An Adaptive Motion Model and Multi-feature Cues Based on Particle Filter for Object Tracking

Ming Li  
LanZhou University of Technology, LanZhou, China  
Email: Lim3076@163.com

Liuqing Yuan and Wenxia Du  
LanZhou University of Technology, LanZhou, China  
Email: liuqingkk@126.com, duwenxia007@163.com

Abstract—If there is an occlusion, the target state model would not match motion model anymore and measurement model also would get worse. To solve these problems, an improved particle filtering algorithm based on adaptive motion model and multiple-cue fusion is presented. Under the theory framework of particle filters, the weighted color histogram and LBP texture feature entropy are used to describe features. And the algorithm adjusts features distribution coefficient $\alpha$, automatically by calculating the Bhattacharyya distance between the object reference distribution and object sample distribution, thus the color and texture features can intelligently be fused to develop the observation model. The simple second order auto regressive model is chosen as the state transition model, and the system noise variance $\sigma_k^2$ is adaptively determined by the minimum noise variance of every feature in object tracking. For the occlusion problem, the system maximum noise variance can be selected, along with particle random motion intensified and disseminating coverage amplified. The posterior distribution of the object is approximated by a set of weighted samples, while object tracking is implemented by the Bayesian propagation of the sample set. The analyses and experiments show that the performance of the proposed method is more effective and robust to target maneuver and occlusion and has good performances under complex background.

Index Terms—Target tracking, Particle Filter, Bhattacharyya coefficient, Occlusion, system model

I. INTRODUCTION

Tracking moving objects in video sequences has many applications ranging from video surveillance to human-computer interfaces. Tracking algorithms aim to estimate the position of a target over time. To this end, a target model is first defined and then to be updated in subsequent frames using observed value. Particle filters (PF) have proven to be very effective for solving tracking problems [1, 2]. In these approaches, tracking is modeled by a state-space time series estimation problem, which is solved using a sequential Monte Carlo estimation method [2]. In the framework of particle filter, the establishment of state transition model and observation model is important to the dynamic system. In the 2D state-space, Target motion is really hard to predict, therefore, it is difficult to accurately establish target state space model. Thus target state space model generally relies on the second-order regression model, a particular statistical model, that everyone who does research in this area knows isn't right. It's a convenient assumption. Because the AR(2) may be powerful for the object with slower motion velocity and relative simple background, is not suited to the conditions of more and more complex environment. In the object description, a target model defined by single feature only can still be misled by out-of-plane object rotations, by changes in scene illumination, and by background clutter. For this reason, how to establish an adaptive motion model for different characteristics and effective integration to obtain the optimal tracking device has been a hot topic.

The major challenges encountered in visual tracking are cluttered background, maneuver, noise, occlusion, change in illumination and scale/appearance change of the objects. Considerable work has already been done in visual tracking to address the aforementioned challenges. Most of the tracking algorithms can be broadly classified into the following four categories.

1) Gradient-based methods locate target objects in the subsequent frame by minimizing a cost function [3,4].

2) Feature-based approaches use features extracted from image attributes such as intensity, color, edges and contours for tracking target objects [5,6].

3) Knowledge-based tracking algorithms use a priori knowledge of target objects such as shape, object skeleton, skin color models and silhouette [7,8].

4) Learning-based approaches use pattern recognition algorithms to learn the target objects in order to search them in an image sequence [9,10].

In recent years, many target tracking methods of adaptive movement model and feature fusion were introduced. A number of researchers have utilized multiple cues which intelligently combined information from a number of different cues to detect and track object
in scenes, and its PF-based solution, has been successfully applied to many video sequence analysis problems. Color histograms allow significant data reduction, can be computed efficiently and are robust to partial occlusions. It has been used in particle filtering for likelihood estimation [11]. A combing color and texture features based on particle filter algorithm has been application in human face tracking. It makes use of the characteristics of particle filter which not only can effectively deal with nonlinear and non Gaussian process but also can combine multiple face features information [12]. In [13], Liu and Quan presented adaptive multipe-
cue fusion based algorithm for anti-occlusions object tracking. They used the binary shape template and color histogram based on kernel ellipse as cues for tracking; and the visibility of an object was introduced to evaluate the reliability of the color and shape information under occlusions. On the other hand, the adaptive movement model has been proposed by many researchers. In [14], the proposed tracking algorithm switches between different motion models depending on a discrete label, included in the state vector. Black and Jepson proposed a mixed state-space approach to gesture recognition [15].

