Massively Multi-target Tracking for Objects in Clutter

Diane E. Hirsh

May 9 2005

Abstract

We propose a method for tracking large numbers of objects in clutter in an open system. Sequences of video may contain hundreds of objects. Measurements of the state of an object may be absent, even when an object is present. Extraneous measurements (clutter) may appear when no object is present. Objects may appear or disappear at any time in a sequence. We use recursive Bayesian filters in conjunction with a probabilistic data association method for tracking objects. We use hypothesis testing for handling measurement irregularities. We quantitatively evaluate our method using simulated data. We also show the effectiveness of our method in a number of difficult imaging situations. Our method works in near real-time, after we employ simple pruning techniques which make the running time of our algorithm practically O(n) in the number of objects in the system.

1 Introduction

The goal of our work is to track a very large but unknown number of entities in an open system in near real-time. Objects may leave or enter the system at any time, and the number of objects in the system may change. Further complicating our task is the assumption that the appearance of individual objects in the system may not be distinctive (objects may have very similar appearance). We also assume that the appearance of objects may change significantly over time.

We use recursive Bayesian filters with probabilistic data association to track and estimate the state of objects. We use multiple hypothesis testing to handle missing and extraneous measurements.

The data association problem for motion correspondence has been studied since the 70's and 80's [6], [1]. Cox [2] introduced several existing data association methods to the vision community, including the joint probabilistic data association filter, JPDAF, and the multiple hypothesis tracking filter, MHT [6].

The probabilistic data association filter (PDAF) [5] is an extension of the Kalman filter. When considering multiple objects, the Joint Probabilistic Data Association Filter (JPDAF) [5] enforces an exclusion principle to keep tracks separate, by considering the joint probability of a set of object/measurement pairs and disregarding unfeasible pairs. The trouble with the JPDAF is that it requires the enumeration of all possible

legal sets of pairs [4], which is NP hard. In our solution, we draw on the idea of working with the joint likelihood of sets of pairs, but address the problem by using a cost function

The multiple hypothesis tracking (MHT) algorithm was used by Reid [3] to track corners in images. The MHT works by keeping a set of hypotheses relating all objects to all measurements. Although MHT is able to handle initiation, termination, continuation and spurious measurements, the number of hypotheses grows exponentially, and so it is only practical to use for a fairly small number of objects, even when pruning is employed. We draw on the intuition of multiple hypothesis testing, by using it at the level of the individual object, when certain failure states are encountered.

Important aspects of our work include the very large number of objects we are able to handle successfully, our ability to handle a variable number of objects in the system, and our ability to handle objects entering and leaving the system. Other important aspects of our work are the speed with which we are able to track, and no need for manual initialization. Important aspects of our solution are our probabilistic approach to the data association problem, and multiple hypothesis testing for handling missing and extraneous measurements.

2 Methods

The heart of our method is an approximate solution to the data association problem for a very large number of objects and measurements. Otherwise, our method uses standard recursive Bayesian filters, augmented by a hypothesis testing step. The steps in our method are illustrated in Figure 2.

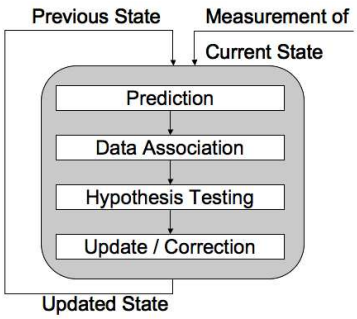

Figure 1: We illustrate the steps of our tracking algorithm.

2.1 Tracking with Recursive Bayesian Filters

There are three steps to recursive Bayesian filtering: prediction, data association, and update or correction. Given the state of an object at time $t-1$, x_{t-1}, there is a probability density over the state of the object at time t, given by $P(x_t|x_{t-1})$.

Here, we make a few reasonable simplifying assumptions, illustrated in Figure 2.1:

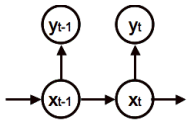

Figure 2: We assume that the state at time t, x_t depends only on x_{t-1}, and that y_t depends only on x_t.

