Attentional Based Multiple-Object Tracking

Mark Calafut
Stanford University
mcalafut@stanford.edu

Abstract

This paper investigates the attentional based tracking framework of Bazzani et al. (2011) and the general performance of attentional based tracking systems using different classification, prediction, and gaze selection techniques. Three object classifiers were implemented using the KNN technique, support vector machines, and ADA Boosting to recognize digits from the MNIST dataset. The performance of the classifiers was considered using foveated and unaltered images as input. The different classifiers, known collectively as the 'what' module were then integrated with a 'where' prediction module to determine gaze location. This module estimates object state based on previous state history, using a traditional Kalman filter. Gaze selection is then performed starting at the expected object position and moving outward using a randomized diamond shaped search pattern. The optimal gaze location is selected and used to estimate track object position. Following completion of the tracking system, its performance was evaluated using dynamic digit videos generated from the MNIST dataset. Test videos included four classes of randomly generated motion profiles and varying degrees of in-scene clutter. In general, the performance of the algorithm was robust to changes in motion profile and degree of clutter.

1. Introduction

Object tracking is an important application in the fields of machine learning and computer vision. However despite the focus it has received, in practice automated tracking systems can rarely meet the performance levels of their human counterparts. Research into the human visual system has made it clear that humans effectively use foveation and selective attention in the process of object tracking (Rensink, 2000). These techniques in conjunction with the human ability to understand context in the ongoing scenes may drive the difference in performance.

This paper attempts to replicate (with modification) the general framework of Bazzani, *et al.* (2011) to build a multiple object tracking model that incorporates foveation and selective attention. Specifically the Bazzani, et al. (2011) model consists of a general classification module (known as the ventral pathway) and an attentional or gaze selection module (known as the dorsal pathway). Bazzani *et al.* (2011) implement a particle filter in the dorsal pathway for state estimation but suggest a variety of alternative approaches for prediction. Gaze selection is learned using the online hedging algorithm of Auer *et al.* (1998). The corresponding ventral pathway performs classification using the distance based appearance model suggested by Larrochelle and Hinton (2010). The appearance model used is more expressive of shape than conventional appearance models. Overall the simultaneous use of more advanced appearance models in the ventral pathway combined with learning of gaze planning in the dorsal pathway was expected to better mimic the tracking of human observers and to lead to improved accuracy.

This work adjusts the general framework mentioned above (using a variety of different techniques) and tests the performance of the developed techniques using the MNIST dataset (LeCun and Cortes). Classification models are trained using samples from the 60,000 MNIST static digit training cases. Testing is performed using samples from the 10,000 MNIST static digit test cases. The gaze selection module is tested by generating dynamic videos from the MNIST training set. Digits are sampled randomly from the test set and placed as background clutter in a static frame. Dynamically a selected number of track objects are moved throughout the scene. Four motion profiles of the track objects are available for testing of combining classification and tracking system.

2. Classification Model

The classification model or dorsal pathway is implemented using three alternative methods. The first two methods are implemented as baseline techniques and the third technique incorporated a more flexible object representation in an attempt to improve performance. Each technique is tested using unaltered and foveated digit representations. All static classification results are presented in the following section.

2.1. Baseline Classification

Digit classification is performed using a K Nearest Neighbor model. The K Nearest Neighbor model was tested with different numbers of sample MNIST digit representations (100, 1000, and 10,000), with different numbers of neighbors considered using majority rule (1, 5, 10), and using foveated test images and unaltered images. Both foveated images and unaltered images are stored and tested as a feature vector. 24 test runs were performed with randomly selected samples over the test conditions mentioned above and the results were averaged. The results of the testing are summarized in Tables 1 and 2. Overall using K Nearest Neighbors the best results with or without foveation occur when using 10,000 representations and a single nearest neighbor. Classification accuracy in this best case was approximately 92%.

In an attempt to minimize the effects of changes in digit location within input images, a centroid adjusted KNN classifier was implemented (also using foveated inputs). This alteration did not lead to a general increase in classifier accuracy. The lack of improvement is likely due to the small 20x20 sized pixel patches used to represent each digit. Adjustment of a digit to place the digit centroid at the center of the patch would often yield rollover of the digit to the top or bottom of the patch, creating disconnected patches on the image. This technique was discarded and its test results are summarized in Table 3.