These methods greatly improved robustness and accuracy of target tracking. However, all of the adaptive filter methods, in filtering process, have only considered one model, but ignore the relationship between motion model and observation model. When the target maneuvering or occlusion occurred, the target state model will not match motion model anymore and transfer model errors are accumulated over time, eventually lost the target. Using a combination of features leads to the problem of how to quantify their reliability. Similarly, the importance of the motion model should be adapted to the changes in target pose and the surrounding background. This adaptation would improve the performance under changes not modeled by the tracker itself, hence removing the need for human intervention to re-tune the algorithm.

For these reasons, in this paper, an improved particle filtering algorithm based on adaptive motion model and multiple-cue fusion is proposed. Under the theory framework of particle filters, the weighted color histogram and LBP texture feature entropy are used to describe features. And the algorithm adjust features distribution coefficient \( \alpha \) automatically based on the Bhattacharyya distance between object reference distribution and object sample distribution. Thus the color and texture features can be fused under particle filter framework to develop the observation model. The state transition model is chosen as the simple second order auto regressive model, and the system noise variance \( \sigma_{v}^2 \) is adaptively determined by the minimum noise variance of every feature in object tracking. For the occlusion problem, the system maximum noise variance can be selected, along with particle random motion intensified and disseminating coverage amplified. The analyses and experiments show that the performance of the proposed method is more effective and robust.

The paper is organized as follows. The basic theory of Particle Filtering is given in Sec. II; Section III discusses the previous work on motion model and describes the selected features for target representation; Observation probability fusion mode is given in Sec. IV. Section V sets dynamic noise variance; Particle filtering algorithm is discussed in Sec. VI; Section VII presents the experimental results and analysis; Finally, Sec. VIII concludes the paper.

II. PARTICLE FILTER

Particle Filtering [16] was originally developed to track objects in clutter. The state of a tracked object is described by the vector \( X_{k} \) while the vector \( Z_{k} \) denotes all the observations \( \{z_{1}, \ldots, z_{k}\} \) up to time \( k \). Using a Bayesian filtering approach and assuming Markovian dynamics, this system can be globally represented by Means of the following two equations:

\[
x_{k} = f_{k}(x_{k-1}, v_{k}) \quad (1)
\]

\[
z_{k} = h_{k}(x_{k}, w_{k}) \quad (2)
\]

where \( f(\cdot) \) and \( h(\cdot) \) are possibly nonlinear functions, and \( v_{k} \) and \( w_{k} \) are possibly non-Gaussian noise variables. Equation (1) is the transition equation describing the dynamics of the state variable and (2) is the observation equation that determines how the measurements are obtained from the unobserved state variable. Ultimately, one would like to compute the so-called posterior probability density function (PDF) \( p(x_{k}/z_{1:k}) \), where \( z_{1:k} = \{z_{1}, \ldots, z_{k}\} \) represents the concatenation of all measurements up to time \( k \). The posterior PDF \( p(x_{k}/z_{1:k}) \) contains all the statistical information available regarding the current condition of the state variable \( x_{k} \). An estimate \( \hat{x}_{k} \) of the state then follows, for instance, as the mean or the mode of this PDF.

The solution to this Bayesian filtering problem consists in the following two steps of prediction and update [17].

Assuming that the posterior density \( p(x_{k-1}/z_{1:k-1}) \) is known at time \( k-1 \), the posterior PDF \( p(x_{k}/z_{1:k}) \) for the current time step \( k \) can be computed using the following equations:

\[
p(x_{k}/z_{1:k}) = \int p(x_{k}/x_{k-1})p(x_{k-1}/z_{1:k-1})d_{x_{k-1}} \quad (3)
\]

\[
p(x_{k}/z_{1:k}) \propto p(z_{k}/x_{k})p(x_{k}/z_{1:k-1}) \quad (4)
\]

where \( p(z_{k}/x_{k}) \) is the prior PDF, \( p(x_{k}/x_{k-1}) \) is the transition density, and \( p(x_{k}/z_{1:k-1}) \) is the so-called likelihood function.

III. TRANSITION MODEL AND OBSERVATION MODEL

Under the framework of particle filter, considered model for the moving object provides invariance to
different motions, such as translations, rotations, and to changes in the object size. This allows covering the different types of motion of the object, also the case when the object is some partial occlusion considerably and hence ensures reliable performance of the PF. So the particular implementation, which AR(2) with selected noise variance for the translational motion and mixed observation model are presented, is taken. The algorithm allows more than one model to be used for dealing with occlusions.