We assume that the state of each object is generated by a first order Markov process; the state of each object at time t depends only on the state at time $t-1$. We also assume that the measurement $y_{x,t}$ of x_t depends only on the state of x at time t. These assumptions are stated formally as:

$$P(x_t|x_t - 1, ..., x_0) \ = P(x_t|x_{t-1}) \tag{1}$$

$$P(y_t|x_t, ..., x_0, y_t - 1, ..., y_0) \ = P(y_t|x_t) \tag{2}$$

Once the data association problem is solved, the estimated state of each object is updated by taking the MAP estimate of the posterior distribution of the current state of the object, given the measurement, which is given by:

$$P(x_t|y_t, x_{t-1}) \ = p(y_t|x_t)p(x_t|x_{t-1}) \tag{3}$$

If an object is not associated with a measurement during data association, we update its state by taking the MAP estimate of the prior distribution, $P(x_t|x_{t-1})$.

2.2 Data association

The goal of the data association step is to create a correspondence between a set of objects and a set of measurements. In the situation we wish to address, the data association problem is of critical importance.

The relationship between the set of measurements and the set of objects can be thought of as a bi-partite graph, where the weights on the edges are given by $P(y_t|x_t - 1)$. In the ideal case, the relationship between the set of measurments and objects is bijective. However, we assume that the sets are misaligned; there are objects with no corresponding measurements, and measurements with no corresponding objects. Due to this assumption, our problem differs from the maximum weight bi-partite matching problem (e.g. the marriage problem).

Associating each object with its most probable measurement, $\max P(y_t|x_t - 1)$ in the style of nearest neighbors, poses two problems: first, a given measurement may be the nearest neighbor to two objects, and it will not capture many properties of global optimality. Consider the case in Figure 2.2. In this case, the optimal solution would give the blue object its nearest neighbor, while aligning the red object with its second closest neighbor.

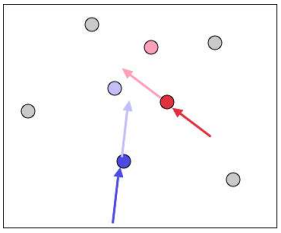

Figure 3: We illustrate a case where the nearest neighbor algorithm is not a suitable way to match objects.

Since enumerating all possible ways of aligning the set of objects and measurements, to find the optimal solution, is NP hard, our goal is to convert our problem into one that looks like the maximum weight matching problem, at the cost of accuracy of the correspondence.

We accomplish this conversion by introducing an energy function $E(y_t, x_{t-1})$, which is related to $P(y_t|x_{t-1})$. The difference between probability of the first and second best matches for an object gives an estimate of the density of measurements available to match with the object. (This density is related to events in the image, and is not related to $P(y_t|x_{t-1})$). We use this density estimate to give a handicap to objects where the density of available matching measurements is low. The energy function used in the resolution process is given by:

$$E(y_t, x_t) \ = P(y(1)_t|x_{t-1}) \ + \gamma(H) \tag{4}$$

Given that the object has at least two possible matching measurements,

$$H = P(y(1)_t|x_t - 1) - P(y(2)_t|x_t - 1) \tag{5}$$

In the case where the object has no second best matching measurement, $H = \ 1$

2.3 Resolution Algorithm

The conflict resolution algorithm converges when each object is associated with one or zero measurements, and each measurement is associated with one or zero objects. Since time t is fixed, we denote each object by x_i, and each measurement by y_j. measurements as Y. given object be denoted as Y_{x_i}.

Until convergence, we iterate over all measurements, finding all of the objects, X_i, where the given measurement, $y_j = \max_{y_j \in Y_{x_i}} P(y_j, x_i)$. Then, of the objects where $x_i \neq \max_{x_i \in X_i} E(x_i, y_j)$, the measurement is removed from Y_{x_i}.

2.4 Multiple Hypothesis Testing

The goal of the multiple hypothesis testing step is to determine the meaning of measurement irregularities. We assume a null hypothesis, and then try to accumulate evidence to refute the null hypothesis.

We define $P(\hat{y_t}|x_t)$ as the probability that an anticpated measurement is not present, even though the object remains in the system. We define $P(y_t|\hat{x_t})$ as the probability that a measurement has been made, even though no object is present in the system.

In the case when an object has not been associated with some measurement, our null hypothesis is that the object is still present in the system, but the measurement of the state is absent. We consider the joint probability that a series of measurements are missing:

$$P(\hat{y_t}, \hat{y_{t+1}}, ...|x_t, x_{t+1}, ...) = P(\hat{y_t}|x_t)P(\hat{y_{t+1}}|x_{t+1})P(...) \tag{6}$$

When this joint probability falls below some threshold $T_{disappearance}$, we conclude that the object has left the system, and the object is terminated. If the object is associated with some measurement before it is terminated, then the hypothesis that it has left the system is no longer considered.

