

Model-Free Video Detection and Tracking of Pedestrians and Bicyclists

Yegor Malinovskiy

Department of Computer Science and Engineering, and Department of Civil and Environmental Engineering,
University of Washington, Seattle, WA, USA

Jianyang Zheng & Yin Hai Wang*

Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA

Abstract: *Pedestrian and bicycle monitoring is quickly becoming an avid area of interest as information regarding pedestrian and bicycle flow is needed not only for developing competent access to particular urban corridors and trails, but also for system optimization scenarios, such as transit system operations and intersection controls. In this article, we present a simple, yet effective method for tracking pedestrian and bicycle objects in a relatively large surveillance area, using ordinary un-calibrated video images. Object extraction is accomplished via background subtraction, while tracking is accomplished through an inherent characteristic cost function. Composite objects are used as a means of dealing with occlusions. The algorithm is implemented using Microsoft Visual C# and was tested on numerous scenes of varying complexity, resulting in an average count rate of 92.7% at the specified checkpoints.*

1 INTRODUCTION

Pedestrian and bicycle detection is an important application of Intelligent Transportation Systems (ITS). Automatic detection of pedestrian presence at signalized intersections helps signal controllers decide whether a pedestrian-crossing-green should be initiated. If pedestrians are detected in the process of crossing, appropriate green extension should be allocated so that pedestrian safety can be enhanced and the efficiency

of the intersection signal control can be improved. Based on studies conducted in Los Angeles, California; Rochester, New York; and Phoenix, Arizona, automated pedestrian detection can significantly improve traffic safety (Hughes et al., 2000). At nonsignalized intersections, automated systems can be used to detect pedestrians in the crossing area and activate a warning device for approaching motorists. This passive system is more reliable and convenient for pedestrians, when compared with active devices, such as the push button based systems (Beckwith and Hunter-Zaworski, 1998; SRF and Minnesota DOT, 2003). Automatic detection systems can also provide pedestrian and bicycle volume data that have been identified by the Bureau of Transportation Statistics as one of the most critical data for tracking nonmotorized facility usage trends, developing exposure measures for crash analysis, evaluating levels of service, identifying and prioritizing improvements, and calibrating travel demand models (BTS, 2000).

Recent methods for monitoring nonmotorized activities have been mostly done by using microwave, infrared, or video sensors. Although a microwave sensor offers lower false alarm rates, it tends to miss more pedestrian objects than an infrared sensor (Hughes et al., 2000). To the best of our knowledge, existing microwave and infrared sensors are for pedestrian presence detection only. They are not capable of monitoring pedestrian movements. Video sensors provide opportunities for pedestrian/bicycle detection and tracking. However, such technologies are still under development. Video cameras range in their parameters (resolution and sensitivity), but of primary interest

*To whom correspondence should be addressed. E-mail: yinhai@u.washington.edu.

are applications of low-resolution surveillance-type of video cameras, as they are widely deployed for traffic operations and can be cost-effectively utilized for automated tracking of pedestrians and vehicles.

It is very challenging to track objects using low-resolution video. Although numerous approaches have been attempted, there are essentially three basic means for obtaining and tracking objects in a scene: feature tracking, pattern matching, and background subtraction. Feature tracking is a method that relies on tracking points of high gradient difference, such as corners, joints, or other points on sharp curvature. However, pedestrian objects often lack such stable features, particularly in grayscale video sequences with very low resolution. Pattern matching is a method that depends on a local library of positive or negative examples of objects of interest. The performance of this approach relies highly on the accuracy of the library. Self-learning algorithms exist for building up a probability-based library of objects; however, they are computationally expensive and heavily dependent on manual initialization. Background subtraction is the most commonly used method for obtaining objects in a scene. The scene is monitored for some time and the mode or mean value is taken at every pixel, creating a background image. This background image is then subtracted from the current scene to reveal the objects that are not part of the background. The extracted objects are then tracked based on proximity and other principles. Background subtraction-based approaches are known to have problems when the camera vibrates or environmental conditions change rapidly. They also have false dismissals when the color of the object is sufficiently similar to the background. However, these are outweighed by the relative simplicity and universality of the algorithm, as well as relatively fast computation time.

This article proposes a system, which is designed to determine pedestrian and bicyclist paths and counts in daytime conditions using image sequences from uncalibrated, low-resolution (320×240) grayscale, or color cameras. The algorithm comprises three steps: acquisition of moving objects, object tracking, and object classification. Once these steps are complete, individual pedestrian paths are maintained to identify the regions of high pedestrian activity as well as count individual pedestrians at specified virtual checkpoints. This results in a useful tool for obtaining pedestrian traffic counts and understanding how particular facilities affect pedestrian movements.

2 LITERATURE REVIEW

Traffic data collection through video detection and object tracking is quickly gaining popularity. Video-based

pedestrian tracking is quick, automatic, and inexpensive compared to the manual alternative. Numerous methods have been developed, each with its own strengths and weaknesses.

Model-based techniques are popular, such as those suggested by Heikkilä and Silvén (1999). However, not only must the model library be fairly complete (including all positive and potentially negative samples), a robust scaling algorithm must present to adapt to size changes (Collins, 2003). Although the results are often quite appealing, the creation and upkeep of the model libraries remain as a huge hurdle for practical applications because selection of desired objects is typically a manual process that could be prohibitively costly.