A. State Transition Model

For target tracking, accuracy of the state transition model is focused with great affect on tracking at high speed and efficiently. But establishing precise state transition model is very difficult. Due to the stochastic simulation mechanism of Monte Carlo in particle filter, the target state through many assumptions samples, steady estimates approximate state transition model, have two kinds of common methods: from specific training sequence learning and choose specific statistical models. However, the learning models lack adaptability. Therefore, the state transition model often uses random Brownian motion model, the second regression model and uniform motion model, etc [18].

For the purpose of tracking an object in video we initially choose a region which defines the object. The shape of this region is fixed a priori and in our case we choose a rectangle window characterized by the state vector . The size of the window is constant for simplicity, where ( ) is the center of the window, ( ) represent the velocity. We only focus on the dynamics of ( ).

As described in [19], the dynamic model could be learned from a set of pre-labeled training sequences, in our application we just model the dynamics as a zero-order motion model

\[ X_k = X_{k-1} + w_k \]  

where \( w_k \) is a zero-mean Gaussian noise, with covariance matrix, describing the uncertainty in the state vector. And exploiting earlier work on motion model [20], object dynamics are modeled as a second order process, conveniently represented in discrete time \( k \) as a second order difference equation:

\[ X_k = AX_{k-1} - BX_{k-2} + Cw_k \]  

where \( w_k \) are independent vectors of independent standard normal variables, the state-vector and \( w_k \) are matrices representing the deterministic and stochastic components of the dynamical second-order AR model, respectively. The sample set is propagated through the application of a dynamic model. While it is possible to set sensible defaults for \( A \), \( B \) and \( C \), it is more satisfactory and effective to estimate them from input data taken while the object performs typical motions. Methods for doing this via Maximum Likelihood Estimation are essential to the work described here [20] and are described fully elsewhere [21]. In the paper, we define \( A, B \) and \( C \) are

\[
A = \begin{bmatrix} 2 & 0 & \Delta T & 0 \\ 0 & 2 & 0 & \Delta T \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, 
B = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, 
C = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

B. Observation Model

Democratic integration, proposed by Triesch and von der Malsburg [22] fuses multiple cues in an adaptive manner so that the contribution of each cue depends on its estimated reliability in the current environment. Consequently such an algorithm is robust with respect to a dynamically changing background. We want to apply a particle filter in a color-based context. Color distributions are used as target models as they achieve robustness against non-rigidity, rotation and partial occlusion [23], but are sensitive to the illumination. Texture provides a way of characterizing the spatial structure of an object and can complement the use of color for reliable object tracking. However, texture is greatest impact on occlusion [24]. Hence, in the paper, a combination of a color and texture can be used advantageously to achieve real-time robust object tracking. Activating and suppressing them dynamically, may increase the performance of the system.

1) Color feature and texture feature

a) kernel-based color histograms

We do not use the entire image \( z_i \) as the measurement, but rather extract from the image the color histogram \( q_i \), computed inside the image region that is specified by the state vector \( x_k \). The center and the size of the rectangle are defined by \( (x, y) \) and \( (h, h) \), respectively. Suppose that the distributions are discretized into \( m \times b \) bins. That assigns the color at location \( x = (x, y) \) to the corresponding bin. To make the algorithm less sensitive to lighting conditions, the HSV color space could be used instead with less sensitivity to V (e.g. \( 8 \times 8 \times 4 \) bins). To increase the reliability of the color distribution when boundary pixels belong to the background or get occluded, smaller weights are assigned to the pixels that are further away from the region center by employing the Epanechnikov kernel [25]

\[
K_k (r) = \begin{cases} 
1 & r < 1 \\
0 & r > 1 
\end{cases}
\]

Where \( r \) is the distance from the region center. Thus, we increase the reliability of the color distribution when these boundary pixels belong to the background or get occluded. The color distribution \( p_s = \{ p_s \} \) at location \( x \) is calculated as

\[
p_s (x) = \frac{1}{K_k (r)} \sum_{i=1,2,\ldots,m} d_i (x)
\]
\[ p_i^n = C \sum_{i=1}^{M} k \left[ \frac{x_i - x_i'}{h} \right] \delta(b(x)) - u \]  

