Pedestrian Detection for Driver Assist and Autonomous Vehicle Operation using Offboard and Onboard Sensing

Paulo Borges, Robert Zlot, and Ashley Tews
Autonomous Systems Laboratory
CSIRO
Brisbane, Australia
Firstname.Lastname@csiro.au

Abstract

Situational awareness for industrial vehicles is crucial to ensure safety of personnel and equipment. While human drivers and onboard sensors are able to detect obstacles and pedestrians within line-of-sight, in complex environments initially occluded or obscured dynamic objects can unpredictably enter the path of a vehicle. We propose a safety system which integrates a vision-based offboard pedestrian tracking subsystem with an onboard localisation and navigation subsystem. This combination enables warnings to be communicated and effectively extends the vehicle controller's field of view to include areas that would otherwise be blind spots. A simple flashing light interface in the vehicle cabin provides a clear and intuitive interface to alert drivers of potential collisions. We implemented and tested the proposed solution on an automated industrial vehicle to verify the applicability for both human drivers and under autonomous operation.

1 Introduction

Around worksites populated by vehicles and people, there is a safety concern regarding path conflicts. A common area is where the vehicle operators and pedestrians are obscured from each other in 'blind spots'. An illustration of this situation is given in Figure 1, where a person who is out of the driver's view is walking towards the road. While the example shown is in a typical outdoor scenario, the problem also occurs in industrial sheds where people and vehicles operate in shared areas around infrastructure. Typical approaches to reducing the problem of collisions in blind spots include making staff aware of shared-area operations through site- and area-specific rules and inductions, drivers sounding horns when approaching, entering or leaving a building, vehicle rotating lights and beepers, and using large convex mirrors placed at traffic intersections. Onboard cameras can also be used to augment a driver's vision of the areas around the vehicle. While many of these approaches may be common practice in industrial environments, the possibility of a vehicle-pedestrian collision still exists. For autonomous vehicles operating around worksites, this problem must also be addressed. Our research is motivated from the human-driven and autonomous vehicle perspectives.

Figure 1: Potential collision example. The pedestrian being tracked (in red) is occluded by the building, from the truck perspective.

For this research, our test vehicle is the Autonomous Hot Metal Carrier (HMC) which is capable of human-driven and autonomous operations. HMCs are used to deliver 10 tonne crucibles of molten aluminium (over 700°C) between sheds at a smelter 24 hours a day, 7 days a week. People and other vehicles are also operating in the transfer sheds and there is a substantial potential for unwanted interactions. We have automated an HMC and demonstrated typical operations over hundreds of hours of autonomous missions [Tews *et al.*, 2007].

In this paper, we describe our system that uses offboard cameras located around blind-spots to detect pedestrians and alert the vehicle operator where the vehicle operator is considered to be either a human or the autonomous controller. For manned vehicle operations, the alert system consists of signal lights mounted on either side of the inside of the cabin. Alternative display methods could also be considered including audio or onboard displays. The signal presented to the operator is at a frequency corresponding to the likelihood of a collision and the proximity of the pedestrian. For autonomous vehicle operations, the signal is used by the onboard navigation system to enable to vehicle to react accordingly.

The main system components are: offboard pedestrian detection, pedestrian tracking, and collision detection systems. Pedestrian detection and tracking is a common area of research in surveillance and safety [Gandhi and Trivedi, 2007; Ran *et al.*, 2007]. Typically, surveillance systems are deployed using cameras mounted to infrastructure that can monitor areas for pedestrian activity [Ran *et al.*, 2007]. Collision detection safety systems commonly consist of onboard cameras used to detect pedestrians to avoid unwanted interactions [Gavrila and Munder, 2007; Caminiti *et al.*, 2010].

