Dependable Perception for Robots

Chuck Thorpe, Olivier Clatz, Dave Duggins, Jay Gowdy, Rob MacLachlan, J. Ryan
Miller, Christoph Mertz, Mel Siegel, Chieh-Chih Wang, Teruko Yata
Robotics Institute, Carnegie Mellon University
<firstname.lastname>@ri.cmu.edu

Abstract

The weakest link in many mobile robots is perception. In order to build robots that are reliable and dependable and safe, we need to build robots that can see. Perception is becoming a solved problem for certain constrained environments. But for robots working outdoors, and at high speeds, and in close proximity to people, perception is still incomplete. Our robots need to see objects; to detect motion; and to detect which of those objects are people. In the current state of the art, this requires multiple sensors and multiple means of interpretation. This paper illustrates those principles in the context of the CMU Navlab Group's work on vehicle safety for busses and passenger cars.

1 Introduction

Mobile robots can be unreliable and undependable for a number of reasons: mechanical failures, actuation failures, planning failures, computing crashes, etc. Most of these weaknesses have been addressed for certain environments by engineering approaches: AGVs operate in factories, robots deliver mail in offices and deliver medicines in hospitals, the ConneXXion system shuttles people from a bus stop to their offices, experimental robots run in museums and train stations. These machines are now taken for granted: nobody worries about mechanical failure or runaway actuators. People are comfortable working in close proximity to them, or even riding on board. A recent report on the ConneXXion robot sounds a disappointed tone: riders who could choose either the driverless vehicle or a conventional bus apparently picked whichever one arrived first.[1] While this may be bad news for companies that want to make a splash with attractive vehicles, it's good news for robotics as a field: the public trusts these vehicles and has stopped even noticing.

These successful applications tend to share two characteristics:

1. The environment is "clean". It is often instrumented (with beacons or buried navigation aids) or at least mapped (by leading the vehicle around by hand and collecting observations). Obstacles are relatively easy to detect, since they are easily sensed and segmented from the flat floor or pre-mapped walls.
2. The vehicles move slowly. If an obstacle is detected, the vehicle can simply stop. In some cases, the robot stays put until its preloaded path becomes clear. In other cases, the robot is allowed certain simple obstacle avoidance maneuvers.

But our ambitions, both within our research group at CMU and in the larger robotics community, extend beyond carefully-mapped factory floors and slow-moving robots. The CMU Navlab group, in particular, is working on robot cars, trucks, and busses. Moving from the environments that have successful applications to fully automatic driving on public roads raises a host of issues. Building dependable mechanisms, actuators, planners, and computers will be a challenge; making them affordable to the automotive market will be more of a challenge. Many of these challenges, however, can be met by extensions of the existing dependable systems on current AGVs and other robot vehicles.

The biggest open research challenge is in perception. Operating outdoors, on streets and roads, there are many more kinds of things to see; many more moving objects; and much more variation in lighting, weather, and other perception conditions. Cars, trucks, and busses also operate at much higher speeds than AGVs, so simply detecting an anomaly and coming to a stop is not a sufficient response.

Our approach to perception for dependable outdoor vehicles is:

1. See everything.
2. See everything again, and fuse.
3. Detect motion and predict future motion.
4. Detect people as a special case.

The rest of this paper outlines our work on those four areas.

2 See Everything

Our current main focus is driving in crowded urban environments. In order to build a practical system in the near term, our emphasis is on driver assistance rather than on full automation. Our largest ongoing project is in assisting bus drivers, giving them perception particularly on the sides of the bus and eventually full 360 degree coverage.

We have studied accident reports in several transit jurisdictions to assess the most likely causes of accidents.[2] In some ways, our anecdotal evidence is even more interesting. A casual observation of the sides of busses gives sufficient evidence of the number of side-impact scrapes. Conversations with bus drivers give illustrations of the unusual cases: bicycles going around the curb side of a bus; pedestrians climbing though fences and stepping into the path of a counter-flow bus lane; pedestrians not seeing the side of a stationary bus, walking into it, falling down, then being struck when the bus resumes motion; passengers alighting from a bus then walking immediately in front of the bus, obscured from the driver's viewpoint by the fare box. It is a tribute to the professionalism of bus operators that bus accident rates are low; but the transit industry would like accident rates to be even lower, and is enthusiastic about help from the robotics community.

It is unlikely that any single sensor will be able to detect all objects around the bus. For example, for one simple scenario that we are addressing, we need to detect parked cars, bicycles, pedestrians, oncoming vehicles, and overtaking vehicles. An analysis of required sensing functions, and of sensors suitable for each function, is shown in the table below.

	Video	Stereo Video	Omni Camera	Optical Flow	Motion Detector	Proximity Sensor	Sonar	Short Range Radar	Long Range Radar	Laser line Striper	Laser Scanner	Lane Tracker
Detect parked object in lane ahead												
Recognize lane markings												●
Detect object in front of it	●	●	●	●		●	●	●	●	●	●	
Classify object as in its lane				●					●	●	●	●
Determine if object is stationary	●	●	●	●			●	●	●	●	●	
Detect a bicycle approaching from right												
Detect object to front and right	●	●	●	●		●	●	●	●	●	●	
Determine velocity / trajectory	●	●	●	●				●	●	●	●	
Find bicycle when occluded	●	●	●	●						●	●	
Detect an object (stationary pedestrian)												
Recognize curb boundary										●	●	
Detect object to the right side	●	●	●	●		●	●	●		●	●	
Classify object on the sidewalk				●						●	●	
Detect an object (moving pedestrian)												
Recognize curb boundary										●	●	
Detect object to the right side	●	●	●	●		●	●	●		●	●	
Determine velocity / trajectory	●	●	●	●				●		●	●	
Classify object as on sidewalk				●						●	●	
Detect oncoming vehicles												
Recognize lane markings												●
Detect object on front and left	●	●	●	●		●	●	●	●	●	●	
Classify object as not in its lane				●					●	●	●	●
Determine that object is moving				●			●	●	●	●	●	
Detect vehicle in same lane behind it												
Recognize lane markings												●
Detect object to the rear	●	●	●	●		●	●	●	●	●	●	
Classify object as in its lane				●					●	●	●	●
Determine that object is moving with the flow of traffic				●			●	●	●	●	●	
Total number of functions supported by sensor type	**10**	**10**	**10**	**17**	**0**	**6**	**9**	**11**	**11**	**19**	**19**	**6**