In the case where a measurement has not been associated with an object, our null hypothesis is that the measurement is a distractor, although it may represent a new object. While treating the measurement(s) as spurious, we also pretend that the measurement represents a new object, and we begin entering it into the data association step as if it were a normal object. We consider the joint probability that a series of measurements has occured, given that there is no object present:

$$P(y_t, y_{t+1}, ...|\hat{x_t}, \hat{x_{t+1}}, ...) = P(y_t|\hat{x_t})P(y_{t+1}|\hat{x_{t+1}})P(...) \tag{7}$$

When this probability falls below some threshold $T_{appearance}$, we consider that the series of measurements represents a new object.

While we maintain the hypothesis that the series of measurements is an object, it might make sense to simply discount the series as soon as a measurement in the sequence is missing. But, this is not fair, since, if the measurements do represent an object, there is a probability that an expected measurement may be missing, given that the object is present. Considering this leads to a competition between the probability that an object has entered the system and the probability that the alleged object has left the system.

While the probability that a new object has entered the system is greater than the probability that the object has left the system, we maintain all three hypotheses. When the probability that the object has left the system exceeds the probability that a new object has entered the system, the alleged object is terminated and discounted.

2.5 Pruning and Running Time

Our method operates in four steps: prediction, data association, update, and hypothesis testing. We let N be the number of objects in the system, and M be the number of measurements from a given frame.

Both, the prediction and update steps run in $O(N)$, since we must simply make an estimate for each object, in isolation. Our data association method, in the worst case, runs in $O(NM)$, since, we must compute the probability that each measurement is associated with each object. The conflict resolution method runs in $O(N^2M)$ in the worst case. The hypothesis testing step runs in $O(\hat{N} + \hat{M})$, where $\hat{N}$ and $\hat{M}$ are the

unassociated objects and measurements, respectively, since it considers each entity in isolation.

Simple pruning techniques can reduce the running time to $O(N)$ in practice, although these techniques may not be suitable in all cases.

To prune the data association, we set a define a search window, and only consider adding measurements to the set of possible matches that are within that search radius.

The worst case running time of the conflict resolution occurs when all objects are competing for the same set of measurements in roughly the same order. In practice, only small sets of objects will compete meaningfully for each measurement. This effectively reduces the time to $O(N)$ If it must run strictly in $O(N)$, it is possible to truncate the list of possible matches that may be associated with each object.

It is worth noting that when creating the ordered sets of possible matches, using a threshold to exclude measurements from the sets changes the logic of the association in an important way : the threshold strengthens the assumption that some objects might not be associated with measurements. A tiny threshold can be very helpful, by excluding very unlikely pairs.

3 Results

Our results were produced on an Apple iBook with a 1.2 GHz PowerPC G4 processor. The size of the frame of all sequences was 320 x 240. In our final system, we use a simplified kalman filter, where we assumed that $P(x_t|y_t)$ is a dirac delta function at the position where the measurement was taken. We assume that $P(x_t|x_{t-1})$ is a Gaussian distribution over the velocity of the objects. The average and covariance of the distribution are determined using a running average, kept with exponential decay. We use a dynamically set search radius, which is set by considering the average and variance of the distance moved by the object, also kept with exponential decay. The first frame any object is on the screen, its search radius is set to be the size of the frame. We use a tiny probability threshold, $T = 0.000001$ We truncate the lists of measurements associated with each object to a length of 10 measurements. In the first frame of any sequence, the data association step is $O(NM)$, since all objects in the frame search through all measurements.

3.1 Simulation

In order to compare the performance of the tracking algorithm against ground truth, a particle system with a turbulent wind, using a variable number of particles, was used to simulate data. The number of objects and the variability in the movement of the objects could be controlled. We used tested the system using simulations with 20, 50, 100, 150, 200, and 400 particles, with a moderate amount of turbulence.