(8)

Where \( I \) is the number of pixels in the region, \( \delta(\cdot) \) is the Kronecker delta function, the parameter \( h = \sqrt{h_x^2 + h_y^2} \) is used to adapt the size of the region, and the normalization factor \( C = 1/\sum_{i=1}^{M} k \left[ \frac{x_i - x_i'}{h} \right] \) ensures that \( \sum_{i=1}^{M} p_i^n = 1 \).

b) Information entropy texture histograms

The LBP operator was originally designed for texture description. The local binary pattern (LBP) is a non-parametric operator which describes the local spatial structure of an image. Ojala et al [26] first introduced this operator and showed its high discriminative power for texture classification. Later, to be able to deal with textures at different scales [27] extended their original LBP operator to a circular neighborhood of different radius size. Their \( \text{LBP}_{P,R} \) notation refers to \( P \) equally spaced pixels on a circle of radius \( R \). See Fig.1 for an example of circular neighborhoods.

The decimal form of the resulting \( P \)-bit word (LBP code) can be expressed as follows:

\[ \text{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_{i'})2^{i'} \]  

(9)

where \( g_c \) corresponds to the grey value of the center pixel, \( g_i \) to the grey values of the \( i \)-th \((i = 0, \ldots, P-1)\) surrounding pixels, and function \( s(g_i - g_{i'}) \) is defined as:

\[ s(g_i - g_{i'}) = \begin{cases} 1 & \text{if } g_i \geq g_{i'} \\ 0 & \text{if } g_i < g_{i'} \end{cases} \]

In order to resolve the high computation complexity in LBP histogram of an image, we exploit this information using an information entropy histogram approximating the distribution of the texture. We use the information entropy theory [28] to describe the texture features of the image. From the viewpoint of information theory, we discuss to study LBP histogram theory using information entropy. In Shannon’s information theory, information entropy is defined as information expectation and gave the calculation formula

\[ E_n(X) = -\sum_{i} p_i \log(p_i) \]  

(10)

Where \( X \) corresponds to the signal set with the value \( x_i \), \( p_i \ (0 \leq p_i \leq 1) \) is the probability of occurrence of \( x_i \). And provides that \( 0 \log 0 = 0 \) and \( 0 \leq p^n \leq 1 \). The each ordered entropy could express different levels of information

\[ E_n(X) = -\sum_{i} p_i \log(p_i) \]  

(11)

Because \( 0 \leq p_i \leq 1 \) as the order of entropy gets higher, the level of \( p^n \) will lower and lower, which will increase the computational error. We use first-order entropy and second order entropy to analyze statistical measurement of texture histograms. Calculation equations for LBP histogram first-order entropy and second order entropy can be summarized as follows

\[ E_n(X) = -\frac{1}{H} \sum_{i=m}^{H} h(x_i) \log h(x_i) \]  

(12)

\[ E_2(X) = -\frac{1}{H^2} \sum_{i=m}^{H} h^2(x_i) \log h^2(x_i) \]

Where binary pattern quantity of LBP histogram denotes as \( H \), \( h(x_i) \) is the LBP value at \( x_i \).

IV. OBSERVATION PROBABILITY FUSION MODEL

In the likelihood function, both target and candidate distributions are represented by histogram: \( h_{\text{ref}} = \{ h_{\text{ref}}^u \}_{u=1,...,m} \) for target, and \( h_{\text{tar}} = \{ h_{\text{tar}}^u \}_{u=1,...,m} \) for the candidate, where \( u = 1, \ldots, m \) denote the bins of the histogram. The metric to measure the similarity of the target and candidate is

\[ d[h_{\text{ref}}, h_{\text{tar}}] = \sqrt{1 - \rho[h_{\text{ref}}, h_{\text{tar}}]} \]  

(13)

Where \( \rho[h_{\text{ref}}, h_{\text{tar}}] \) is the Bhattacharyya coefficient [18] that has the following form

\[ \rho[h_{\text{ref}}, h_{\text{tar}}] = \sum_{u=1}^{m} \sqrt{h_{\text{ref}}^{u} h_{\text{tar}}^{u}} \]  

(14)

The larger the coefficient \( \rho[h_{\text{ref}}, h_{\text{tar}}] \) is, the more similar the distributions are. Conversely, for the distance \( d \), the smaller the value the more similar the distributions (histograms) are. For two identical normalized histograms we obtain \( d = 0 (\rho = 1) \) indicating a perfect match.

