

Person and Vehicle Tracking in Surveillance Video

Andrew Miller, Arslan Basharat, Brandyn White,
Jingen Liu, and Mubarak Shah

Computer Vision Lab at University of Central Florida

1 Introduction

This evaluation for person and vehicle tracking in surveillance presented some new challenges. The dataset was large and very high-quality, but with difficult scene properties involving illumination changes, unusual lighting conditions, and complicated occlusion of objects.

Since this is a well-researched scenario [1], our submission was based primarily on our existing projects for automated object detection and tracking in surveillance. We also added several new features that are practical improvements for handling the difficulties of this dataset.

2 Previous Approach

Our previous efforts in automated surveillance led to the development of the KNIGHT system[2], which uses a three-stage process. First, the frame is segmented into foreground regions by maintaining an adaptive model of the background appearance[3], [4]. Second, the correspondence between foreground regions in adjacent frames are found using motion and appearance cues[5]. Third, the objects are classified as either vehicles or people by analyzing a Recurrent Motion Image (RMI) that indicates the periodic motion of a person walking.

3 Illumination Change Detection

In this dataset, abrupt illumination changes frequently occur when clouds move and obscure the sunlight. This causes a large portion of the frame to be mislabeled as foreground. If the illumination change is sufficiently rapid, then the incorrect foreground regions will remain a problem even for a while after the scene reaches a new steady state.

There is a direct tradeoff between the sensitivity to changing appearance because of moving foreground objects and the robustness to change because of a varying background. This balance is controlled by the learning rate parameter. Trying to increase this parameter to account for illumination changes will tend to cause slow-moving or low-contrast objects to disappear into the background.

Our proposed approach to this problem is to detect the occurrence of a global illumination change and temporarily increase the learning rate so that the system recovers quickly once the scene reaches a new steady state. This has the advantage that the system will not suffer in the case of constant illumination, yet a brief change won't cause long-lasting problems.

The challenge in detecting a global illumination change is that they often occur over an inconvenient time interval of about one second, which is too sudden for the background model to update but too gradual to detect using the foreground segmentation image alone. We exploit the observation that such an illumination change will cause a smooth intensity change over a period of 15 to 20 frames and that the change is approximately linear and monotonic. Thus for each pixel, we perform a linear regression of intensity over a temporal window of 20 frames. Pixels with a high correlation coefficient signify an illumination change, while a low correlation coefficient indicates change due to random fluctuation with no monotonic trend. This coefficient is summed over the entire image and then thresholded to trigger a temporary increase in the learning rate. An example of frames and correlation coefficients before and during an illumination change are shown in Figure 1.



Fig. 1. Original frames (left column) and correlation-coefficient maps (right column), before (top) and during (bottom) an illumination change. The illumination change from cloud movement causes an increase in the correlation coefficient of a large number of pixels.

4 Particle Swarm Optimization

Nearly all vision algorithms have several adjustable parameters that affect (sometimes significantly) performance. Although there are often guidelines or rules-of-thumb for setting typical values, these parameters are normally tuned manually - by trial and error - which is tedious and leads to possibly suboptimal results.

Our proposed solution to this problem is to use an automated machine-learning process [6] to choose the best parameter values for a training sequence. The first step is to prepare ground-truth segmentation images for approximately 5 key frames in a training sequence. Then the background subtraction algorithm is automatically run dozens of times with different combinations of parameters, and scored for how well its output matches the ground truth.

Although theoretically any optimization technique could be used to choose new parameter values (such as gradient descent or simulated annealing), we choose Particle Swarm Optimization because it performs consistently well with few iterations and lots of highly nonlinear parameters.

The Particle Swarm Optimization method involves treating each parameter as a spatial dimension. A specific configuration of parameter values is a position vector in this space. A set of about 10 and 50 particles is initialized, each with a position, velocity, and acceleration vector. In each iteration, the background subtraction is run on the training sequence with each of the particles indicating a parameter configuration. Then the particles are updated with swarm motion equations that combines 'cognitive' and 'social' forces to move each particle towards its own best location and towards the globally best location, with added random entropy:

$$v_t = wv_{t-1} + c_1r_1(p_i - x_{t-1}) + c_2r_2(p_g - x_{t-1}),$$
$$x_t = x_{t-1} + v_t,$$

where w is an 'inertial' weight, p_i is the particle's previous best location, p_g is the globally best location of the entire population, r_1 and r_2 are randomly generated noise, and c_1 is the cognitive weight, c_2 is the social weight.

