Improving Performance via Post Track Analysis

Lisa M. Brown, Max Lu, Chiao-Fe Shu, Ying-li Tian, Arun Hampapur
IBM T.J. Watson Research Center, Hawthorn, NY 10532
lisabr@us.ibm.com

Abstract

In this paper, we improve the effective performance of a surveillance system via post track analysis. Our system performs object detection via background subtraction followed by appearance based tracking. The primary outputs of the system however, are customized alarms which depend on the user's domain and needs. The ultimate performance therefore depends most critically on the Receiver Operating Characteristic curve of these alarms. We show that by strategically designing post tracking and alarm conditions, the effective performance of the system can be improved dramatically. This addresses the most significant error sources, namely, errors due to shadows, ghosting, temporally or spatially missing fragments and many of the false positives due to extreme lighting variations, specular reflections or irrelevant motion.

1. Introduction

In practice, the performance of digital video surveillance depends on the conditions and needs of the environment and users of the system. Furthermore, the final outputs of such a system, although they rely on background subtraction and tracking, are specific alarms which are generated. This paper addresses the effective performance of such systems, specifically the rate of false positives, false negatives and fragmentation errors generated by a range of scene conditions.

Our system [Hampapur 05] has two major components. The first component, the Smart Surveillance Engine or SSE, performs video analytics. The second component, Middleware for Large-Scale Surveillance or MILS, stores the output of the video analytics in a database for retrieval over the web.

The video analytics makes several alarm conditions available to the user. These are: motion detection, directional motion detection, abandoned object, object removal, and camera obstruction or blind.

In order to detect alarm conditions, the system depends on the performance of the background subtraction, and tracking. This is followed by a new process which we call post track analysis. The post track analysis plays a fundamental role in "cleaning-up" the tracking output by removing tracks which are likely to be false positives, merging tracks which are likely to be a single object and detecting global system failure. The following list details the tracking problems addressed by the post track analysis:

- false positives due to lighting changes
- false positives due to specular reflections and irrelevant motions
- false positives due to "ghosting" or "healing" – that is, due to an object appearing from the background or temporarily becoming part of the background.
- temporal fragmentation
- spatial fragmentation
- global failure detection

In the next section of the paper, we discuss the related work in post track analysis and the motivation for our algorithm. In Section 3, we describe the methods we use to develop and perform post track analysis including track evaluation, false positive detection and a track stitching algorithm. In Section 4, we show the results of the system with and without post track analysis. Finally we give our conclusions in Section 5.

2. Background

Typically, surveillance systems perform tracking to associate the moving objects detected in each frame with the moving objects detected in subsequent frames. The tracking algorithm may be based on motion information, (such as mean-shift or Kalman filtering), raw image information (such as appearance-based tracking) and/or shape/contour information (snake-based approaches). Traditionally, these algorithms try to address the problems of track merge and split and may even "filter" tracks which are likely to be "unviable." However, the performance of these latter operations depends significantly on the type of input data and camera noise.

For example, the requirements for successful tracking differ significantly when objects appear to be very large in the scene and move very close to the foreground and hence very quickly in the image view than in the case where objects appear small and to be moving slowly across the image. Tracking algorithms are sensitive to the size and speed of the objects in the scene and hence an algorithm/parameterization which performs well on an outdoor road/pedestrian scene may not perform as well when viewing large aircraft moving slowly and sporadically on a tarmac. Similarly, the performance of the background subtraction phase can affect the performance of the tracker but the knowledge of a noisy environment can be used to predict the likelihood of certain types of track fragmentation and false positives. Many of the pros and cons of different algorithms and parameterizations can be corrected via the use of a generic post processing step which is performed after tracking is completed.

Mature tracking algorithms are equipped to handle many of the issues discussed here. Investigators have considered feedback between background subtraction and tracking in [Taycher 05] or even the combination of tracking algorithms as in [Leichter 04]. This paper is an attempt to modularize the track clean-up step while remaining agnostic to the underlying tracking algorithm. The system has been developed to be plug-and-play so different background subtraction methods, tracking algorithms and post processing parameters can be set, in an attempt to achieve the optimal performance for a range of problem domains and allow the system to evolve with the latest technological developments.