Based on (13) a distance \( D^2 \) for color can be defined that takes into account all of the color channels

\[ D^2(h_{\text{ref}}, h_{\text{tar}}) = \frac{1}{3} \sum_{c \in \{R,G,B\}} d^2(h_{\text{ref}}, h_{\text{tar}}^c) \]  

(15)

The distance \( D^2 \) for texture is

\[ D^2(h_{\text{ref}}, h_{\text{tar}}) = \frac{1}{8} \sum_{c \in \{x_R,x_G,x_B\}} d^2(h_{\text{ref}}, h_{\text{tar}}^c) \]  

(16)

The likelihood function for the cues can be defined by

\[ \text{Figure 1. The circular (8, 1), (8, 2), and (16, 2) neighborhoods} \]
\[
p(Z_i \mid X_i) = \exp \left( -\frac{D^2(h_{ref}, h_i)}{2\sigma^2} \right) \tag{17}
\]

Where the standard deviation \( \sigma^2 \) specifies the Gaussian measurements. This allows an estimate to be made for taking into account information from the latest frame. The likelihood obtained for each cue based on the current system noise can be decided by the larger one of \( D^2_{c,\text{min}} \), of color and texture is based on \( (22) \), the minimum squared distance \( D^2_{c,\text{min}} \) of

\[
\sigma^2_{c,\text{vol}} = -\frac{D^2_{c,\text{min}}}{2\ln p} \tag{23}
\]

\[
\sigma^2_{c,\text{tex}} = -\frac{D^2_{c,\text{min}}}{2\ln p} \tag{24}
\]

Therefore define system dynamic noise variance at time-step \( k \)

\[
\sigma_k^2 = \max \left( \sigma^2_{c,\text{vol}}, \sigma^2_{c,\text{tex}} \right) \tag{25}
\]

The system noise can be decided by the larger one of smallest noise variance between characteristics, thus make the robustness of tracking system.

VI. OBJECT TRACKING BASED ON PARTICLE FILTER

Formally, we are interested in the problem of tracking targets in an occlusion environment.

A. Occlusion Handling

To further discuss adaptive updating problems of features distribution coefficient and system noise variance. In the process of prediction, the distribution coefficient and noise variance coefficients will directly affect the tracking precision, the distribution coefficient directly reacts amount of clues information. Noise variance represents the particle distribution range and reflects diversity of particles. When object is partly occluded, select the maximum noise variance to expand wave range of particles, which increases particles distributing near the target, improves particles utilization. The following figure displays the particle utilization in the dynamic system noise. See Fig.2 for an example of particle efficiency with dynamic noise variance.

B. Concrete Steps of the Algorithm

In the framework of particle filter, the basic steps of target tracking algorithm as follows:

Step1 Initialization

Detect moving region; establish Color and texture histograms; both target and candidate distributions are represented by histogram: \( h_{\text{of}} = \{ h_{\text{of}}^{(i)} \}_{i=1}^{m} \) for target, and \( h_{\text{cr}} = \{ h_{\text{cr}}^{(i)} \}_{i=1}^{m} \) for the candidate, the metric to measure the similarity of the target and candidate is Bhattacharyya (BH). When \( k = 0 \) for \( (i = 1, 2, \cdots, N) \)
generate samples \( s_0 = \{x_0, y_0, x'_0, y'_0\} \) from the initial distribution \( p(x_0) \). Based on experimental data statistics set the noise minimum-variance \( \sigma^2 = \frac{D_{2\min}}{2\ln p} \) calculate particle weights \( w_n = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d^2}{2\sigma^2}\right) \) and normalized \( w_n = \frac{w_n}{\sum w_n} \).

**Step 2 Selection step**

From the “old” sample-set \( \{s_{k-1}^{n}, w_{k-1}^{n}, c_{k-1}^{n}, n = 1, \ldots, N\} \) at time-step \( k-1 \), \( w_{k-1}^{n} \) is for particle weight and \( c_{k-1}^{n} \) for accumulation probability) construct a “new” sample-set \( \{s_{k}^{n}, n = 1, \ldots, N\} \). Select a sample as follows:

1) Generate a random number \( r \in [0,1] \), uniformly distributed.
2) Find, by binary subdivision, the smallest \( j \) for which \( c_{k-1}^{(j)} \geq r \).
3) Set \( c_{k-1}^{n} = c_{k-1}^{(j)} + w_{k-1}^{n} \), \( s_{k}^{n} = s_{k-1}^{n} \).