Many similar systems use cameras onboard vehicles (usually cars) or monitoring systems at traffic intersections to assist with pedestrian/vehicle warning systems. However, most do not use offboard systems to supplement a vehicle operator's field of view beyond line-of-sight and provide onboard signals of potential pedestrian/vehicle path conflicts. With the systems that do send some sort of alert to the vehicle, they do not change the vehicle behaviour in case that risk is high. A similar system to the one described in this paper was proposed by Aycard et al. [2006], where cameras are mounted on a bus station and parking lot to alert drivers. Our system, in contrast, is applied to industrial environments in which onboard localisation is currently implemented. More importantly, the system also applies to autonomous navigation, such that when a pedestrian is in a potential collision zone the vehicle slows down or stops. Such a support system is essential for autonomous operations, especially when large vehicles are considered.

This paper is organised as follows. In Section 2 we provide additional details on the set-up of the proposed system. In Section 3 we explain the video analysis module, which includes the pedestrian tracking and classification. In Section 4 we describe the proposed algorithm for estimating the risk of collision. We present experimental results in Section 5 followed by relevant conclusions in Section 6.

2 System Description

The offboard subsystem is comprised of cameras mounted on infrastructure around a worksite such that they cover pedestrian areas that may be fully or partially obscured from a vehicle operator's view. The cameras also cover areas outside of the field of view of additional onboard sensors, such as ranging sensors (laser, radar, etc.) or vision-based systems for cars [Enzweiler and Gavrila, 2009]. Onboard the vehicle, a localisation system is required to estimate the vehicle position, and control may be achieved either through software or a human operator.

Hazardous scenarios may arise as pedestrians emerge into or near the vehicle's intended path. To detect these situations, images from the camera are used firstly to identify any pedestrians in the scene and secondly to determine if the path of any pedestrians might intersect with the vehicle's path. The vehicle's path and its uncertainty is predicted based on the localisation system and current state (velocity and steering angle) or path plan in the case of autonomous control.

The vehicle operator may be a human driver or an autonomous system capable of controlling the vehicle [Tews *et al.*, 2007]. In the case of autonomous navigation, the speed of the vehicle can be reduced according to the estimated risk. For human operators, an in-cabin interface alerts the driver by flashing a light on the side of the predicted potential collision at a rate proportional to the perceived risk. The decision for choosing a simple flashing light system was based on careful consideration of several factors including: ensuring the interface is intuitive; minimising distraction of the drivers and clutter within the cab; and limiting the complexity of installation and maintenance. We further considered the location, colour, intensity, and flash frequency of the lights to ensure a minimalist and ergonomic integration into the cabin infrastructure.

The main components of the system are illustrated in Figure 2. These include:

1. The imaging device(s), which can be visible spectrum or infrared cameras.
2. The video analysis module, which performs the vision-based pedestrian tracking and classification.
3. The vehicle localisation and control system. The control can be a human operator or an autonomous navigation system.
4. The collision prediction module.
5. The signalling system, such as warning lights.

There are several alternatives as how to implement these components. The next sections provide details on the algorithms we use for the video analysis and collision prediction.

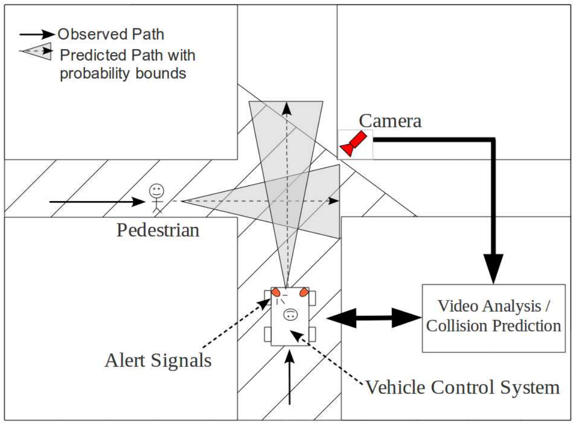

Figure 2: Diagram illustrating system concept.

3 Video Analysis

The video analysis consists of blob tracking and pedestrian classification, followed by a perspective transformation of the pedestrians to the coordinate frame of the vehicle. In this section we provide details about each of these parts in our implementation.

