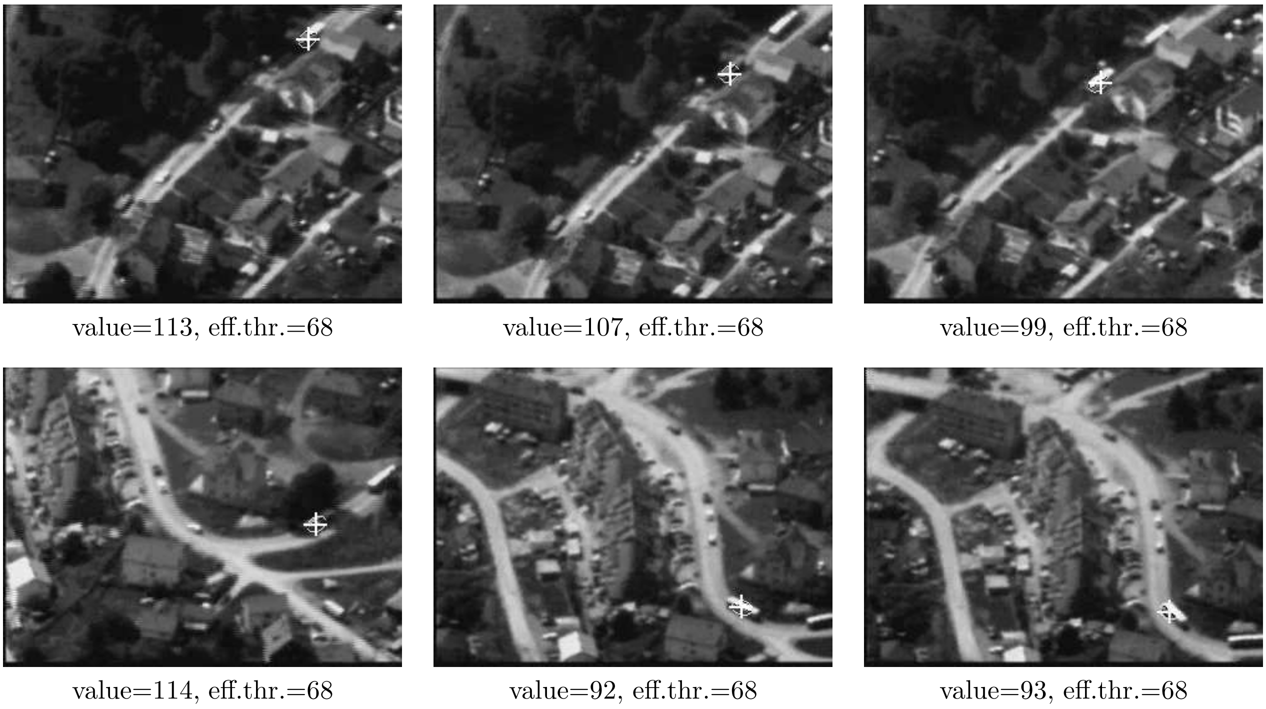
In fact, the system dynamics in our bus example is also non-linear. We assume that the vector of velocity is parallel to the body the bus. Then, the direction of velocity is uniquely determined by the angular orientation of the bus. This brings non-linear trigonometric functions into the system dynamics. In our simple case the system can be easily linearized. For the lack of space we do not give any specific details here. General information about linearizing state dynamics for the Kalman filter can be found, for example, in [8].

Note that to compensate for sporadic camera movements we also estimate affine background motion. We use a reliable robust technique based on corner features of the background. The background motion is then added to the system dynamics.

The measurement vector y_t includes coordinates of the reference point and the angular orientation of the bus in the image I_t. These are observable parameters which can be estimated at each frame using the recognition technique of section 2.1.

3.2 Spatially Coherent Matching

In this section we provide details of a particular implementation of the recognition part of our tracking algorithm. We consider a special case where maximization of the energy (4) can be done via simple SCM



value=113, eff.thr.=68

value=107, eff.thr.=68

value=99, eff.thr.=68

value=114, eff.thr.=68

value=92, eff.thr.=68

value=93, eff.thr.=68

Figure 2: Shade example (top row) and curve example (lower row).

technique [2] that is explained below in the context of the bus tracking example. Despite its simplicity, SCM based tracker is able to demonstrate the main properties of our general method.

The SCM technique gives an optimal MAP match for the recognition framework of section 2.1 for the special case when only “neighboring” model features interact directly (see [2]). SCM also relies on the likelihood function

$$g_i(I \mid S_i, y) = \begin{cases} C_0 & \text{if } S_i = 0 \\ C_1 & \text{if } S_i = 1,\ \ d_I(M_i \oplus y) \leq r \\ 0 & \text{if } S_i = 1,\ \ d_I(M_i \oplus y) > r \end{cases}$$

where $C_1 > C_0 > 0$ and r are some fixed constants, and $d_I(\cdot)$ is a distance transform of the edge features in the image I. That is, the value of $d_I(p)$ is the distance from p to the nearest edge in I. Note that r is the largest allowed distance between $M_i \oplus y$ (the bus model edge M_i positioned at y in the image) and a matching edge observed in I. For notational clarity, we dropped subindex t from “I” and “y”. The frame number should be implicit.