For each sequence, we calculated the percent of objects tracked correctly between two frames. The average was taken over all frames. As expected, as the number of objects in the system increases, the percent of objects tracked correctly decreases. With 20 objects in the system, 98.03% of the objects are tracked correctly, on average. With 50 objects, 97.68 are tracked correctly; with 100 objects, 95.79%; with 150 objects,

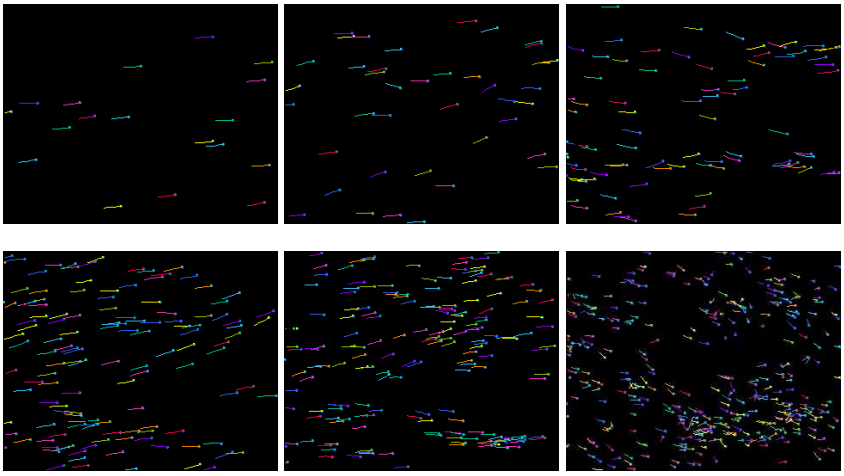

Figure 4: Frames of data from test sequences, simulated with a particle system, are shown, with short tails on each object, indicating its trajectory. From the top-left corner, there are 20, 50, 100, 150, 200, and 400 objects in the system.

95.42%; with 200 objects, 93.16%; with 400 objects, 81.47% are tracked correctly. We summarize the success rates in figure 3.1

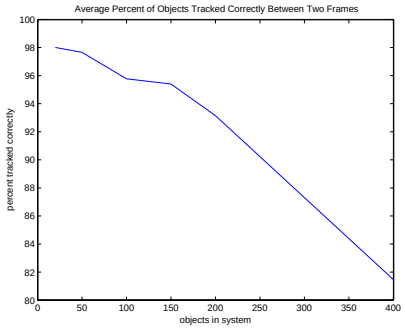

Figure 5: The average percent of correct matches decreases with the number of objects tracked, although, with 200 objects in the system, the tracker matches 93.16% of the objects correctly.

3.2 Real Data

Two types of real data were used to qualitatively evaluate the tracker. Producing ground truth for this data is an extremely difficult task for a human to do. Therefore, the method is evaluated by visual inspection of the results.

The first type of data is infrared thermal video of bats in flight. The bats were filmed during their nightly emergence from a cave, for a period of several hours.

The first set of images we show in Figure 3.2 is from the beginning of the emergence, when the beahvior of the bats is very orderly, although there is a very large number of bats and they appear very close together. The first image in the sequence shows the initial positions of the bats. The subsequent images show the bats with short tails, indicating their recent trajectory. We observe that the results of the tracking appears to make visual sense – groups of bats that appear in certain configurations between frames have similar trajectories, and the trajectories of individuals appear smooth.