Model-free approaches are much more attractive in terms of upkeep (less managed components) and compatibility, but are generally less robust in terms of object classification. Model-free object classification resorts to classifying by size and/or height-to-width ratio, as demonstrated by Owens et al. (2002). In pedestrian/bicycle detection, motor vehicles and pedestrians/bicycles can be separated by height-to-width ratio and relative size. Although the position and angle of the camera does affect the accuracy of the model-free approach for pedestrian/bicycle detection, the impact is not too great under common settings of traffic surveillance video cameras. Large errors may result from groups of people/bicyclists that form up an object that in size is similar to a car. However, such cases can be dealt with, as will be demonstrated in this article.

Pedestrian/bicycle detection and tracking typically involves three steps: acquisition of moving objects, tracking, and classification (Sheikh et al., 2004; Masoud and Papanikolopoulos, 2001). Acquisition of moving objects can be done through background subtraction, as described by Zheng et al. (2006a) and Zhang et al. (2007). A good quality background image is necessary for background-based detection and tracking approaches. A background image can be extracted by the mode-based approach, an approach where the mode value of each pixel that is observed for some consecutive image frames is taken for the background (Zheng et al., 2006b). Compared to the commonly used median-based approach (one where the median pixel value is taken), the mode-based approach is quick in computation and robust to a broad range of traffic flow conditions. In practice, the mean-based background extraction approach is also utilized when traffic volume is not too high. Although this approach is less accurate, it is much faster because the mean pixel values do not require as much computing power to calculate and update.

As mentioned earlier, an object can be classified by simply comparing its size and height-to-width ratio to some thresholds. However, object tracking is not as

simple. As noted by Jorge et al. (2001), robust occlusion reasoning is the key to any successful tracking system. The key issue in occlusion reasoning after background subtraction is that resulting object blobs tend to leak together (merge) when they get close to each other, making it difficult to track the individual objects. Here an object blob refers to an enclosed region of non-background pixels resulted from background subtraction. Attempts to dissect the merged blobs have been made by Owens et al. (2002), but they do not work well with a fairly common type of occlusion—a momentary total occlusion—such as one that happens when one pedestrian walks in front of another and obstructs the line of sight from the camera. Thus, although a method similar to what suggested by Owens et al. (2002) is used to track individual objects, a different approach should be taken for occlusion reasoning.

In most cases, it is not important to know where exactly in the merged blob a particular object is. Therefore Jorge et al. (2001) suggested to track merged blobs and called these blobs “composite” objects. A composite object may split into several objects. Research efforts have been directed to discern the individual objects once they split. Direction-based approaches have been used to discern pedestrians, as by Jorge et al. (2001). These approaches assume that pedestrians in the scene have an origin and a destination, traveling the shortest possible distance between them, that is, a straight line. That is true in many cases, but does not work well for larger surveillance areas, where pedestrians may change direction abruptly to cross street, or move to attraction spots, such as bus stops or pedestrian crossing buttons. Pai et al. (2004) used a pedestrian model together with the walking rhythm of a human being to conduct pedestrian recognition and tracking. Though the experimental results were favorable, the algorithm requires a side-angle view of the pedestrians and a fairly high image resolution to execute.

At the current state, no single algorithm or system has been widely accepted in pedestrian/bicycle detection and tracking. However, previous studies mentioned above provide valuable insights to our study of pedestrian/bicycle detection and tracking.

3 METHODOLOGY

The bicycle/pedestrian detection and tracking algorithm proposed in this study contains three steps. The first step is acquisition of moving objects. The second step is object tracking that recognizes the same moving object over consecutive frames. This is the most challenging part, because objects may appear on images as merging or splitting as they come closer or get

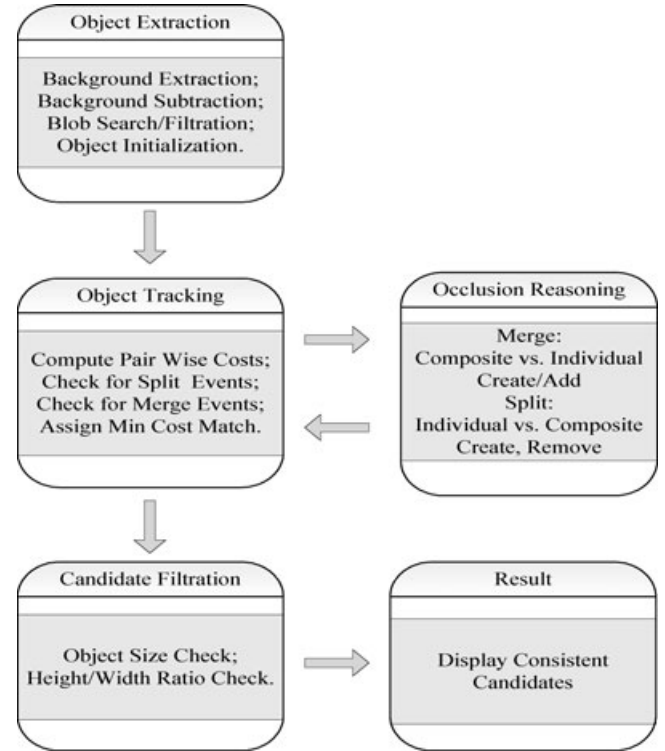


Fig. 1. Flow chart of the proposed algorithm.

further apart. Therefore, occlusion reasoning must be performed when necessary. The last step is object classification; only pedestrians and bicyclists will be assigned ID tags. Figure 1 shows a simplified flow chart of the algorithm.