Let $M_y = \{i \in M : d_I(M_i \oplus y) \leq r\}$ denote the subset of object features that lie within distance r of image features I, when the model is positioned at y. We think of M_y as a set of *matchable* features for a given positioning y when the observed image is fixed. In addition, we define the complementary subset of *unmatchable* features of the object $U_y = \{i \in M \mid d_I(M_i \oplus y) > r\} = M - M_y$, also corresponding to a given positioning y. The set U_y consists of features of the object that are greater than distance r from any image features.

The main idea of the SCM scheme is to require that matching features should form large connected groups. There should be no isolated matches. Let B_y denote the subset of features in M_y that are “near” features of U_y. That is, $B_y = \{i \in M_y \mid \exists\ j \in U_y,\ d_{ij} \leq R\}$, where d_{ij} is a distance between edges M_i and M_j, and R is a fixed integer parameter. The set B_y is referred to as a *boundary* of the set of matchable features M_y. The SCM technique reports an observation of the bus

$$\hat{y} \;\; = \;\; \arg\max_y \;\; \lambda \cdot (|M_y| - |B_y|) + \ln f(y) \qquad (14)$$

if it satisfies

$$\lambda \cdot (|M_{\hat{y}}| - |B_{\hat{y}}|) + \ln f(\hat{y}) \;\; \geq \;\; K. \qquad (15)$$

Constant $\lambda > 0$ is a fixed parameter ([2] derives it from C_0, C_1, r, and R) and $f(y)$ is a prior distribution of

y that we get from (8). Note that $\hat{y}$ in (14) is a MAP estimate of y_t maximizing the posterior energy (4) for a given image $I = I_t$. The test in (15) is a special case of (5). Thus, if the optimal value of y in (14) does not satisfy (15) then the optimal decision is to report absence of the bus in the current image. Note that $\hat{y}$ can be found efficiently using hierarchical search space pruning similar to what is used for Hausdorff matching.

3.3 Discussion of Results

The algorithm is initialized by an estimate of the object's state $\hat{x}_0$ and an error covariance P_0 for the first image I_0 from which the edges of the bus model M were extracted. For $t > 0$, the estimate $\hat{x}_t$ and the matrix P_t are computed by the algorithm. We use a constant measurement noise covariance R_t. In general, it can be based on the quality of the MAP estimate $\hat{y}$. Some tracking results are shown in Figures 1 and 2.

The search region (12) in our case is a 3D ellipsoid. The particular formula for the ellipsoid

$$\left\{ y \;\middle|\; f(y) \;\geq\; e^{K-\lambda|M|} \right\}$$

can be obtained directly from (15) using a simple fact that $|M_y| - |B_y| \leq |M|$ where $|M|$ is the total number of model features. A projection of this 3D ellipsoid into 2D image plane is shown in pictures as a white contour. The enclosed area gives the region where the bus reference point is searched for in each frame. The missing dimension corresponds to angular orientation of the bus. Note that the search for $\hat{y}_t$ in (14) is restricted to a very small region when the level of noise in its state estimate is low. If the bus is in "focus" then the tracker searches through an ellipsoid of size $\approx [10 \times 5 \times 3]$ (main axis in X, Y, α dimensions).

The white cross shows the current observation $\hat{y}_t$. Position of a black cross shows the location of a would-be-observation (14) which fails the presence test (15).

Figure 1 was discussed in Section 2.3. In the upper row of Figure 2 we show that the algorithm can track a bus through a cluttered area where the model becomes significantly occluded by a shade. In this episode the estimate of object's state is quite accurate and the tracker correctly recognizes the bus even when the level of partial occlusion and clutter is high. The lower row of Figure 2 shows that the algorithm successfully tracks the bus on the curved part of the road where the object appearance significantly changes.

References

[1] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 495–501, 1997.

[2] Y. Boykov and D. Huttenlocher. A new bayesian framework for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 517–523, 1999.

[3] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:90–99, 1986.

[4] D. Gennery. Visual tracking of known three dimentional objects. *International Journal of Computer Vision*, 7(3):243–270, 1992.

[5] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[6] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.

[7] Michael Isard and Andrew Blake. *Active contours*. Springer-Verlag, 1998.

[8] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[9] T. Kanade, R.T. Collins, A.J. Lipton, P. Burt, and L. Wixson. Advances in cooperative multisensor video surveillance. In *DARPA Image Understanding Workshop*, 1998.

[10] R. Szeliski and D. Terzopoulos. Kalman snakes. In Andrew Blake and Alan Yuille, editors, *Active Vision*, pages 67–98. MIT Press, 1992.