3.1 Tracking

Efficient visual tracking is essential for the system to work satisfactorily. In the following we describe the tracking algorithm used, which can be divided into three main stages: background modeling, foreground blob detection, and blob tracking. A block diagram of the process is shown in Figure 3.

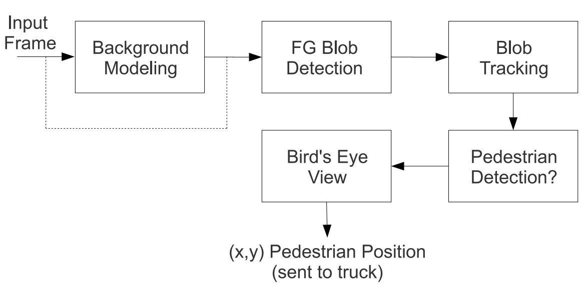

Figure 3: Block diagram of the video tracking process. The dotted line indicates that the background modeling can be a one-off operation when the algorithm is initiated or it can be adaptive.

Background Modeling

As the proposed system is based on static cameras, the algorithm assumes a relatively static background for the foreground segmentation. Foreground segmentation consists of identifying moving objects by selecting parts of the image which differ significantly from a background model, according to some metric such as luminance or color difference. The background model corresponds to a statistical representation of the background scene, generated after observing a given number of frames, which ideally contain no moving elements.

In its simplest form, background segmentation can be achieved by averaging a number of frames to generate the model and subsequently labeling as foreground any pixels that exceeds a given difference threshold. However, in outdoor environments, scenes can be complex, with wavering trees and moving clouds, for example. In this case, a more robust algorithm is necessary for the segmentation. In our case, we use the algorithm proposed by Li *et al.* [2003], which has the capability of processing relatively complex backgrounds. The technique is based on pixel color and co-occurrence statistics. The pixel color and the color co-occurrence distributions are represented by histograms, which serve as basis for the Bayes decision rule to classify pixels to foreground or background. The algorithm is relatively robust to gradual changes as well as abrupt changes.

Foreground Blob Detection

The goal of this stage is to separate "noise blobs" from relevant objects that are to be tracked. It uses as input the foreground segmentation performed in the background modeling stage. For this task, we use the technique proposed by Senior et al. [2006]. From the foreground segmentation mask, a connected component operation is performed to merge neighboring blobs and a size filter is used to remove small components. From frame to frame, blobs are tracked according to a spatial overlapping rule. If the blob is tracked successfully across a given number of frames, it is added to the tracked blob list, and it is then passed into the advanced tracking stage, described next.

Blob Tracking

In the blob tracking stage, a Kalman filter is used to predict the position of the blob in the next frame. If there is no predicted overlap among any of the blobs in the frame, the tracking is based simply on the connected-component analysis. If overlapping is predicted, the tracking is based on a mean-shift tracker [Comaniciu and Meer, 2002] and a particle filter [Nummiaro *et al.*, 2002].

3.2 Pedestrian Detection

Once a relevant (*i.e.*, non-noise) blob is tracked, we classify it as pedestrian/not-pedestrian by using motion and blob shape features [Johnsen and Tews, 2009]. The features are combined according to a support vector machine (SVM) classifier for the final decision. An alternative for the detection is the use of trained boosting classifiers, which can also detect pedestrians that are not moving. The position assumed for the pedestrian is the bottom of the blob. This position ideally corresponds to the feet of the pedestrian, although noise and shadows can cause some shift to it. In practice, this shift is often small and does not significantly affect the overall system performance, as discussed in Section 5.

3.3 Perspective Transformation

Once we have a pedestrian blob being tracked, its pixel coordinates must be transformed to the coordinate frame of the on-board navigation system. For this task, we determine the homography matrix $\mathbf{H}$ describing the projection from the camera plane to the ground plane, or a perspective transformation [Hartley and Zisserman, 2004]. It can obtained by selecting 4 points in the source and destination images and solving for $\mathbf{H}$ in a linear system framework. $\mathbf{H}$ is a 3×3 matrix such that

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

where x and y are the pixels coordinates in the source image and x' and y' are the pixel coordinates in the destination image. The x' and y' coordinate frame corresponds to a top-down "bird's eye view" of the site. The images in Figures 4a and 4b illustrate this coordinate transformation from the camera plane to the ground plane, indicated by the red dots. This information is then sent to the next module, where the collision prediction analysis is performed.

