MVA '98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

12—2

Visual Feature Tracking with Automatic Motion Model Switching

P. Tissainayagam and D. Suter
email: [prithi, suter]@zeno.eng.monash.edu.au
Dept. of Electrical and Computer Systems Engineering
Monash University, Clayton Campus, Victoria 3168, Australia.

Abstract

In this paper we present a motion determination and tracking technique based on the combination of Bayesian multiple hypothesis and a Multiple Model Filtering (MMF) algorithm. Corner features appearing in the initial frame of an image sequence were predicted in the subsequent frames using an extension of the multiple hypothesis algorithm (MHT [1]) based on different motion models. The collection of data provided by such a system was then provided to a MMF algorithm to determine the correct motion of features. We considered different order velocity and acceleration models for the MMF algorithm and applied them to two image sequences, the PUMA and Toy car sequences. The study shows that the method proposed can distinguish between different motions depicted in an image sequence with very good tracking results.

1. Introduction

In the recent years there has been an interest in using surveillance tracking techniques for visual tracking applications. One such proposal is outlined in [1] by Cox et al. In this paper we combine the system in [1] with an MM⌐ to track and determine the motion of objects in a long dynamic image sequence. An important reason for considering the MHT algorithm is because the MHT is one of the statistical data association algorithms that integrates all the capabilities such as track initiation, track termination, track continuation, explicit modelling of spurious measurements, and explicit modelling of uniqueness constraints.

In this paper we consider the trajectories of 3 features appearing in the first frame of a sequence and analyse their motion. Our contribution is primarily on determining the motion model appropriate for the feature and introducing the MHT/MMF tracker. In section 2 & 3 we outline the MHT and feature extraction procedure used. Section 4,5 outlines the MMF technique used and section 6 provides the results and discussion. Finally section 7 gives the conclusion.

2. Multiple Hypothesis Algorithm

The Multiple Hypothesis Tracking (MHT) algorithm was originally developed by Reid [5] in the context of multi-target tracking. Cox et. al. later modified the MHT with significant computational efficiency (fig.1). See [7,1] for complete details.

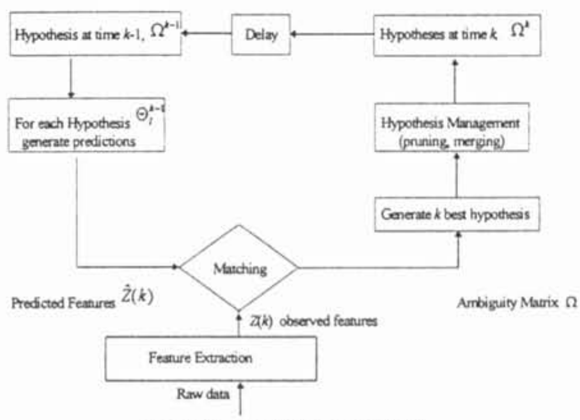

Fig.1: Outline of the multiple hypothesis algorithm

It has been shown in [1] that the predicted next hypothesis (Θ_m^k), given measurements upto k (Z^k) will be given as follows.

$$P\left\{\Theta_m^k | Z^k\right\} = \frac{1}{c}\lambda_N^{v}\lambda_F^{\phi}\prod_{i=1}^{m_k}\left[N_{t_i}[z_i(k)]\right]^{\tau_i} \left\{\prod_t (P_D^t)^{\delta_t}(1-P_D^t)^{1-\delta_t}(P_\chi^t)^{\chi_t}(1-P_\chi^t)^{1-\chi_t}\right\} P\left\{\Theta_{l(m)}^{k-1} | Z^{k-1}\right\} \tag{1}$$

Using (1) (with appropriate pruning strategy) combined with a tracking system (MMF) is what we are going to use to track features.

3. Feature Extraction

To use the multiple hypothesis tracking (MHT) technique for visual tracking, it is necessary to extract the features to be tracked in every frame of the image sequence. Normally the positions of features appearing in the first frame are predicted in the subsequent frames (matched /discarded) using a tracking system. The extracted features are also used as measurements for the tracker.

For the PUMA sequence, we used the corner detector proposed by Harris [2,7] while for the Toy car sequence we used a variant of the Lucas and Kanade's corner detector [3]. We maintained the number of corners extracted per frame to around ⌞0-50 for both seq_ences purely for clarity.