In the paper by [Lipton 03] from ObjectVideo, system performance is measured based on the probability of detection for a given alarm and false alarm rate. They show significant performance improvements by using a combination of alarms and the addition of filters for specific domains. We believe this methodology is very useful in practice and incorporate a similar outlook.

The method described here has been developed based on the output of our track evaluation system described in [Brown 05]. The track evaluation system is capable of detecting track false positives (FP) and track false negatives (FN) for visualization. A "track" (FP/FN) refers to an entire track for which there is not sufficient evidence. In Figure 1 we show several examples of different types of track false positives including track false positives due to a stationary object which begins to move – so called "ghosting," a specular reflection from a glass building, and a strong shadow of a preson cast on the road on an extremely bright day.

Figure 1 Several examples of different types of track false positives marked in white circles, including (top) track false positives due to a stationary object which begins to move – so called "ghosting," – from PETS 2001 data, (middle) a specular reflection from a glass building not visible in the scene at right and (bottom) due to a strong shadow of a person cast on the road from an extremely bright day.

3. Method

Our method involves the detection of three types of problems: false positives, spatial fragmentation and temporal fragmentation. The method was developed and tested using the results of our track evaluation system which can be used to detect examples of each of the three types of problems.

The track evaluation system uses a two-phase one-to-many track matching scheme. In this way, both track false positives and track false negatives are properly discriminated and both track over-merging and track fragmentation can be detected. Errors in both merging and fragmentation can be due to either spatial or temporal split/merge or both. The track evaluation system is shown in Figure 2. Evaluation is based on ground truth (GT) data gathered using our annotation tool. The results of our track post processing algorithm are assessed using this evaluation method and are given in the next section of the paper. The outputs of this system were also used to estimate the threshold parameters of the post processing algorithm which will now be described.

Two Phase Track Matching
Two-Way One-to-Many Matching

1. System-Track-Matching – for every system track find all "GT-matches"

"GT-match" = *Temporal-Overlap* and *Spatial-Overlap*
Temporal-Overlap = overlap/(system duration)
Spatial-Overlap = GT centroid inside E1% enlarged system bounding box

If cumulative temporal/spatial overlap < T1, then this system track classified as track false positive (FP)
If multiple GT-matches, this system track has merge error = #matched tracks/system track

2. GT-Track-Matching – for every GT track find all "system-matches"

"System-match" = *Temporal-Overlap* and *Spatial-Overlap*
Temporal-Overlap = overlap/(GT duration)
Spatial-Overlap = system centroid inside E2% enlarged GT bounding box

If cumulative temporal/spatial overlap < T2, then this GT tracks classified as track false negative (FN)
If multiple system-matches, this GT track has fragmentation error = #matched tracks/GT track

Figure 2. Track evaluation method to detect track false positives/false negatives and track fragmentation/merging for system performance measurement and to identify typical errors and their quantitative profiles.

The track post processing method detects track false positives and stitches fragmented tracks together. The detection of track false positives is based on the measurement of several characteristics of each track and computed immediately after the system declares the track to have ended. These features include:

- Average Size
- Average Size at Start/End
- Velocity (magnitude and direction)
- Velocity at Start/End
- Stillness Measure: how stationary at Start/End (depends on full length)
- Area Variance (normalized by average size of object)
- Boolean Flag for Entering/Exiting on border
- Class (Vehicle or Person) & associated confidence

We have found the area variance normalized by the size of the object to be a useful determinant of track feasibility. In addition, objects with low stitching probabilities and which do not appear and exit at the border, are also good candidates as track false positives. Objects which exit in the middle of the image are also tested for stillness which is an acceptable way to "exit" the scene. Using scene context information (such as knowledge of doors and other passage entries in the scene) would be a useful addition here.

These features were also used by the track stitching algorithm. We intend to augment these features with both scene context and normalization information gathered by our object classification system [removed for blind review]. For example, scene context information (probability of object being a certain class at a certain location/time) and normalization information (probability of size at a given location for given object class or for previous size at different location) are useful cues when stitching tracks. People do not change into vehicles and object should not drastically alter their predicted absolute size as they move in front of the camera.

Track post processing is performed in *quasi-real-time*. It cannot be run in real-time since it relies on information which is not yet available. System alarms which need to run in real-time should not depend on track post processing information. However, there are several reasons why it is useful to run the post processing with the video analytic system and not off-line in the event database. The two primary reasons are (1) the video analytics contains more information about each track and is capable of more complex computations; (2) there is a continuous stream of events being ingested by the database and these need to contain the primary results of the system. Search in the database which requires filtering of all or a large portion of the information is impractical.