4 Collision Prediction

In order to predict whether a collision or near collision might be likely, both the path of the vehicle and the states of recently observed pedestrians are required. In the case of an autonomous vehicle, vehicle path prediction is typically straightforward since it can be obtained from the path planner or given knowledge of the navigation algorithm. However, for human-operated vehicles, the intended path must be estimated based on available state information such as current pose, velocity, steering angle, and knowledge of the environment. Since we assume that there are no additional onboard sensors capable of measuring or mapping the environment available, we predict that the immediate path of the vehicle will approximately follow an arc starting at its current position and maintaining a constant steering angle and velocity for a fixed amount of time. If a prior map is available, a more accurate estimate can be obtained using a path planning algorithm when in the vicinity of site infrastructure.

Because pedestrian motion can be relatively unpredictable, we consider recent pedestrian observations and incorporate uncertainty in future locations into an adaptive threshold in the collision prediction stage. The locations of pedestrians observed by the vision system over a fixed time interval are compared to the predicted path

(a)

(b)

Figure 4: Illustration of the tracking (a) and its corresponding bird's eye view (b) in the coordinate frame of the vehicle on-board localiser.

using a nearest neighbor search. A *kd*-tree [Arya *et al.*, 1998] is constructed using the 2D pedestrian locations to enable efficient fixed-radius nearest neighbor searches[1]. Samples are taken along the predicted vehicle trajectory at regular intervals and each sample point is used as a query for a nearest neighbor search which returns the nearest pedestrian point if within the specified fixed radius. To account for uncertainty, the search radius increases with distance further from the vehicle. A more accurate motion model for pedestrians could also be incorporated to predict the behavior of pedestrians in the future or downweight older observations.

In order to identify potential collisions on both sides of the vehicle to correspond to the two cabin indicator lights, two separate *kd*-trees are constructed: one for

[1]In our implementation we use the ANN library available from http://www.cs.umd.edu/∼mount/ANN.

pedestrians observed on the right side of the vehicle (relative to the current position and orientation), and one for pedestrians on the left. Both kd-trees are searched at each trajectory sample starting from the current vehicle position and moving forward in time. If a potential collision is detected, the cabin light corresponding to the side of the detected pedestrian is flashed at a rate inversely proportional to the proximity of the pedestrian to the vehicle (capped at a maximum rate of 20 Hz). No further searches are required on additional trajectory samples from that side of the vehicle, though the algorithm proceeds with the other side as it is still critical to communicate the hazards on both sides of the vehicle to the operator. The collision prediction algorithm is illustrated in Figure 5.

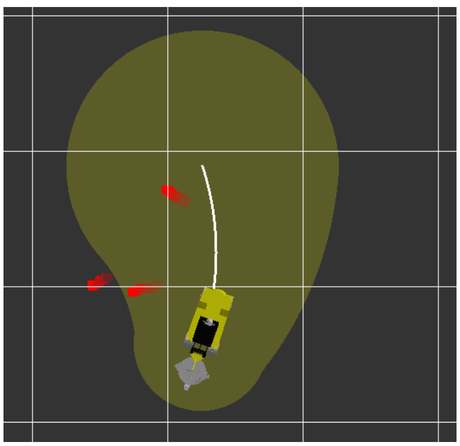

Figure 5: A snapshot of the collision prediction algorithm. The predicted path of the HMC is shown as a white arc. The recent pedestrian positions (observed by the vision system) are indicated in red (older observations are faded implying the trajectories). The yellow region is the collision zone within which the vehicle indicator lights will be activated. In reality, the algorithm would preempt the search upon identifying the nearest pedestrian approximately 3.5 m from the front left corner of the HMC. In the illustrated scenario, two of three observed pedestrians are within the collision zone on the left side of the vehicle; therefore the left indicator light would flashing in the vehicle cabin. The grid lines are spaced at 10 m intervals.