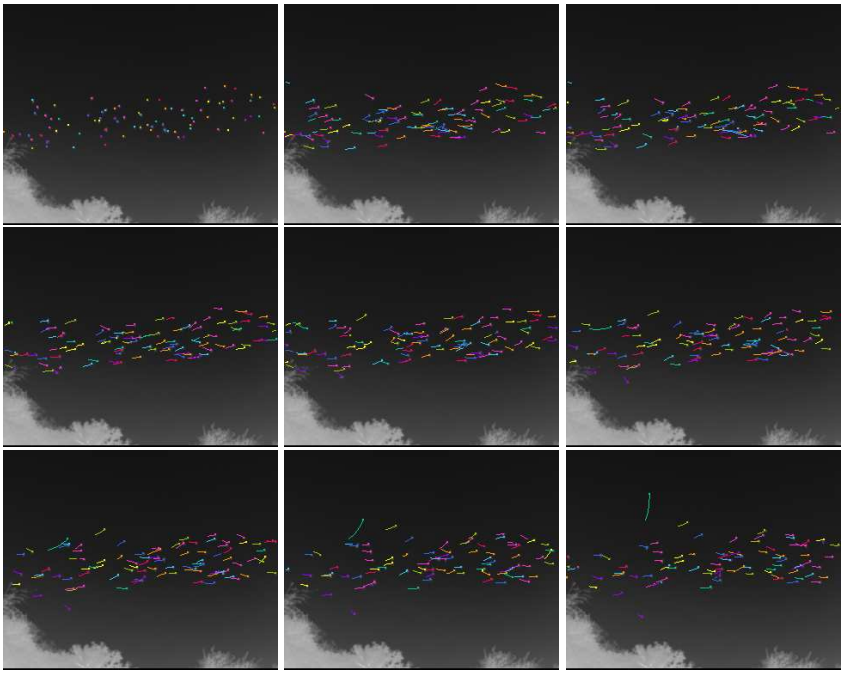

Figure 6: Frames from the early part of the emergence, when the bats appear close together, although their movement is similar in direction and magnitude. The upper-left-most frame shows the initial positions of objects to be tracked. Subsequent images are taken a few frames apart.

The second and third sets of images we show arealso taken from the early part of the emergence. The first series (Figure 3.2) shows the performance of the tracking in the presence of clutter in the trees, due to camera movement. This clutter does not cause the algorithm to make greivous errors – errors it does make, normally at the left edge of the frame, where bats are entering the field of view, are minor. The second series (Figure 3.2) shows a close-up view of the tracker's handling of occlusion.

The fourth set of images, shown in Figure 3.2 are taken from the later part of the emergence, when we observe foraging behavior. In this portion of the sequence, the bats appear farther apart, but they are flying in many different directions, at many

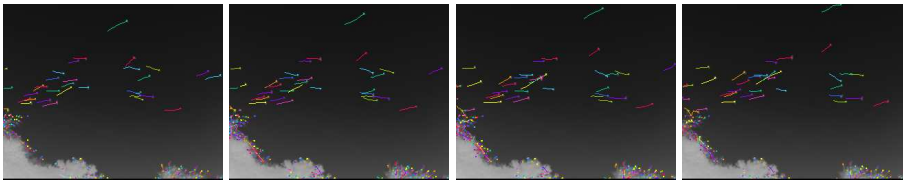

Figure 7: Although there is extensive clutter in the trees, our method is still able to track the bats well.

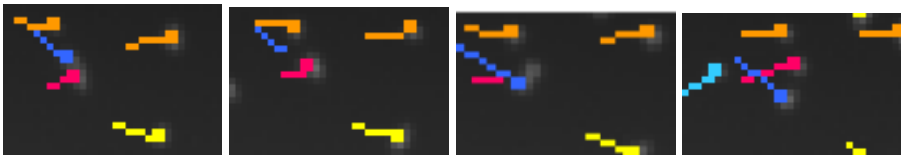

Figure 8: A close-up view of the tracker's handling of occlusion, in the midst of a dense cloud of bats. Note the two bats with the blue and pink trails.

different speeds.

The second type of data is visual camera video of a person manipulating a series of objects as part of a gestural interface. For this case, we incorporated a fairly weak appearance prior, given that the object being manipulated is very distinctive. This video is challenging to analyze because the camera's frame rate is fairly slow, and so positions of the object are fairly far apart.

Note that as the ball passes in front of the face, the ball-tracking does not skip onto some part of the face. The final image is from a frame at the end of the sequence, showing the tracking of three spheres through the course of the entire sequence. Notice that the first two balls are correctly localized in the user's hand.

3.3 Speed

Using simulated data with 20,50,100, 150, 200, 250, 400, and 800 objects, the tracking system was timed. Each sequence (except the 800 object sequence) was timed for 100 frames. The tracker has some overhead in its display function, but the time used by this overhead should have been roughly the same for each sequence. The sequence with 800 objects was timed for 76 frames, and the time for 100 frames was extrapolated.

For the sequences the 20 and 50 objects, the system took 28 seconds to track the objects through 100 frames. The sequences with 100, 150 and 200 objects took 29 seconds. The sequence with 250 objects took 30 seconds. The sequence with 400 objects took 33 seconds, and the sequence with 800 objects took 37 seconds. These times are summarized in Figure 3.3