Background subtraction is used to extract objects from the scene. Tracking is accomplished by matching the size, position, and a 16-bin histogram of the grayscale intensity distribution of the object of interest to those of objects in the next frame, as done by Owens et al. (2002). The 16 bins allow for an even number of grayscale levels in each bin in 256 and 16 bit images. Occlusion reasoning is conducted by tracking and watching for merge and split events. On a merge event, all merged objects are considered a composite object, which is registered and traced until a split event happens. On a split event, any object that split from the composite object is compared to each of the initial objects that formed the composite object to re-identify it. Object filtration is then applied to distinguish pedestrians or bicyclists from other moving objects that may be present in the scene—usually automobiles. Pedestrians/bicyclists are distinguished by their size and the height-to-width ratio. Figure 2 shows an example of the proposed system’s results with identified individual pedestrians in bounding boxes marked with object IDs.



Fig. 2. A simple case of individual pedestrians.

The main advantages of the proposed algorithm are its simplicity and robustness. Simplicity complements robustness—because the algorithm requires no libraries or manual training, it is suitable for essentially any surveillance camera placement with little calibration—only the background difference, height-to-width ratio, size constraints, and background update rate need to be configured before use for optimal detection. However, extensive video testing using identical parameter settings for all test locations also provided favorable results. Details of the algorithm are described step by step as follows.

3.1 Object extraction

The first step of our algorithm is to identify the objects to be tracked. As mentioned before, this is accomplished through background subtraction. The background is extracted using a quick (constant time, as no sorting is required) running average at every pixel. The background pixels are continuously updated to better adapt to the changing background. Other background extraction approaches such as the mode-based method developed by Zheng et al. (2006b) were attempted and received better results, but they were not used in live video testing for the sake of computational speed. The background is comprised grayscale values. Thus if the input pixel is in color, it is first converted to grayscale using the standard luminance formula:

$$\text{GRAYSCALE} = 0.299 * \text{RED} + 0.587 * \text{GREEN} + 0.114 * \text{BLUE} \quad (1)$$

The background image's value at a location is calculated as:

$$B(x, y) = \text{Mean}(M(x, y)) \quad (2)$$

where M is a running average matrix containing previous intensity values present at that location.

Applying the same process to each pixel location creates the entire background image. This method is fast enough for the background image to be updated continuously.

After the background is obtained, it is stored and subtracted from each incoming frame. A threshold Th is applied to the result of the subtraction to create a binary image where the white pixels represent areas of sufficient difference from the background and the black pixels show regions close enough to the background. That is, for a given frame i , the binary image's pixel value at location $\delta(x, y)$, is calculated as follows:

$$\delta(x, y, i) = \begin{cases} 1, & \text{for } |I(x, y, i) - B(x, y, i)| > Th \\ 0, & \text{for } |I(x, y, i) - B(x, y, i)| \leq Th \end{cases} \quad (3)$$

where $\delta(x, y, i)$, $I(x, y, i)$, and $B(x, y, i)$ represent the pixel value at coordinate (x, y) on the binary image, current frame i , and background image, respectively.

Clusters of white pixels that are larger than a user specified threshold are then considered to be moving

Table 1
Tracking parameters

	Parameter	Function
1	(Integer) x	X coordinate
2	(Integer) y	Y coordinate
3	(Integer) prevX	Previous X coordinate
4	(Integer) prevY	Previous Y coordinate
5	(Integer) maxX	Maximum X value
6	(Integer) maxY	Maximum Y value
7	(Integer) minX	Minimum X value
8	(Integer) minY	Minimum Y value
9	(Integer) size	Object size
10	(Integer array) histogram	Object histogram
11	(Integer) frequency	Object age (frames)
12	(Integer) previousSize	Previous size
13	(Object array) compObjs	Composing objects
14	(Boolean) composing	Composing flag
15	(Boolean) composite	Composite flag
16	(Boolean) pedestrian	Pedestrian flag
17	(Double) velX	Velocity in X direction
18	(Double) velY	Velocity in Y direction

objects. The threshold is currently an input value to the system; however, an automatic, dynamic threshold would be best and is being considered for future work. A dynamic threshold would be able to adjust to changing lighting conditions, such as those present in morning and evening periods as well as cloudy days.

3.2 Object tracking

Once the objects are obtained, they have to be tracked from frame to frame. To accomplish this, several parameters are kept for each object of interest. Overall, there are 18 parameters that are preserved as shown in Table 1, but most important of them are position, grayscale intensity distribution, frequency, ID tag, and a list of its composing objects, if any.