5 Experiments

In this section we present a practical implementation of the full system, illustrating its applicability in a real industrial scenario.

The onboard localisation system [Tews *et al.*, 2007] discussed in Section 2 is implemented on an automated HMC, a 20-tonne forklift truck, shown in Figure 1. This localisation system has been extensively tested in vehicles in a real smelter environment, presenting dependability and good precision - in our test environment, the average error is less than ten centimeters. As the forklift has autonomous navigation capability, we perform tests in both manual driving mode and autonomous driving mode. As described in Section 2, in manual driving mode an alert signal is sent to the driver indicating whether there are pedestrians in the area that can potentially be on the truck route. For signalling, we mount two flashing lights inside the cabin, as illustrated in Figure 6a. Notice that there are two lights, one mounted on the right side and one mounted on the left side of the cabin. Whenever a potential collision route between the vehicle and a nearby pedestrian is detected, these lights flash inside the cabin indicating to the driver from which side the risk of collision has been identified. The flashing frequency is a function of the absolute distance between the vehicle and the pedestrian and the risk of collision. A possible alternative or complement to the lights could be a beeping signal whose beeping frequency is synchronized with the light flashing frequency, or map display showing the locations of pedestrians similar to Figure 4.

5.1 Practical Parameters and Considerations for the Video Analysis

We utilize video streams obtained from different cameras and views which are permanently located across our industrial site in Brisbane, Australia. As an example, the approximate locations of two views are shown in Figure 7, represented by the red camera icons. Their corresponding images are given in Figures 1 and 4a for cameras 1 and 2, respectively. The resolution of the images is 768×576. The video frame rate varies slightly due to network oscillations, ranging from 10 to 12 frames per second. This type of oscillation is common in intranets depending on network traffic, making it a more challenging yet real scenario for the tests. The tracking and classification algorithms were coded in C++ using the OpenCV Library [Bradski and Kaehler, 2008], which contains efficient functions implementing the modules described in Section 3.

For the path prediction algorithm, pedestrian observations from the past five seconds are used, and the vehicle path is predicted ten seconds into the future. The path uncertainty increases linearly with distance from the vehicle from a starting nearest neighbor search radius 5 m at the current position, and increasing by 0.5 m for every meter travelled.

5.2 Results

We have performed several tests with pedestrians both occluded and not occluded from the driver's view, when in manual driving mode. We also tested the system with a single pedestrian as well as multiple pedestrians. Because we have no ground-truth for the pedestrian positions, it is difficult to quantify the system performance in terms of position errors in the tracking or collision

(a) Lights mounted inside the cabin, highlighted by the orange circles in this figure. During manual operation, these lights alert the driver of a potential collision route between the vehicle and a nearby pedestrian.

(b) Flashing light inside the cabin indicating pedestrian on the left side (shown inside the red box). While in autonomous operation, not only do the lights flash but also the vehicle slows down, according to the distance to the pedestrian and the probability of collision.

Figure 6: Illustration of the alert system set-up, with two lights mounted inside the cabin. The side on which the light flashes indicates the most likely pedestrian position.

path prediction. We define a false-positive as a flashing light when no pedestrian is in the vicinity and a false-negative when the system fails to flash the lights when a pedestrian is close to the truck.

Although we experienced examples of false-positives and false-negatives, considering the different views and different trials, these rates were sufficiently low for a practical operation. The main source of errors were strong winds which caused excessive motion in nearby trees, causing false-positives in the video analysis stage. In this type of scenario, a false-positive error is less significant than a false-negative, as the latter can incur a greater loss, such as a collision. Overall, the results illustrate that the system can be very useful as an auxiliary safety measure, successfully alerting drivers of nearby pedestrians with a practical error rate.