The solution we have developed involves immediate output from the video analytics followed by updated corrections shortly there-after. In this way, alarms are still run in real-time, the database has preliminary information in real-time, final information is available in quasi-real-time and search from the database does not rely on any intensive filtering.

There is one caveat to this solution which should be mentioned. Our event database includes a video clip of each event. The video record server provides this clip based on the track information provided in real-time by the video analytics. Therefore, tracks which are created by the track post processing method and are composed of multiple tracks because of temporal stitching – require video footage which may not have been recorded. To address this issue, we have added a parameter (see T5 below) to our video record server to always record an additional set period after each real-time track so that in the event of a later track update including stitching the video record will be available.

```
At end of each Track(i)
Set Used(i) = 0
If (FalsePositive(i) > T1)
  Kill(i)
  Return
For all previous alive Tracks j (Used(j) >0)
  If  TimeDifference(Track(i),Track(j)) < T2)
        If Significant Temporal Overlap
          M = TemporalOverlapMetric
        Else
          M = SpatialOverlapMetric
        EndIf
        If (M>T3)
             AddTracksToStitchList(i,j)
             Increment Used counters appropriately
        EndIf
  Endif TimeDifference
  If (Existence of Track(j) > T4)
        If  (Used(j)=0)
                 Kill(j)
                 For all StitchLists(S)
                   If (AllDeadOnStitchList(S))
                      CreateNewTrack
                      For Tracks On StitchList(S)
                        Decrement Used counters
                      End For Tracks  On StitchList(S)
                 End For all StitchLists(S)
        EndIf (Used(j)=0)
     EndIf (Existence of Track(j) > T4)
End For all Track j
```

Figure 3. The algorithm used to stitch fragmented tracks in quasi-real-time for either spatial or temporal fragmentation.

The post processing algorithm used to reduce track false positives and "stitch" tracks in quasi-real-time is shown in Figure 3. Track information is kept "alive" for a set time period after they have ended; (see T3 in the next paragraph). Tracks which are designated to be stitched together into a single track are maintained on a stitch list. In this manner, subsequent tracks can inspect the previous "alive" tracks and determine if they should create a new stitch list or be added to an existing stitch list.

The algorithm relies on several parameters which can be adjusted based on the domain, user needs and limitations:

- T2 - Stitch Time: this is the maximum time between the end of one track and the beginning of another track which are candidates for "stitching"
- T3 -Track Life Time Limit: this is the maximum time a track is "alive" if it is on a candidate stitch list – in order to output the "new merged" track is quasi-real-time.
- T5 – Video Recorder Extra Time: this is the additional time in which the video recorder will record video after a track (all tracks at the time) have ended (usually T5=T2)

The algorithm also relies on the detection of temporal and spatial fragmentation which are computed as follows. The measure of the probability of temporal fragmentation is based on the difference between a first order linear prediction from the last point of the previous track to the first point of the subsequent track. This is weighted by the square root of the average area of the two the tracks (using the end of the previous track and the beginning of the subsequent track.) The measure of the probability of spatial fragmentation is based on cumulative distance between the track centroids during the overlap again normalized by the average area of both tracks during the overlap. Note that overlap is only considered if the overlap represents a sufficient portion (we use 20%) of one or both of the tracks in order to prevent merging of objects which are only crossing or passing by each other.

In addition to post track processing, we have also added the ability for the user of the system to enter polygonal regions which are not of interest. This is a simple and useful mechanism which allows the system to ignore regions which are irrelevant. This is useful for both tracking and alarm detection.

4. Results

The track post processing was performed on sequences for which we have ground truth data. These sequences are from four cameras viewing the parking lot of our laboratory. The sequence, Lab1, was taken during very bright sunlight reflecting on the cars and for which the background subtraction and tracking were very challenging.

We report our results based on our track evaluator. A track is considered to be a false positive if less than T1=50% of its existence is accounted for by the ground truth, i.e., there is overlap with the ground truth bounding box which is expanded by E1=20%. (See Figure 2.) All system tracks which can be sufficiently "matched" to a ground truth track are used to compute the average number of system tracks per ground truth track. This is the measure of track fragmentation. If we eliminate the cases in which ground truth tracks overlap (spatially and temporally), then the optimum value for this metric should be one. However, when camera noise is high or the environmental conditions are extreme (such as extreme brightness, wind or precipitation) then typical values can be as high as 2 or 3 system tracks per ground truth track.