The key to tracking objects is to identify the objects present in the current frame in successive frames. This is done by comparing the parameters of the old objects with those of new objects, as recommended by Owens et al. (2002), and finding the minimum cost match. For each object, there is a threshold radius confining the range in which it should appear in the next frame. Thus, the search is only conducted in a localized region around its center. The difference vector of the two potentially matching objects O_i and O_j is calculated as follows:

$$d(O_i, O_j) = (\|A_i - A_j\|, \|H_i - H_j\|, \|W_i - W_j\|, Q) \quad (4)$$

where A_i and A_j are object areas, H_i and H_j are object heights, W_i and W_j are object widths, and Q is the ob-

jects' grayscale histogram difference. The grayscale histogram difference Q is calculated as follows:

$$Q = \sum_{k=0}^{255} \|f_{k,O_i} - f_{k,O_j}\| \quad (5)$$

where f_{k,O_i} and f_{k,O_j} are the frequencies for grayscale value k in the histogram of objects i and j , respectively. To compare the objects, a cost is computed between the current object and each of its potential matches. The cost function is the normalized difference of the parameter vectors of two potentially matching objects, and it is calculated as follows:

$$c(O_i, O_j) = \sum_{n=1}^4 \frac{d_n(O_i, O_j)}{R_n(O_i)} \quad (6)$$

where $d_n(O_i, O_j)$ is the n th element of $d(O_i, O_j)$ and $R_n(O_i)$ is the n th element of $R(O_i)$. $R(O_i)$ is the attribute vector of object O_i , which is calculated as follows:

$$R(O_i) = (A_i, H_i, W_i, G_i) \quad (7)$$

where

$$G_i = \sum_{k=0}^{255} f_{k,O_i} \quad (8)$$

Once the minimum cost match for a current object is found, it is considered to be the same object as it progresses from frame to frame, with exceptions for newly appearing or nearly disappearing objects. Therefore, the tracking of individual objects is accomplished reliably with a low chance of confusion between the objects.

Occlusions are possible problems in a system based on background subtraction, as the clusters of white pixels leak together when the objects they represent get close. Occlusion reasoning is the most important part of any robust tracking system. Because occlusion reasoning can be done at many levels, it is important to know how much information can be extracted from occluded objects and used for occlusion reasoning. For our purposes, we are only interested in the approximate paths of pedestrians and thus it is not imperative to know the exact location of a particular pedestrian inside a group. This key assumption simplifies the problem to one of much more manageable size. Now, the only reasoning that needs to be done is the creation of composite objects—objects that contain several pedestrians, as well as logic for splitting these composite objects into individual pedestrians and bicyclists. That is, when an object–object occlusion occurs, a temporary composite object is created that is known to contain the objects that are involved in the occlusion. Upon the

termination of the occlusion event, that is, when the composing objects separate, the temporary composite object is dissected into the composing parts based on information recorded prior to the occlusion (i.e., size and color distribution).

A composite object is suspected when two or more objects in the current frame have minimum costs associated with the same object in the next frame. If such an event occurs, the sum of the parameter vectors of the potentially composing objects is compared to the larger single object of the next frame. Once the match cost is lower than that of one of the individual object's matches to the larger object, it is considered composite. In that case, a composite object is created as one containing those objects and is tracked as a single object. A split event happens when there are two or more objects in the next scene matching one in the current scene and the sum of those two makes a better match than any individual pairing. In that case, it becomes necessary to find out which of the composing objects split off. This is done by comparing the current split object to ones stored inside the composite object and finding the lowest cost match. Because most occlusions are fairly short, usually involving one pedestrian walking in front of another, it is reasonable to assume that their inherent characteristics change only slightly, thus still allowing a correct match. Very long occlusions that involve a large deformation of the composing objects as well as those that happen outside the field of view and then enter as one object are still issues to be dealt with. An additional restriction is also enforced to assure better matches—objects may split off only one at a time—this reduces the amount of confusion between the composing objects. Because we use a high frequency video input of 30 frames per second, the chance of having more than one object splitting off in the same frame is reasonably low and the overall reduction in ambiguity outweighs the potential error.

3.3 Candidate filtration

Object classification is an important procedure for identifying the correct object types. In this study, objects are classified based on their inherent immediate properties rather than behavioral patterns. The inherent properties in the system include size and height-to-width ratio. An object O_i will be identified as a pedestrian or bicycle if and only if:

$$A_{ThL} < A_i < A_{ThH} \quad (9)$$

and

$$K_{ThL} < \frac{H_i}{W_i} < K_{ThH} \quad (10)$$

where A_{ThL} , A_{ThH} , K_{ThL} , and K_{ThH} are the given thresholds of minimum area, maximum area, minimum height-to-width ratio, and maximum height-to-width ratio, respectively. Presently, the same threshold values are used for pedestrians and bicycles.

Composite objects are absolved of these constraints—as only valid pedestrian objects may combine to become composite objects, and composite objects may grow in size as well as erratically change their height-to-width ratio. Thus, groups of pedestrians do not pose a problem for this simple, yet effective method of classification.

For each identified pedestrian or bicycle, a unique ID is assigned to it. This ID will remain the same during the video stream, including when the tagged pedestrian or bicyclist is occluded by others.