Figure 7: Satellite image of the CSIRO site in Brisbane, Australia. The red spots represent the position of the cameras used the experiments.

6 Conclusions

We have proposed and implemented a system for enhancing safety in industrial environments. The system aims to reduce the risk of collisions between pedestrians and vehicles. It uses an onboard navigation system to localise vehicles and offboard cameras to identify and estimate the position of people moving around the worksite. Analysing the information on the vehicles' and pedestrians' positions, the risk of collision can be estimated with a path prediction algorithm. If the risk is above a given threshold, the driver receives an alert, which is a flashing light in our implementation. If the vehicle is driving in autonomous mode, the vehicle slows down according to the risk of collision.

The system was tested in a real industrial scenario and the results indicate that it can be an effective safety measure, successfully alerting drivers of nearby pedestrians. The error rates obtained are considered negligible for practical applications. Still, as with most safety systems, it should always be seen as a support system, and not as a system that the driver should solely rely on by decreasing his/her situational awareness. Future work includes includes storing information about pedestrian and vehicle movements for further analysis. This information can be used to identify common paths taken (temporal movement maps) and record close encounters to improve site safety procedures. In addition, we are preparing to trial the system in a smelter in order to obtain more comprehensive results and to enhance the system for permanent deployment in industrial environments. We plan to further experiment with other variations of the operator interface in order to capture and assimilate driver feedback into the design.

Acknowledgments

This project is supported by the CSIRO Light Metals Flagship. The authors thank Dave Haddon and Paul Flick from the CSIRO Autonomous Systems Lab for their technical support and assistance in running the experiments.

References

[Arya *et al.*, 1998] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45:891–923, 1998.

[Aycard *et al.*, 2006] O. Aycard, A. Spalanzani, M. Yguel, J. Burlet, N. Du Lac, A. De La Fortelle, T. Fraichard, H. Ghorayeb, M. Kais, C. Laugier, et al. PUVAME– New French approach for vulnerable road users safety. In *IEEE Proceedings of Intelligent Vehicles Symposium*, 2006.

[Bradski and Kaehler, 2008] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 1st edition, 2008.

[Caminiti *et al.*, 2010] Lorenzo Caminiti, Jeffrey Clark Lovell, and James Joseph Richardson. Communication based vehicle-pedestrian collision warning system. Patent Application, April 2010. US 2010/0100324 A1.

[Comaniciu and Meer, 2002] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[Enzweiler and Gavrila, 2009] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detecion: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, December 2009.

[Gandhi and Trivedi, 2007] T. Gandhi and M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):413–430, September 2007.

[Gavrila and Munder, 2007] D. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, 2007.

[Hartley and Zisserman, 2004] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 1st edition, 2004.

[Johnsen and Tews, 2009] Swantje Johnsen and Ashley Tews. Real-time object tracking and classification using a static camera. In *Workshop on Pedestrian Detection and Tracking at IEEE International Conference on Robotics and Automation*, 2009.

[Li *et al.*, 2003] Liyuan Li, Weimin Huang, Irene Y.H. Gu, and Qi Tian. Foreground object detection from videos containing complex background. In *ACM International Conference on Multimedia*, 2003.

[Nummiaro *et al.*, 2002] K. Nummiaro, E. Koller-Meier, and L. Van Gool. A color based particle filter. In *1st International Workshop on Generative-Model-Based Vision*, 2002.

[Ran *et al.*, 2007] Yang Ran, Isaac Weiss, Qinfen Zheng, and Larry S. Davis. Pedestrian detection via periodic motion analysis. *International Journal of Computer Vision*, 71(2):143–160, February 2007.

[Senior *et al.*, 2006] A. Senior, A. Hampapur, Y. L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24(11):1233–1243, November 2006.

[Tews *et al.*, 2007] Ash Tews, Cédric Pradalier, and Jonathan Roberts. Autonomous hot metal carrier. In *IEEE International Conference on Robotics and Automation*, April 2007.