Table 1 shows the results on the Lab1 sequence. The number of false positives and the track fragmentation error rate are significantly reduced. False negative were not measured, but in general when track stitching is successful this can reduce the percent of track false negatives depending on the parameters (T2, E2) from Figure 2. However, there is always a trade-off between track stitching to remove fragmentation error and the possibility of causing over merging.

Figure 4-7 shows the results as output by our backend system which shows a key frame for each track event and the time at which it began. Figure 4 illustrates the track stitching process in which four track fragments of a person walking toward the building are stitched together. Figure 5 shows a false positive due to specular reflection on a car. This false positive was removed by the track post processing.

Figure 6 shows the comparative results on the sequence Lab1 before and after post-processing not including the false positive shown in Fig 5. The sequence begins at the bottom right with a temporal fragmentation as a distant car is tracked, temporarily lost, and then tracked again as it parks. The post processing stitches the two tracks. The next two tracks are a similar situation, but the post processing is unable to stitch the pair because of parameter T3. Too much time elapses while the car parks and is tracked as a stationary object, and the previous track is no longer alive.

In the remaining part of the sequence several other tracks are stitched together, some of which are actually false positives due to shadows. However, this is an effective way to deal with shadows cast by moving objects since the number of "track events" and their time, duration and approximate location and path are still correct.

Table 2 is based on 10 sequences from our laboratory with challenging weather conditions and various viewpoints from four different cameras. This table shows the improvement in the effective performance of the system via the use of (1) removing FN on border, (2) track post processing to remove false positives and (3) the use of regions on non-interest.

8. Conclusions

In this paper, we introduce a procedure for post track processing as part of a modular system in which background subtraction, tracking and post tracking are each self-contained and can be replaced in a plug-and-play fashion. The post track processing can be used to improve system performance by reducing the number of false positives and temporal and spatial fragmentation errors. This procedure is parameterized so it can adapt to different environments, cameras, or user requirements. The system was developed and tested using a track evaluation method and quantitative results are given. The emphasis of the post track processing is on improving effective performance and is based on the practical requirements of the system.

9. References

[Brown 05] Brown, L.M., et al., "Performance Evaluation of Surveillance Systems Under Varying Conditions," IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance, Breckenridge, CO, Jan 2005.

[Hampapur 05] Hampapur, A. et al., "Multi-scale Tracking for Smart Video Surveillance," IEEE Trans. on Signal Processing, Vol 22, No. 2, March 2005.

[Taycher 05] Taycher, L. et al., "Incorporating Object Tracking Feedback into Background Maintenance Framework," IEEE Workshop on Motion and Video Computing, WACV/MOTION 05, Breckenridge Colorado, Jan 2005.

[Leichter 04] Leichter, I. et al., "A Probabilistic Framework for Combining Tracking Algorithms," IEEE Conf. on Computer Vision and Pattern Recognition, Vol 2, p445-451, 2004.

[Lipton 03] Lipton, A. et al., "Automated Video Protection, Monitoring & Detection," IEEE AESS Systems Magazine, May 2003, p3-18.

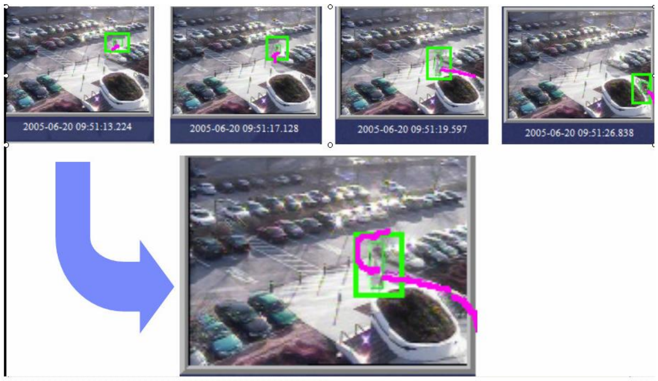

Figure 4. Illustration of track stitching: Top four boxes show four tracks resulting from a person walking slowly toward the building on a very bright day. The bottom image shows the "stitched" track.