4 TESTS AND RESULTS

The proposed algorithm has been tested under live field conditions, using a COHU i-Dome surveillance camera, as well as footage obtained by ordinary consumer products and intersection surveillance cameras. Operating on a 1.83 GHz Centrino Duo processor laptop, the system is able to process live video signal for extensive periods of time with minimal deterioration in detection accuracy. Coupled with a camcorder and a tripod, the system can be used at almost any location with a good vantage point to provide live automatic pedestrian/bicycle counting and local route choice information (within the surveillance area) for a fraction of the cost of manual counting.

It has been a challenge to thoroughly test video tracking systems, as it is impossible to predict all types of situations and scenes that may be encountered during real-life exploitation of the tracking system. Often an algorithm is developed using a limited number of video sequences as test subjects and works well for those particular sequences, yet when it is applied to another common situation that is still in scope, the results are less encouraging. In an attempt to avoid this issue, a library of synthetic tests was created for common merging/splitting scenarios, as well as noise and distraction events. Figure 3 shows the examples of merging and splitting in synthetic tests. The progression of a merge event is shown in Figure 3a. The leftmost frame of Figure 3a shows four separate objects. Objects “1” and “2” then combine to form a composite object “1/2,” as shown in the middle frame of Figure 3a. The rightmost frame of Figure 3a shows the result of the addition of object “4” to composite object “1/2,” resulting in the composite object “1/2/4.” Figure 3b shows the continuation of the same sequence

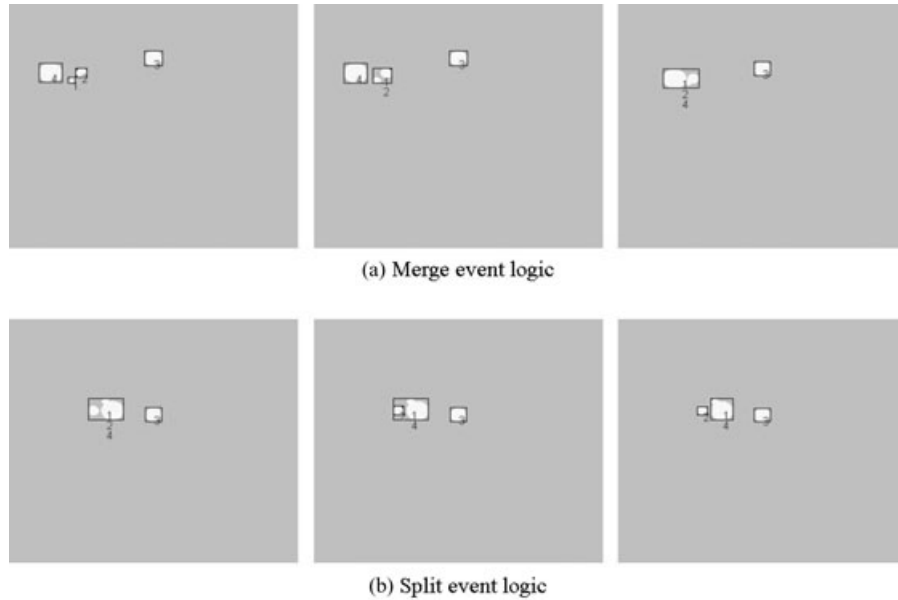


Fig. 3. Merge and split event logic using synthetic test.

in which a split occurs, dividing object “1/2/4” into objects “2” and “1/4.” These types of tests on synthetic objects are specific enough to cover most of the events that can occur and general enough to represent any camera placement situation in a large surveillance area—that is, one where perspective changes are minimal. There are only a few types of cases that can happen between the pedestrian objects—merge/split of two single objects, merge/split of two composite objects, or a merge/split of a composite and a single object. All these cases were tested with simulated events. Other tested events involved noise or distractions—such as a momentary appearance/disappearance of an object. We tested our algorithm with salt and pepper (random minor noise) and randomly generated distraction objects (illogically placed temporary objects). The algorithm performed very well in various synthetic tests conducted.

After establishing a competent library of these synthetic tests and ensuring the system’s performance was adequate in the tests, real video was used for further testing. Although the algorithm performed well in synthetic tests, tests with real-world video data are still necessary due to the following reasons: (1) a real-world object with motion looks different from frame to frame in two-dimensional images; (2) there are various disturbance types other than the simulated random noises; (3) in order for this system to be practical, it has to be sufficiently tested under possible environmental conditions to understand the abilities and limitations of the approach.

4.1 Tracing

Because the system can be potentially used for tracing object movements, the tracing function of the system was also tested. Tracing paths of individual bicyclists and pedestrians is an interesting application, as it allows the creation of density or activity maps of the targeted objects in observed regions. The tracing occurs at the bottom of every object, thus being as close to the actual feet or wheels as possible. Checkpoints (registration lines) can be created to obtain pedestrian counts at desired locations in the surveillance areas (crosswalks, entries, corridor segments, etc.). Several checkpoints can be created in the same corridor to ensure accurate results. The maximum value obtained by any of the checkpoints should be taken because a continuously tracked pedestrian or bicycle can only be counted once at a checkpoint while they have not left the scene.