W.O Foveat.	**1 Neighbor**	**5 Neighbor**	**10 Neighbor**
100 Repre.	69.9%	64.6%	64.4%
1000 Repre.	87.3%	85.6%	86.7%
10000 Repre.	92%	91.4%	91.3%

Table 1*: KNN Results over 24 runs with 200 Samples (unaltered inputs)*

W. Foveat.	**1 Neighbor**	**5 Neighbor**	**10 Neighbor**
100 Repre.	68.0%	64.5%	63.7%
1000 Repre.	85.5%	83.2%	84.5%
10000 Repre.	92.1%	92.0%	92.8%

Table 2*: KNN Results over 24 runs with 200 Samples (Foveated inputs)*

W. Foveat.	**1 Neighbor**	**5 Neighbor**	**10 Neighbor**
100 Repre.	65.5%	59.0%	56.0%
1000 Repre.	84.5%	81.5%	79.5%
10000 Repre.	90.0%	88.0%	88.5%

Table 3*: KNN Results over 24 runs with 200 Samples (Foveated inputs and centroid adjustment)*

Also, a two class SVM representation was implemented to differentiate between a selected test digit ('8') and all other digits in the dataset. The SVM was trained using 2,000 training samples. The classification accuracy of the SVM was approximately the same using foveated inputs and unaltered inputs and was greater than 90%.

The results of LeCun et al. (1998) suggest that these baseline techniques could be further improved to 98% accuracy with modification to include deskewing and training with more samples.

2.2. Finalized Object Representation

Rather than maintaining complete raw pixel lists for each patch, alternative object representations were considered. Following initial investigation, an ADA Boost framework was selected and implemented for comparison with the KNN and SVM baseline performances. A variety of image properties were extracted from the training digit images and stored in a feature vector including the Euler Number, Orientation, Extent, Perimeter, Area, Convex Area, Solidity, Minor Axis Length, Eccentricity, Major Axis Length, Equiv Diameter, Min Intensity, Weighted Centroid, Mean Intensity, and Filled Area. These properties were chosen experimentally and used to represent training and test objects. After training they functioned as discriminators for the ADA Boost weak classifiers.

The ADA Boost algorithm was trained using the 16 feature object vectors described above with 50,000 digit training cases. The classifier was then tested using the remaining 10,000 digit cases in the MNIST database with a two-class classification (8 or not-8) accuracy of 95.9% without foveation after algorithm convergence (requiring approximately 200 boosting iterations). The accuracy of the classifier using foveated images was 96.3%. The classification accuracy of the ADA Boost technique exceeded the accuracy of the baseline methods investigated in Section 2.1, and the algorithm was ultimately selected due to its significantly superior runtime in comparison to KNN. Although the training period for ADA Boost was significant (487.7 seconds on average over the 50,000 case training database), the evaluation of new test cases after the completion of training was very rapid (~0.0098 seconds on average). This was significantly shorter than the (~0.17) second average evaluation time for test cases using KNN with 10,000 representations. This speed increase in conjunction with the slight improvement in classification accuracy lead to the selection of the ADA Boost technique as the primary classifier.

Other techniques were considered for object classification and for reducing required data storage for object representations. Several of these techniques could be explored in future work to expand upon this project and to potentially improve performance. For example, an information accumulation technique could be implemented using PCA or ICA (Koster and Hyvarinen, 2007) and PCA techniques (in conjunction with quadratic classifiers) have previously achieved accuracy in excess of 97%. Another potential method for future exploration is the use of shape context matched K-NN as implemented by Belongie *et al.* (2002) with reported classification accuracy in excess of 99%.

3. Gaze Selection

Two steps are essential to the gaze selection process. Initially the current state of the track object must be represented using descriptors such as location, orientation, speed, scale, and appearance. Second, following the estimation of the object properties at the next time increment, an optimal gaze location must be selected.