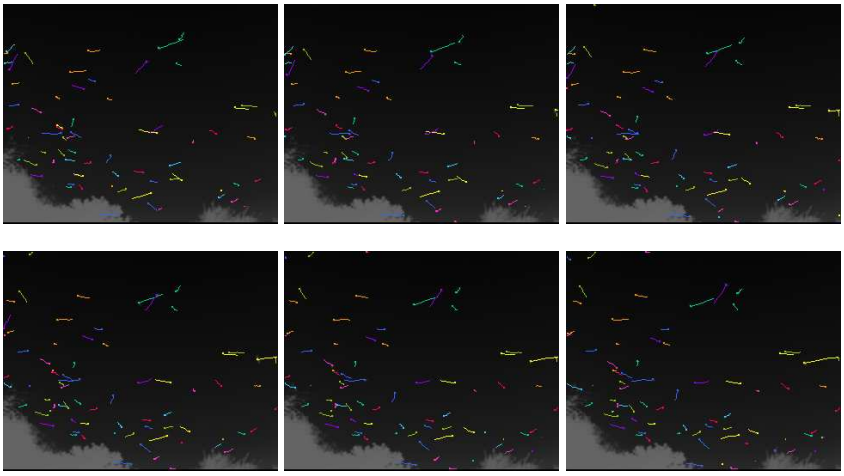

Figure 9: Frames during the late part of the emergence, when many of the bats are foraging. The bats are well-separated, but their movement is very different from each other.

4 Conclusion

We have introduced our algorithm for tracking a large number of objects in the presence of measurement irregularities in an open system. The important aspects of our work include a probabilistic approach to the data association problem, and multiple hypothesis testing for resolving ambiguities between objects entering and leaving the system and measurement irregularities. We have demonstrated the effectiveness of our method in simulated situations, in a difficult, non-traditional set of imaging situations with small, indistinct objects, and in a more traditional tracking situation with a few, visually distinctive objects.

5 Acknowledgements

Significant portions of this work were completed at Boston University under the supervision of Margrit Betke and Thomas H. Kunz. Thanks to the field team in Texas who braved many prickly pear cactuses to collect data : Eddie Lee, John Reichard, Louise Allen. Thanks to Michael Theriault for his help with quantitative validation.

References

[1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, Inc., 1988.

[2] I. J. Cox. A review of statistical data association techniques for motion correspondence. *Int J Comput Vis*, 10(1):53–66, 1993.

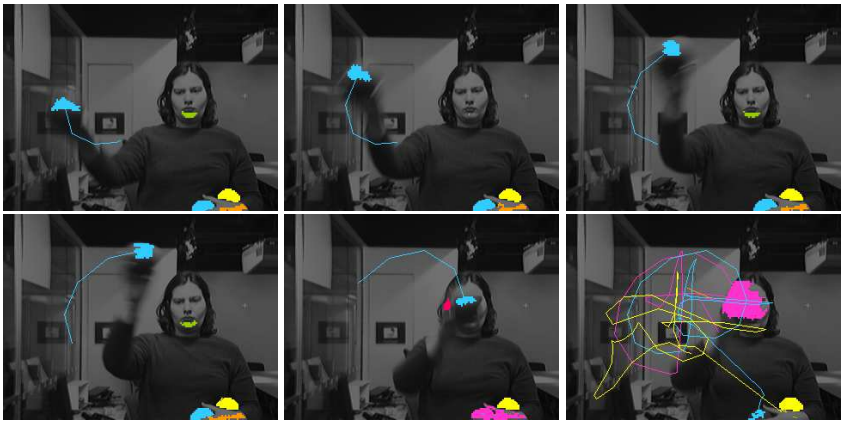

Figure 10: Our system tracks objects well in traditional imaging situations. Notice that the ball is not confused with patches on the face. The last image shows the paths of a series of three balls over the entire sequence of video. The first two balls tracked are successfully localized in the user's hand while the third ball is in use.

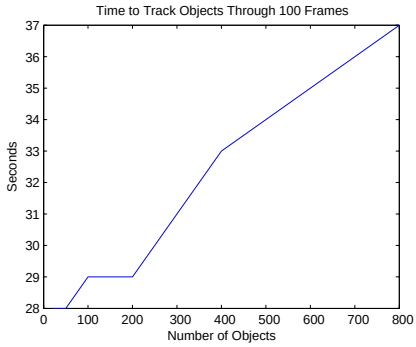

Figure 11: We show the time required to track a differing numbers of objects for 100 frames

[3] I. J. Cox and S. L. Hingorani. An efficient implementation and evaluation of reid's multiple hypothesis tracking algorithm for visual tracking. In *ICPR94*, pages A:437–442, 1994.

[4] I. J. Cox and M. L. Miller. On finding ranked assignments with application to multitarget tracking and motion correspondence. *IEEE Trans Aerosp Electron Syst*, 31:486–489, 1995.

[5] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):560–576, June 2001.

[6] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans Automat Contr*, AC-24(6):843–854, December 1979.