The system was first tested in a simple urban trail environment involving a straight corridor and a low-volume road intersection. The camera was placed on a bridge over the Burke-Gilman urban trail near the University of Washington (UW). Two registration lines were then set up to count movements in and out of the intersection. Figure 4 demonstrates the setup and the resulting paths. White and gray lines show the traces of pedestrians (white means currently active). The black lines are the registration lines (checkpoints) for pedestrian/bicycle counting, with the counts at each line shown in white. This location had little vehicular traffic, and hence the pedestrian and bicycle movements were predictable and the viewpoint creates few



Fig. 4. Burke-Gilman trail.

occlusions. Also, as a commuting route, there were few inter-object occlusions as there were few groups present and most people commute alone. Bicyclists avoided riding close to one another as well as pedestrians. Because of the above conditions and overcast weather conditions, which dissolved many of the shadows, this was considered “low” scene complexity.

“Medium” scene complexity can be seen in Figure 5. Motorized transit was present. Also occlusions from static objects, such as light poles, were frequent. A few false detections can be noted, such as a path fragment left by a scooter in the vehicle lane. Scooters fit the size and dimension criteria and can cause overcounting, as they can be falsely taken for a cyclist or pedestrian. This is a difficult problem to deal with on shared roads, as there are few ways to distinguish a motorized scooter from a bicycle, without resorting to potentially misleading speed thresholds.

Finally, “high” scene complexity can be observed in Figure 6. The UW Hub Bus Stop presented many challenges, as pedestrians no longer had predictable trajectories and could remain stationary for long periods of time. Buses also created nonstatic occlusion problems by stopping and potentially concealing pedestrians or bicyclists behind them. Shadows caused by trees were present, although the overall shadow presence was low. Glass reflection was also an issue, as the camera was placed inside a building and looking out of the 4th story window. The side-firing camera perspective resulted in numerous occlusions with passing vehicles. The registration line was often crossed by nonpedestrian or bicycle objects, thus candidate filtration was much more

crucial. Finally, the number of pedestrians present in the area would grow quickly during class breaks and grouping behavior was much more prevalent.

4.2 Tracking

Another type of use for this system is to collect individual pedestrian/bicyclist crossing behavior data. Figure 7 shows an example application in an intersection crosswalk using the regular surveillance camera already in place. An ID tag was given to each pedestrian and his/her movement through the intersection was individually tracked. Surveillance camera positions often result in slightly side-firing views and suffer from a high occlusion rate. In Figure 7, a merge event had occurred. Pedestrians “1” and “2” leaked together and formed a composite object. Because of the camera’s setting angles, pedestrian occlusion is a major problem at this test site. A common occlusion event is shown in Figure 8, in which pedestrian “2” passed pedestrian “1,” creating a temporary occlusion. As a result of the occlusion, a composite object was created containing both pedestrians, as can be seen in Figures 8b and c. Once pedestrian “2” split off the composite object, only pedestrian “1” was left—therefore the composite object turned into a regular pedestrian with ID 1, as can be seen in Figure 8d.

In another test site, the camera was set about 60 ft (18 m) above the ground.

Pedestrians appeared much smaller at this site than those in test site one. This created some challenges for pedestrian detection and tracking because the object



Fig. 5. University bridge.



Fig. 6. UW Hub Bus Stop.

size was small and features of objects become similar. In fact, there were some cases where the occlusion reasoning fails, as can be seen in Figure 9. In the frame on the left, a group of three pedestrians was mistaken for a group of two, as two of the pedestrians in the group walked into the scene together. The frame on the right shows that a momentary split of the group allowed tracking of the correct number of pedestrians in the group, yet the labeling was incorrect. The newly found pedestrian was labeled 5, while pedestrian “2” was lost—too much time has passed since pedestrian “2” had appeared as an individual—thus when it appeared again, its appearance was too different to ensure a successful match.

4.3 Crossing times

Crossing time calculations are also a potential use for our system. The times are derived by recording timestamps when each individual trace (path) reaches the origin and destination registration lines. The timestamps are then compared pairwise to determine the travel time for each pedestrian/bicyclist. Because paths are drawn at the bottom of the objects, the crossing times are not severely impacted by projection error.

Figure 10 demonstrates a 1-second resolution histogram of time it took for 82 pedestrians and bicyclists to negotiate a crossing section of the Burke-Gilman corridor, defined by the two registration lines. There

are two peaks, suggesting that there are two different groups present at this location. This is consistent with the mode characteristics of this particular corridor, which is almost exclusively used by bicyclists and pedestrians. Furthermore, it can be seen that the slower group comprises about 15% of the total, which is indicative of the actual mode split of 17.1% pedestrian (14 of the 82 persons were pedestrians, as verified by manual inspection). Although this method can be used to distinguish the two groups statistically, it may not be correct for certain cases. For example, a bicyclist can be going as slow as a pedestrian, and a person may be running as fast as a bicyclist. This does, however, present a rough distribution of mode choice on this segment of the corridor. Speeds can be easily calculated from the travel times if the distance between the two registration lines can be measured—in our case the lines were drawn on easily identifiable markers—the poles denoting the entrance to the crosswalk. The accuracy of the speeds obtained will largely depend on the interval chosen—the longer the interval, the more accurate the speed estimation will be. In-field measurement is not always an option, as some locations are inconvenient to reach or taped video may no longer represent the markers present. In that case, certain known distances can be used to estimate the desired parameters. Automatic calibration is also a potential improvement of the system and would begin by identifying objects of known proportions.