4. Tracking Features

For a visual tracking system to be efficient and reliable, the tracker needs to evolve around a correct motion model. Most visual tracking systems assume a single motion model. This assumption can be wrong if there's a change in motion depicted in the image sequence or there's multiple motions of an object. It is also quite well known that a potential weakness of an estimator based on a single model is that it can lead to under-modelling and/or over-modelling [6].

To overcome this limitations, one solution is to use a number of filters based on different motion models (sub-models) covering the range of possible expected observed motions, and to some how combine the estimates from these filters based on the expectation of each model being the correct descriptors of the features' motion. Such a system can be achieved with an MMF algorithm. As well as improving estimation accuracy, such a MMF could help in segmenting a scene into independently moving objects. It has been proposed that the segmentation process may be performed by utilising the confidence measures generated by the individual filters that make up the MMF [6]. If all objects in a scene are assumed to be rigid, all points on an object will move in an identical fashion, i.e, with the same motion model.

5. Multiple Model Adaptive Estimation (MMAE)

One type of MMF is the MMAE algorithm. For further description refer to [4]. The MMAE consists of K separate Kalman filters, each based on a particular state model. The overall state estimate is the linear combination of the state estimates generated by the individual Kalman filters, and is calculated using the following equation.

$$\hat{x}(t_i) = \sum_{k=1}^{K} p_k(t_i)\hat{x}_k(t_i), \tag{2}$$

where $\hat{x}_k$ is the state vector of the k-th Kalman filter, $p_k(t_i)$ is the weighting factor of the k-th filter at time t_i, and K is the total number of filters. The weighting factors p_k are recursively updated using:

$$p_k(t_i) = \frac{f(z(t_i)|a_k, Z_{0,i-1})p_k(t_{i-1})}{\sum_{j=1}^{K} f(z(t_i)|a_j, Z_{0,i-1})p_j(t_{i-1})} \tag{3}$$

where $p_k(t_i)$ is the probability that the actual system model, a, equals the k-th model a_k at time t_i given the past observations, $Z_{0,i-1}$ and,

$$f(z(t_i)|a_k, Z_{0,i-1}) = \frac{1}{(2\pi)^{m/2}\left|S_k(t_i)\right|^{1/2}} \exp\left\{-\frac{1}{2}\nu_k^T(t_i)S_k^{-1}(t_i)\nu_k(t_i)\right\} \tag{4}$$

where m is the number of measurements at time t_i and ν_k is the residual. The extra computation in updating the weighting factors compared to the normal Kalman filter is negligible. It should be noted that the separate Kalman filters may be run simultaneously and in parallel [4].

Equation (4) assumes that the residuals ν_k, are Gaussian and zero mean. Hence, the MMAE algorithm effectively chooses between filters based on the size of the mean of their residuals, with the one having the smallest being the *correct* filter.

For our analysis we used MMAE with 3 second order motion models. These were a constant acceleration model (M1), a constant velocity model (M2) and a constant coordinated turn model (M3) (see [7] for complete description of motion models and detailed results). Brief results of the experiments are given in figure 2 (a,b) and tables 1 and 2 for MMAE method.

The MMAE algorithm proposed by Maybeck [4] assumes that each separate KF has identical states and is of the same order. This can be seen from equation 2, where combination of the individual estimates require all the states to be present in each filter. However, this restriction is not imposed when calculating the hypothesis conditional probability (equation 3). This equation requires the separate KFs to have common measurement state variables only; the conditional probability (equation 4) is composed entirely from measurement states. Since it is only the state estimate combination equation that requires common state variables among all the KFs, it has been proposed that the standard MMAE algorithm may be extended to cope with filters having different structures and different order (but with common measurement states). Such a multi-order/differing state Multiple Model Adaptive Estimator (we call it MMAE2) uses the same probability equations (Eqs. 3,4) as the standard MMAE algorithm, but requires the state estimate combination equation to be re-written to account for any *missing* states. This can be done as given in [7].

For MMAE2 we used a third (M4) and a second (M1) order acceleration model for PUMA sequence and a third order acceleration (M4) model and a second order velocity model (M2) for the Toy car sequence. See tables 1,2 & fig. 2 (c,d) for results.

6. Results

From figure 3 it is quite clear to the naked eye that the constant acceleration model gives the best tracking performance for the PUMA sequence and the constant velocity model gives the best tracking performance for the Toy car sequence. However, we have shown experimentally that the correct motion can be 'discovered' by implementing our tracking technique (fig. 2).