During the experimental evaluation of this project sample digits from the list of '8's in the MNIST dataset were selected and used as track objects (more details on this are provided in Section 4). The appearance of track objects was also maintained using a combination of techniques. First, the classification algorithm described in Section 2.2 (ADA Boost trained with 50,000 sample digits) was used to determine whether or not foveated gazes were similar in appearance to typical '8's. This step in the algorithm was essentially a 'sanity check', as any specific '8 'should generally resemble the class of all '8's. In addition to comparing gaze locations to generalized '8's using ADA Boost, foveated gaze locations were also compared to the specific '8' track object in the particular track video. In the first frame of each tracking experiment, the foveated template representing the specific track '8' was recorded. This template was then compared to analyzed gaze locations using metrics such as correlation (in conjunction with the weighted results of the ADA Boost classifier) and an optimal gaze location was selected. In general, it was expected that the combination of the ADA Boost framework and a generalized template tracker would provide a high degree of confidence that identified gaze locations depicted the true track object.

3.1. State Representation

A Kalman filter was implemented to estimate the position and speed of the track object in subsequent video frames. The acceleration of the object was implicitly assumed to be zero and the velocity was used to estimate the change in position of the track object. The following state equations were used in the Kalman filter:

$$x = Ax + Bu + w, w \sim \mathrm{N}(0,\mathrm{Q})$$

$$z = Hx + v, v \sim \mathrm{N}(0,\mathrm{R})$$

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = I, B = 0$$

The x variable refers to the predicted state, the z variable refers to the observed state, the A matrix reflects the state transition matrix for the dynamic system, and the observation and process noise were modeled using the Q and R matrices.

In each time step, the Kalman filter was provided an updated object location and estimated velocity based on the selected optimal gaze location. These inputs were then used to correct the previous state prediction of the Kalman filter, and following the filter update a new position and velocity prediction was made for the next time step.

3.2. Gaze Selection

The gaze selection process was critical to the performance of the tracking algorithm. The center of the gaze search at each time step was the predicted object location provided by the Kalman filter. A diamond shaped search pattern was followed, radiating outward from the Kalman selected midpoint.

The gaze search and selection process occurred in a series of iterations. At each iteration, the search pattern would be followed with 18 total gaze locations analyzed. 8 of these gaze locations were at the edges of the diamond (3 at each edge) with the 9th location was at the center of the diamond. Two sets of random values were added to the 9 template search locations, yielding a total of 18 disparate gaze search points at each iteration. The size of the diamond search pattern was initialized as 3 pixels (with a total width of 6 pixels from one edge of the diamond to the other in its widest dimension). The added pixel shift factors at each location were drawn randomly from the range of -2 to 2. An example of the diamond search pattern with randomization is shown in Figure 1 below:

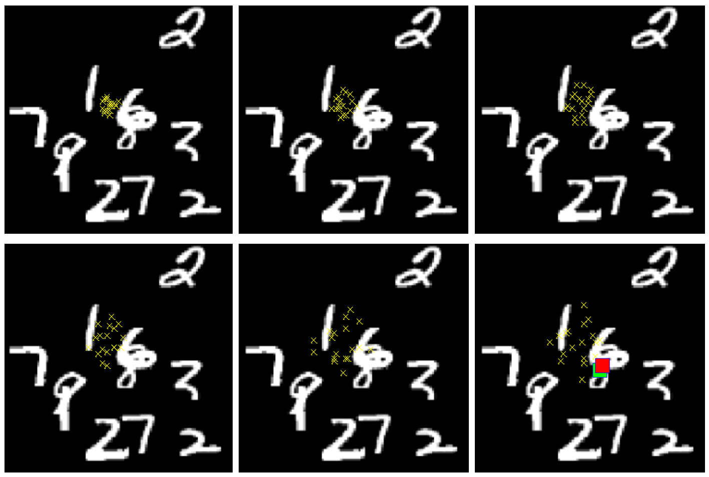

***Figure 1**:This image sequence depicts the spread of gaze locations (shown as yellow x's) according to the diamond template plus randomization. The last frame shows the true object location (marked by a red square) and the estimated location determined by the algorithm after searching (marked by a green square under the red square).*

At each gaze location the ADA Boost classifier from Section 2.2 provided an estimate of how close the object was to a typical ‘8’ object. Additionally, the correlation between the extracted foveated object template and the foveated gaze was determined. These metrics were weighted based on performance and combined to estimate the optimality of each gaze location. The optimal gaze location is ultimately the location yielding the highest value of this combined metric.