Fig. 7. Sequence showing merge event.

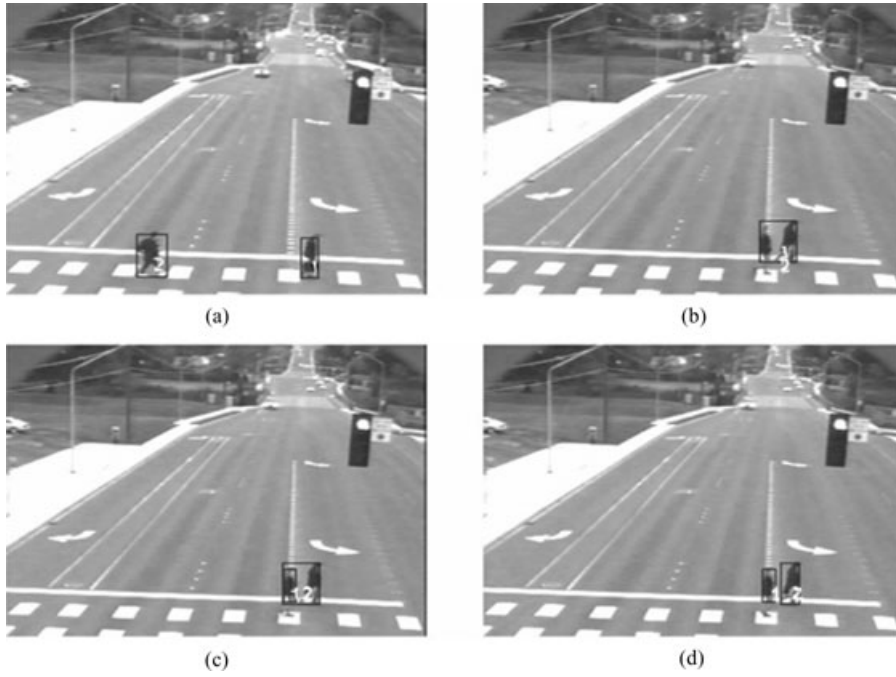


Fig. 8. An occlusion occurs while a pedestrian passes another.



Fig. 9. Occlusion reasoning failure.

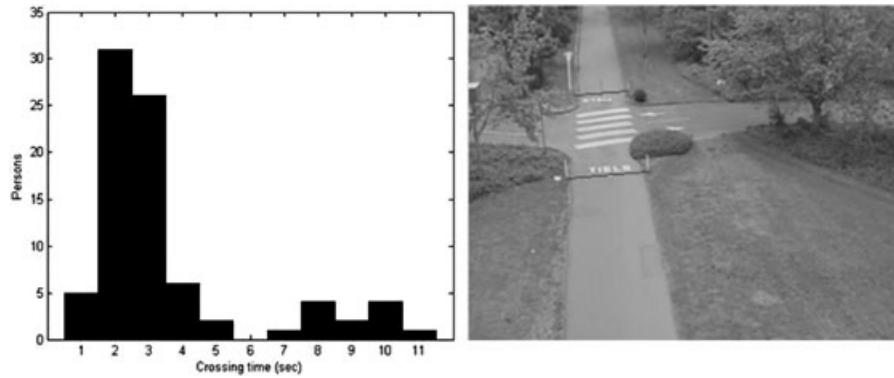


Fig. 10. Crossing time histogram (left) at the Burke-Gilman trail (right).

Table 2
Summary of counting test results

<i>Location</i>	<i>Test length</i>	<i>Scene complexity</i>	<i>Ped. density</i>	<i>Manual count</i>	<i>Automatic counts</i>	<i>Missed counts</i>	<i>Overcounting</i>	<i>Accuracy (%)</i>
Burke-Gilman Line 1	10 min	Low	4.9	49	46	3	0	93.88
Burke-Gilman Line 2	10 min	Low	4.9	49	46	3	0	93.88
University Bridge Sidewalk	10 min	Medium	0.6	6	7	0	1	83.33
University Bridge Bike Path	10 min	Medium	0.5	5	5	0	0	100
UW Hub Bus Stop	10 min	High	2.7	27	24	3	0	88.89
Overall	30 min		4.5	136	128	9	1	92.65

4.4 System accuracy

Table 2 summarizes the resulting counts and accuracies of the system under various conditions. The automatic counts were generated by registration lines. Pedestrians and bicyclists were also manually counted as ground-truth data. The results show that the scene complexity affects the accuracy to some degree, but the results remain reasonably accurate. The overall test accuracy, calculated as $100 - 100 * ((\text{Overcounting} + \text{Missed Counts}) / \text{Manual Counts})$, was about 92.7%. Overcounting was generally low. The primary reasons for overcounting include moving scene objects such as trees or incomplete vehicle silhouettes formed due to occlusion or “ghosting”—a side effect of the average background method, where a faded version of a temporary object appears as part of the background if the object remains stationary.