For the experiments, we used constant motion models (M1-M3) for the MMAE algorithm, initialising the probability of selecting a model to 0.3333 (1/3), that is, at the start all models have an equal chance of getting selected. For MMAE2, with two models, both models were initialised to a selection probability of 0.5. Track results for the selected 3 tracks are given in figure 2 for both image sequences. The results for MMAE & MMAE2 methods are given in figure 2. Error and velocity statistics are only given for one track for PUMA and Toy car sequences (table 1 & 2). A complete results set is provided in [7].

7. Conclusion

Our study has shown how the Multiple Hypothesis Tracking (MHT) technique combined with a Multiple Model Filtering (MMF) algorithm can discriminate between different motions described by an image sequence. The results have provided evidence of our method being able to identify different motions. One of the drawback of this system is that the features need to be extracted independently of the MHT. A coupled feature detection and tracking mechanism is worth investigating in the future.

References

[1] I. J. Cox and S. L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, Feb. 1996.

[2] C. G. Harris and M. J. Stephens. " Combined Corner and Edge Detector", In *Proceedings of the Fourth Alvey Vision Conference*, Manchester, Pages 147-151, 1988.

[3] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proc. 7th International Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.

[4] P. S. Maybeck. Stochastic Models, Estimation and Control. Vol 1 & 2, *Academic Press*, NY, 1982.

[5] D. B. Reid, "An Algorithm for Tracking Multiple Targets", *IEEE transactions on Automatic Control.*, vol.24, pp.843-854, Dec. 1979.

[6] J. M. Roberts. Attentive Visual Tracking and Trajectory Estimation for Dynamic Scene Segmentation, *Phd Thesis, Dept. of Elect. and Comp. science*, University of Southampton, UK, 94.

[7] P. Tissainayagam and D. Suter, "Motion Model Selection for Visual Feature Tracking", *Tech. Report. MECS-1997-4, Monash University, Australia.*

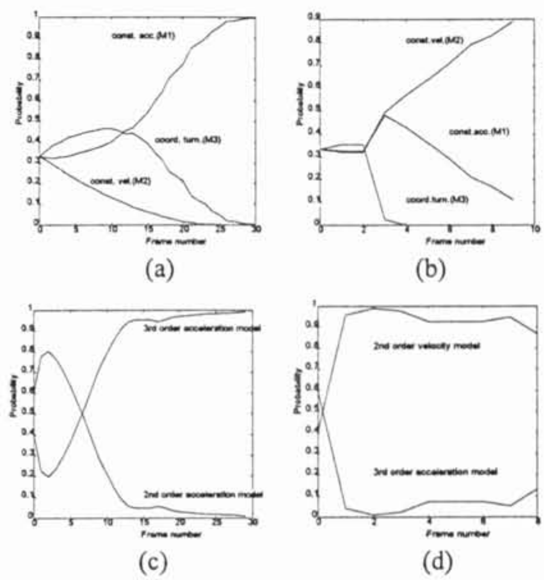

Fig. 2: (a) PUMA Seq. (MMAE): M1 selected as the ***correct*** *motion model. (b) Toy Car Seq. (MMAE): M2 selected as the* ***correct*** *motion model. (c) PUMA seq. (MMAE2): M4 selected over M1. (d) Toy car seq. (MMAE2): M2 selected over M4. See fig.3 for the corresponding track results.*

Filters in bank	**mean absolute** error (x)	**mean absolute** error (y)	RMSE (Pos.)	**mean velocity** (x-dir.)	**mean velocity** (y-dir.)	**mean velocity (mag.)**
models M1, M2, M3 (MMAE method)	0.0933	0.0791	0.1399	3.7428	0.7726	4.6897
models M1, M4 (MMAE2 method)	0.1681	0.1709	0.2625	4.4298	0.9276	5.3980

Table 1. Error & velocity statistics for PUMA sequence for track one.

Filters in bank	**mean absolute** error (x)	**mean absolute** error (y)	RMSE (Pos.)	**mean velocity** (x-dir.)	**mean velocity** (y-dir.)	**mean velocity (mag.)**
models M1, M2, M3 (MMAE method)	0.4315	0.1089	0.4506	16.3658	2.3025	16.5547
models M2, M4 (MMAE2 method)	0.2024	0.0666	0.2254	17.0010	2.2002	17.1468

Table 2. Error & velocity statistics for Toy Car sequence for track one.