**Step 3 Prediction Step**

Predict by sampling from \( p\left(X_{k} \mid X_{k-1} = s_{k}^{n}\right) \) to choose each \( S_{k}^{n} \). The new sample value may be generated as: \( S_{k}^{n} = A S_{k-1}^{n} - B S_{k-1}^{(c_{k-1})} + C w_{k} \).

**Step 4 Measurement update**

1) Measure and weight the new position in terms of the measured features \( Z_{k} \), \( w_{k}^{n} = p\left(Z_{k} \mid X_{k} = s_{k}^{n}\right) \) then normalize so that \( \sum w_{k}^{n} = 1 \) and store together with cumulative probability

\[
c_{k}^{n} = 0, c_{k}^{n-1} + w_{k}^{n}, (n = 1, \ldots, N).
\]

2) According to prior particles search for new color and texture area, and calculate the similarity of target template and candidate template. Compute the color information likelihood and texture information likelihood.

**Step 5 Measurement Update**

1) On the receipt of a new measurement, compute the weights as the present moment importance weights \( w_{k}^{n} \).

2) Normalize the weights \( w_{k}^{*} = w_{k}^{n} / \sum w_{k}^{n} \).

**Step 6 Output**

Once the \( N \) samples have been constructed : estimate, if desired, moment of the tracked position at time-step \( k \) as the likelihood fusion \( p(Z_{k} \mid X_{k}) \) is calculated \( p(Z_{k} \mid X_{k}) = E\left[f(S_{k})\right] = \sum w_{k}^{(n)} f(S_{k}^{(n)}) \).

**Step 7 Loop**

Set \( k = k+1 \) and return to step 2.

Fig.3 shows the algorithm flow chart:

**VII. THE EXPERIMENTAL RESULTS AND ANALYSIS**

We test our tracking algorithm fusing multiple image cues with dynamic motion model on an image sequences. The image sequence is concerned with a person walking in a tree-lined trail. The program is implemented with MATLAB 7.0 on a computer with 2.93GHz mobile Pentium(R) 4 CPU and 512M Memory. The standard variances of color and texture likelihoods are set empirically.

The image sequence is concerned with a person walking in a tree-lined trail. The following factors existed that make tracking difficult in this case: occlusions from the roadside tree and complex non-linear deformation due to the walking behavior. We implement both the new way and an algorithm in [12] to make some comparison. At frame 35, occlusion happens until frame 50. During this process, the person is completed occluded by the tree (Fig. 4, frame 49) for several frames and the occlusion lasts for 15 frames. In these sequences, near the vicinity of frame
49, it notices that the tracker is heavily disturbed by the clutter in the background. Seen from the following images, the algorithm in [12] with a fixed motion model is hard for those target state model to match motion model to handle this occlusion. Seeing the blue rectangle, it has gradually missed the target. Thanks to the algorithm that fuses of color and texture information with adaptive motion model, the tracking results in a complex scenario as illustrated in Fig.4 are satisfactory. The red rectangle which utilizes the algorithm proposed in the paper succeeds to overcome occlusion. Once the occlusion ends the people can be recaptured and correct labeled quickly.

VIII. SUMMARY

Based on different motion models a multi-feature tracking algorithm that adaptively weights the contribution of each feature was proposed in this paper. The proposed adaptive particle filter improves the flexibility of the representation by exploiting the motion model of the various models in a simple and efficient way. Experimental results showed that the adaptive multi-feature representation formed by a combination of color and texture histograms based on dynamic motion model is more descriptive, and leads to more accurate results than a multi-feature based on fixed model.

The proposed feature reliability score with dynamic model is general and can be extended to a larger set of features. Future work includes the investigation of fusion mechanisms that account for inter-dependencies between features. Moreover, selection algorithms could be employed to dynamically disable redundant features to improve the computational efficiency of the algorithm. Also, we aim to investigate an integrated probabilistic treatment of the interaction between feature reliability estimation and particle filter sampling. Finally, we are studying a robust model update criterion driven by the estimates of the feature reliability in order to achieve longer-term tracking under varying conditions.

ACKNOWLEDGMENT

This research is supported by Natural Science Foundation of Gansu Province (0803RJZA025) and Science Foundation of Gansu Education Department (0803-07).

REFERENCES