5 CONCLUSIONS

Pedestrian and bicycle detection and tracking are important issues for traffic operations and planning. Of the several technologies developed for pedestrian and bicycle detection and tracking, video-based systems have attracted more attention in the past several years due to the wide deployment of video cameras for traffic surveillance. In this study, a simple, yet effective algorithm was developed for pedestrian and bicycle detection and tracking with low-resolution video cameras. The algorithm comprises three steps: object extraction, object tracking, and candidate filtration. This algorithm can trace individual pedestrian/bicyclist paths, and therefore provides pedestrian counts, the regions of high activity, as well as crossing time. Speed estimations can be given with knowledge of the distance between the registration lines.

The system was tested using synthetic, live, and taped video data to assure compatibility and measure performance. Video samples collected from numerous test

locations were applied to test the performance of the system. The overall average count accuracy was 92.7% under a range of scene complexities, with little calibration. The system can be used for obtaining pedestrian/bicycle counts and paths in a variety of environments, given that the pedestrian/bicycle traffic is not too high—such that individual pedestrians/bicyclists would be spotted apart from a group for at least a few frames.

The system provides a solid base for further refinements to improve overall accuracy. In particular, further work will be done to allow longer occlusions as well as refinements in the object classification step of the algorithm. More complicated scenes with severe shadows and complex lighting have to be dealt with via shadow subtraction and variable threshold techniques. Higher pedestrian and bicycle flows have to be dealt with through feature-based object extraction techniques. Distinction between pedestrians and bicyclists can also be achieved through harmonic motion filtering. With the above improvements, the system would no longer be as restricted to certain lighting conditions and would be much more accurate in crowded areas, allowing the observation of busy city intersections and shopping malls. Although there is much room for improvement, we believe that the current system is a useful prototype tool that, if used under a few reasonable constraints, can generate important data about bicycle and pedestrian behavior automatically.

REFERENCES

- Beckwith, D. M. & Hunter-Zaworski, K. M. (1998), Passive pedestrian detection at unsignalized crossings, Transportation Research Record: Journal of the Transportation Research Board, No. 1636, 96-103, National Research Council, Washington DC.
- Bureau of Transportation Statistics (BTS). (2000), *Bicycle and Pedestrian Data: Sources, Needs, and Gaps*, Report No. BTS 00-02. Bureau of Transportation Statistics, U.S. Department of Transportation.
- Collins, R. T. (2003), Mean-shift Blob Tracking through Scale Space, in *Proceedings of the 2003 IEEE Computer Society*

- Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Vol. 2, pp. II 234–40.
- Heikkilä, J. & Silvén, O. (1999), A real-time system for monitoring of cyclists and pedestrians, *Second IEEE Workshop on Visual Surveillance (VS'99)*, 74–81.
- Hughes, R., Huang, H., Zegeer, C. & Cynecki, H. (2000), Automated pedestrian detection used in conjunction with standard pedestrian push buttons at signalized intersections, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1705, 32–39, National Research Council, Washington DC.
- Jorge, P. M., Abrantes, A. J. & Marques, J. S. (2001), Automatic tracking of multiple pedestrians with group formation and occlusions, in *IASTED International Conference on Visualization, Imaging and Image Processing*, Marbella, Spain, 613–18.
- Masoud, O. & Papanikolopoulos, N. P. (2001), A novel method for tracking and counting pedestrians in real-time using a single camera, *IEEE Transactions on Vehicular QI Technology*, **50**(5), 1267–78.
- Owens, J., Hunter, A. & Fletcher, E. (2002), A fast model-free morphology-based object tracking algorithm, in *Proceedings of British Machine Vision Conference*, Cardiff, UK, 767–76.
- Pai, C., Tyan, H., Liang, Y., Liao, H. M. & Chen, S. (2004), Pedestrian detection and tracking at crossroads, *Pattern Recognition*, **37**(5), 1025–34.
- Sheikh, Y., Zhai, Y., Shafique, K. & Shah, M. (2004), Visual monitoring of railroad grade crossing, in *Proceedings of SPIE-Int. Soc. Opt. Eng. (USA)*, **5403**(1), 6546–60.
- SRF Consulting Group, Inc. and Minnesota Department of Transportation. (2003), *Bicycle and Pedestrian Detection: Final Report*, FHWA, U.S. Department of Transportation and Minnesota Department of Transportation.
- Zhang, G., Avery, R. P. & Wang, Y. (2007), A video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1993, 138–147, National Research Council, Washington DC.
- Zheng, J., Wang, Y., Nihan, N. L. & Hallenbeck, M. (2006a), Detecting cycle failures at signalized intersections using video image processing, *Computer-Aided Civil and Infrastructure Engineering*, **21**(6), 425–35.
- Zheng, J., Wang, Y., Nihan, N. L. & Hallenbeck, M. (2006b), Extracting roadway background image: a mode-based approach, *Transportation Research Record: Journal of the Transportation Research Board*, No. 1944, 82–88, National Research Council, Washington DC.