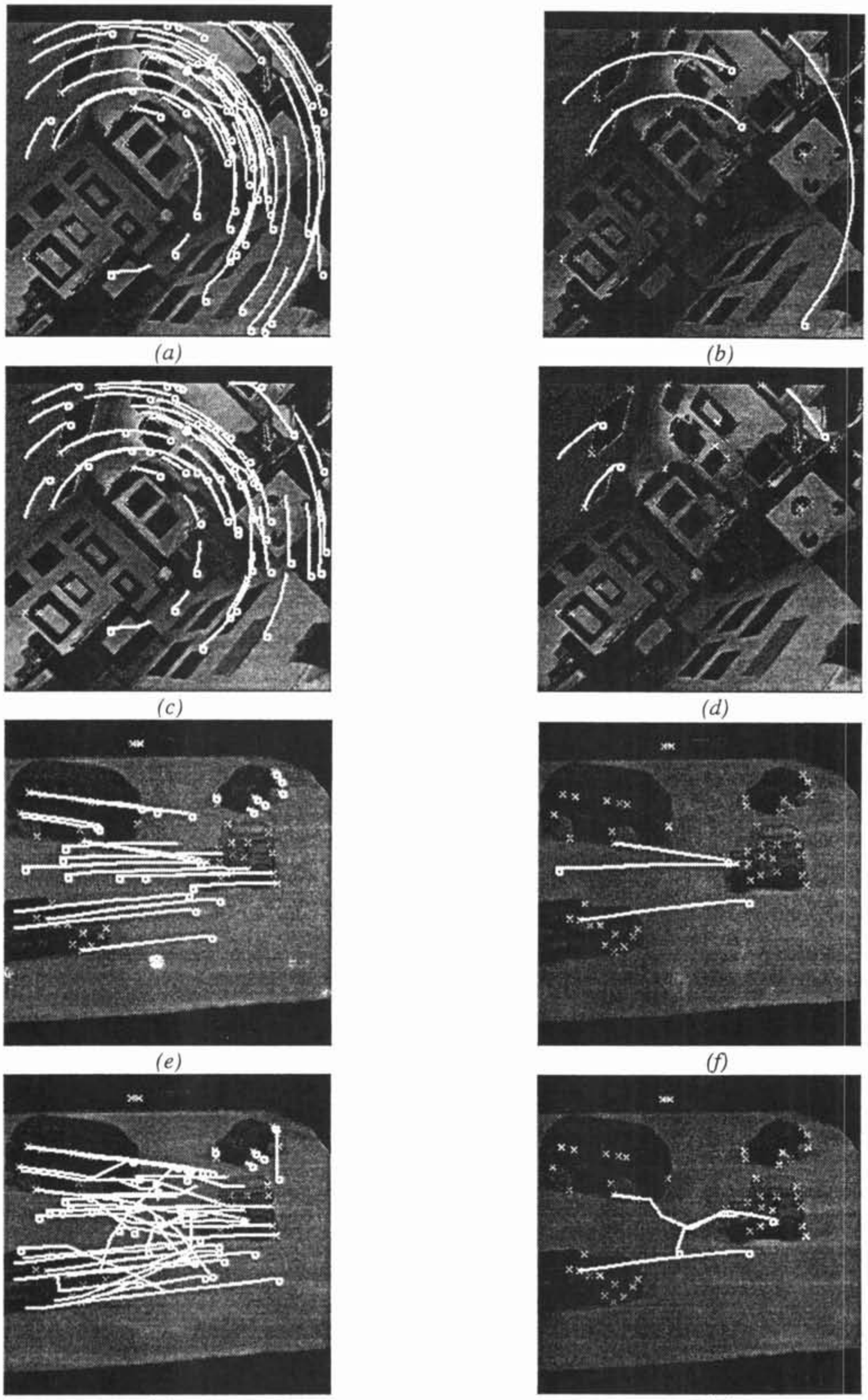

*Fig.3: Track length of more than 6 are only displayed (frame 1of PUMA & Toy car sequences). For each track the circle indicates the end of track and the 'x' indicates the corners extracted in the first frame. (a) PUMA seq., M1 with all the tracks (**correct** model). (b) The selected 3 tracks for M1 in case (a). (c) M2 with all the tracks (**incorrect** model). (d) The selected 3 tracks for M2 in case (c). (e) Toy car seq., M2 with all the tracks (**correct** model). (f) The selected 3 tracks for M2 in case (e). (g) M1 with all the tracks (**incorrect** model). (h) The selected 3 tracks for M1 in case (g).*