Figure 5. Example of false positive due to specular reflection which is removed by track post processing

Sequence	**Track Post Processing**	**Track True Positives**	**System Tracks**	**Track False Positives**	**Fragmentation**
Lab1	No	9	16	4	1.78
Lab1	Yes	9	10	0	1.11

Table 1. Performance results on two sequences before and after track post processing.

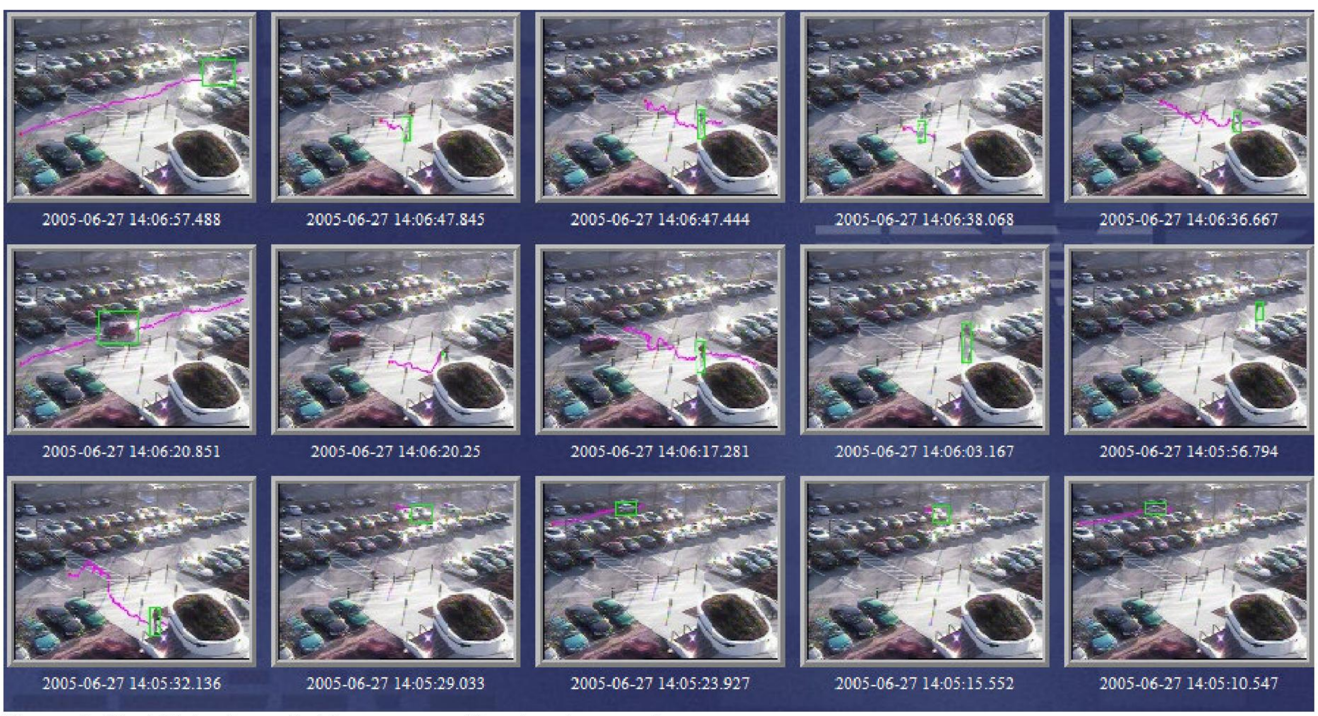

Figure 6. First 15 tracks on Lab1 sequence without post processing.

Figure 7. The 10 "stitched" tracks on Lab1 sequence after post processing.

10 Hawthorne Sequences using MOG	**Probability Of Detection**	**False Alarm Rate**	**Track False Positives**	**Track False Negatives**
Raw – 90TP/45min	87%	8/hour	6	12
Removing FN on border	92%	8/hour	6	7
Removing FN/FP in regions of non-interest	99%	2.7/hour	2	1
Removing FP due to shadows related to track	99%	1.3/hour	1	1
Removing FP due to holes	99%	0	0	1

Table 2. Improvement in the effective performance of our system by (1) removing FN on border, (2) track post processing to remove false positives and (3) the use of regions on non-interest.