Following the completion of an iteration of the gaze search, the diamond shaped search pattern was expanded by a constant factor of 1.2x. Additionally the random shift factor was increased in proportion with the increase in the search pattern size. During each search iteration, the algorithm maintains the location of the optimal gaze location found up to that point. The search continues for a minimum of five diamond step sizes, but will continue until a reasonable steady state is achieved in the value of the estimated maximum metric at that time step.

At the completion of the gaze selection process, the object location is recorded as an observation and passed to the Kalman filter state estimator. The gaze search pattern and other parameters listed above were optimized experimentally through testing in Section 4.

4. Experimental Results

The tracking algorithm was tested by generating videos from the MNIST dataset. Clutter sources were randomly drawn from the test set and interspersed at random locations in the scene. A test digit (from the list of ‘8’s) was selected and moved throughout the scene using a linear profile with varying step size, a linear profile with added noise, a sinusoidal profile with noise, and a complex profile incorporating noise and different movement types varying over time. Static digit representations were used for initial classifier (ADA Boost) training and all training and tested was performed with uncluttered images prior to dynamic testing. Videos without clutter were used for initial state representation testing and to help select a gaze search pattern.

The tracking algorithm was tested using randomly generated dynamic videos (with varying degrees of clutter) of the types mentioned above.
Testing was performed with 0, 10, 20, and 30 randomly selected clutter digits (placed at random locations) with each of the four motion profiles. The tracking error (Euclidean distance between the estimated centroid and the true centroid at each time step) was recorded over 10 random video sequences in each setting and averaged. The results (in average pixel error) are shown in Table 4. For context, digit sizes in the image were on the order of 20 pixels in height and width. Therefore an absolute error of about 10 pixels in height and 10 pixels in width would still in general place a track point on body of the desired object. For this reason, average tracking errors exceeding 20 were indicative of qualitatively poor performance and a reasonable likelihood that the track point would be fully separated from the track object for a period of the video.

Motion Pr.	**0 Clutter Objects**	**10 Cl. Objects**	**20 Cl. Objects**	**30 Cl. Objects**
Linear	0.557	0.989	13.6	12.8
Lin. + Noise	.716	6.0	16.4	18.2
Sin. + Noise	0.550	0.705	18.4	22.5
Complex	0.490	5.2	10.6	8.6

Table 4: *Pixel error with each motion profile/clutter object combination (10 test average value for each entry).*

5. Discussion

Regardless of the motion profile being tested, the algorithm was qualitatively and quantitatively effective in tracking in all cases without background clutter. The average percentage tracking error across all tests with no clutter was 0.58%. Throughout test runs the algorithm appeared to retain track almost perfectly at all times. In general, the success in this base case provided confidence that the algorithm functioned correctly.

As the level of clutter increased, the algorithm was further stressed to classify objects seen at each gaze location. With 10 clutter objects present, effective tracking was still seen with all motion profiles. The average percentage tracking error in these situations was 3.22%. Qualitatively although the percentage error was slightly higher for the linear plus noise motion profile and the complex motion profile, this was likely attributable to randomness in the type and location of clutter present in the scene. The presence of many ‘8’s as clutter sources for example tended to be more challenging for the ADA Boost typical ‘8’ classifier and would occasional yield small increases in uncertainty in the 10 clutter object case.

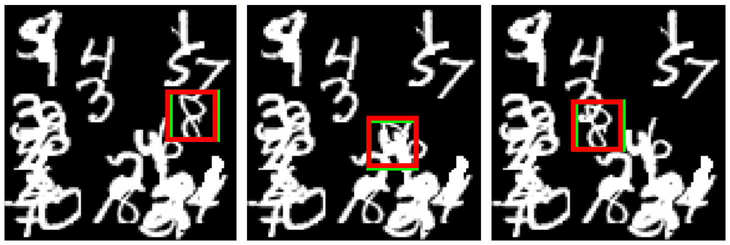

Figure 2: *This image sequence depicts the successful tracking of an object following a sinusoidal motion profile through dense clutter (30 objects). (Track Box – Green, True Location – Red)*