5 Person and Vehicle Classification

The RMI classifier used in KNIGHT depends heavily on precise foreground segmentation and object localization. In order to be more robust with respect to partial occlusion and clutter, we chose to use a classification method based on appearance models instead.

Edge histograms are known as an effective cue for object recognition. In our project, we compute the local edge orientation histogram for each chip using the following steps:

1. Convolution with Sobel filters. Sobel filters are applied to each chip in eight directions as shown in Figure 2. The filter that produces strongest response is chosen as the gradient direction of the pixel.

2. Edge pixels detection. The pixels whose magnitude of gradient is larger τ (in our experiment, it is $0.30 * Gmax$, where $Gmax$ is the maximum magnitude of the edge pixels) are considered as edge pixels.
3. Edge Histogram computation. The edge histogram has eight bins counting the number of edge pixels in eight directions corresponding to the Sobel filters.

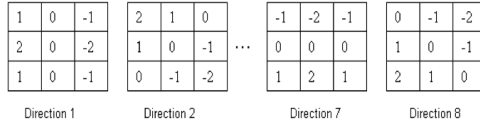


Fig. 2. Oriented Sobel filters



Fig. 3. The examples of features of cars or person. The first row shows there different car chips with similar feature distributions. The second row shows three examples of person chips with their feature plots.

Other than the Edge Histogram, another feature we apply is the aspect ratio, which is the height-to-width ratio of the detected chip. Our observation is that vehicles have a generally wide aspect ratio while people have a narrow and tall aspect ratio.

Finally, each detected chip is represented by a nine-dimensional feature vector, which contains eight dimensions of edge histogram and one dimension of aspect ratio. Figure 3 shows some example of chips with their corresponding features. The edge histogram for cars are strong in several directions, while people have mostly only horizontal edges.

Once feature vectors are constructed, we use a linear Support Vector Machine (SVM) to classify each chip. A training set of chips from people and from cars is initially used to find the ideal hyperplane in this nine-dimensional feature space that separates the car vectors from the person vectors.

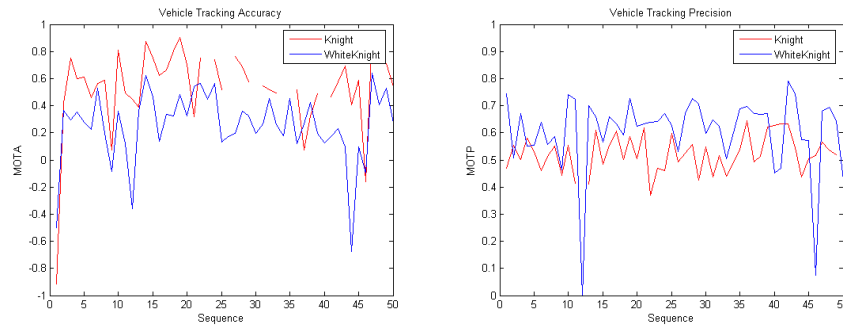


Fig. 4. Vehicle tracking results for Knight (using SVM classification) and WhiteKnight (automatic parameter tuning and illumination change detection). Since neither system is better in both accuracy (left) and precision (right), it will take more experiments to evaluate these changes.

6 Conclusion

We have presented several extensions to our previous system for object detection and tracking. Ultimately we did not have enough time to fully evaluate our new work, so we submitted two separate system results: one with the SVM classification (Knight), and the other with automated parameter tuning and illumination change compensation (WhiteKnight). As shown in Figure 4, WhiteKnight received a higher precision score, but Knight received a higher accuracy score. It will take further analysis and possibly the generation of a ROC curve to arrive at a meaningful assessment of these.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Comput. Surv.* (2006)
2. Javed, O., Shah, M.: Tracking and object classification for automated surveillance. In: *The Seventh European Conference on Computer Vision.* (Denmark, 2002)
3. Javed, O., Shafique, K., Shah, M.: A hierarchical approach to robust background subtraction using color and gradient information. In: *IEEE Workshop on Motion and Video Computing.* (Orlando, 2002)
4. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. *PAMI* (2005)
5. Shafique, K., Shah, M.: A noniterative greedy algorithm for multiframe point correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.* (2005)
6. White, B., Shah, M.: Automatically tuning background subtraction parameters using particle swarm optimization. In: *IEEE International Conference on Multimedia and Expo.* (Beijing, China 2007)