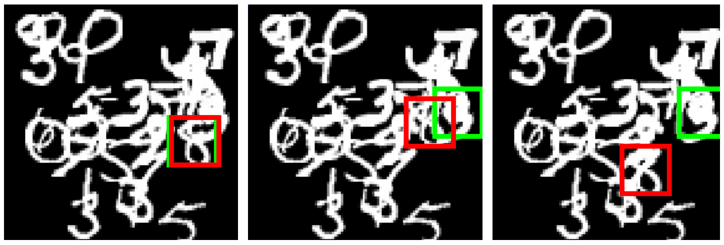

Figure 3*: This image sequence depicts the challenge of tracking through dense clutter (30 objects). After several frames of the track object moving behind dense clutter, the state estimator begins to yield inaccurate predictions. (Track Box – Green, True Location – Red, Linear + Noise Motion Profile)*

The 20 and 30 clutter object tests were more stressing for the algorithm and demonstrated interesting aspects of its performance. Specifically the average percentage tracking error in the 20 clutter objects test was 14.8% and the average percentage tracking error in the 30 clutter objects test was 15.5%. In the vast majority of test cases the qualitative tracking performance of the algorithm was very good (shown in Figure 2). In comparison to the lower clutter cases, the track algorithm would often correctly identify the target but have some small error (~10 pixels or around half of an object width) about its exact location. In a smaller percentage of track cases, the track object would be completely lost due to long term occlusion of the target. These cases would occur irregularly based on the inherent randomness of the clutter position and the motion profile of the track object. In some of these cases, the clutter was so dense that it was impossible for the human observer to identify where the track object was located for a significant portion of the video. During these long periods of uncertainty, error accumulated in the Kalman state estimate and eventually led to loss of track. In these cases, the only way for a human observer to regain the track point was to scan the entire scene repeatedly for any object movement. A frame to frame differencing technique could be built into later versions of the algorithm to specifically deal with these stressing cases. Figure 3 below depicts a situation where the track object was eventually lost while moving through a 30 clutter object scene.

In general, the tracking algorithm was able to track all motion profiles effectively. This indicates that although the constant velocity assumption in the Kalman filter was violated (particularly by the complex and sinusoidal motion profiles) the use randomization in the diagonal search pattern provided enough variety of search points to effectively identify the target. In addition, the combined appearance metric was successfully able to identify the target even with high levels of clutter. Frequently the track object would be occluded for significant periods of time, yielding low confidence matches in the appearance metric. In these scenarios, the use of the Kalman state estimator helped alleviate this uncertainty for short periods by guiding the algorithm to most likely object locations. The most stressing situations involved occlusion for very long periods (or alternatively occlusion simultaneous with a large and unpredictable change in direction of the track object).

6. Future Work

The general framework of attentional based tracking provides for a wide variety of variation. Bazzani *et al.* explored different learning mechanisms for classification and gaze selection. Aside from the algorithms implemented in this paper, there are a tremendous number of alternative options that could be tested and compared in performance using the MNIST database for testing. A focus could be placed on identifying other effective algorithms that can function in real-time. Additionally limited testing was performed using multiple-test objects in this paper (due to time constraints). Future work could specifically attempt to identify algorithms with superior performance in multiple-object tracking scenarios using the same framework presented above.

7. References

Auer P., Cesa-Bianchi N., Freund Y., and Schapire, Robert E. *Gambling in a rigged casino: the adversarial multi-armed bandit problem.* Technical Report NC2-TR-1998-025, 1998.

Bazzani L., Freitas N., Larochelle H., Murino V., and Ting J.. *Learning Attentional Policies for Tracking and Recognition in Video with Deep Networks*. Proceedings of the 28th International Conference on Machine Learning, 2011.

Belongie S., Jitendra M., Puzicha J. *Shape Maching and Object Recognition Using Shape Contexts.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24. No. 24. 2002

Freund Y. and Schapire R. *A Short Introduction to Boosting*. Journal of Japanese Society for Artificial Intelligence, 14(5). 1999

Koster U. and Hyvarinen A. *A two-layer ICA-like model estimated by score matching.* In International Conference of Artificial Neural Networks, pp. 798-807, 2007.

Larochelle H. and Hinton G. *Learning to combine foveal glimpses with a third-order Boltzmann machine.* In Neural Information Processing Systems, 2010.

LeCun Y., Cortes C. *The MNIST Dataset*.

Rensink, Ronald A. *The dynamic representation of scenes.* Visual Cognition, pp. 17-42, 2000.