Table 1: Functions vs. sensors for an urban driving scenario

The conclusion from this analysis is that no one sensor is capable of performing all sensing functions; instead, we need a set of sensors, and sensor fusion methods. We are using radars and ladars from commercial vendors. We are building our own light-stripe range sensor and optical flow object detector, as described below.

2.1 Light-Stripe Range Sensor

We are building a laser line-stripe rangefinder suitable for use outdoors. The principle of a line-stripe range sensor is well known: the light of a laser is focused in one direction and fanned in the other and thereby produces a plane of light. A video camera is placed at a distance from the laser and observes where the light intersects objects. An example can be seen on the left in Figure 1. The line can just barely be identified on the garbage can in the foreground and on the legs of the person.

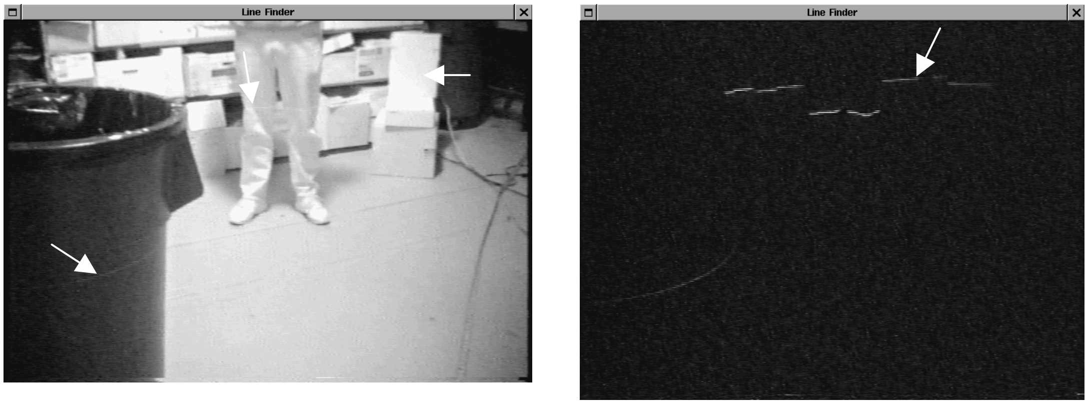

Figure 1 Left, view of the camera without background suppression. The arrows point to places where the laser line can be seen. The box in the back is positioned at 4m from the camera. Right, view of the camera with background suppression. The arrow points to where the laser line hits the box.

To make the line stand out more clearly, the background can be suppressed. This is done on the right in Figure 1. The laser points are extracted, then the distance to each point is calculated by triangulation based on the geometry of the laser and camera. The resulting map is shown in Figure 2. The advantage of this sensor is that it can produce a single stripe of range information, at frame rates, with no moving parts.

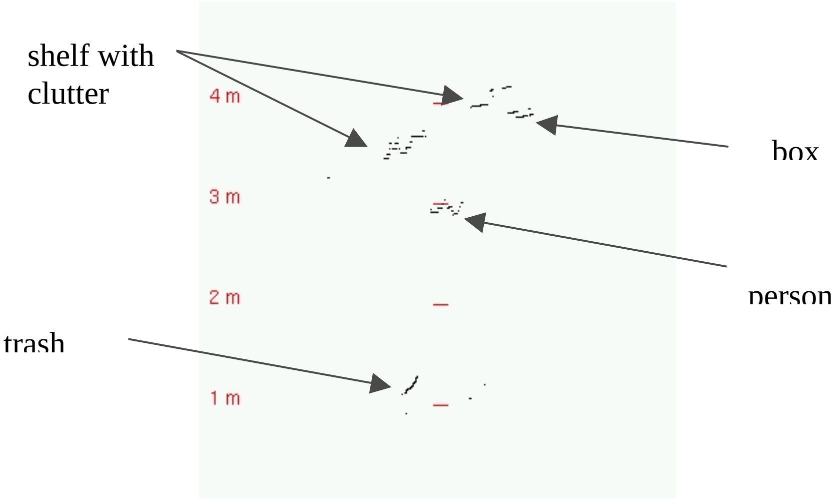

Figure 2 The extracted line in x-y world coordinates

The difficulty of building this sensor is making it work in bright sunlight, where the laser light stripe would normally be hard to detect against the bright background. The laser power is limited by eye safety regulations. We use several approaches in parallel to make the laser more easily detectable:

- Filtering: a bandpass filter that closely matches the laser is put over the camera
- Shuttering: the laser is fired in a pulse of a few microseconds, and the camera's electronic shutter is opened for the same interval
- Image subtraction: For a stationary scene, it is easy to collect two images, one with the laser on and one with it off, and subtract images. For a moving scene, implementing this process requires image-splitting optics or image registration.

The results to date are that the light striper is suitable for limited ranges under most viewing conditions. We plan to use the light stripe ranger to detect nearby objects, and specifically to detect and track the curb.

2.2 Omnicamera Optical Flow

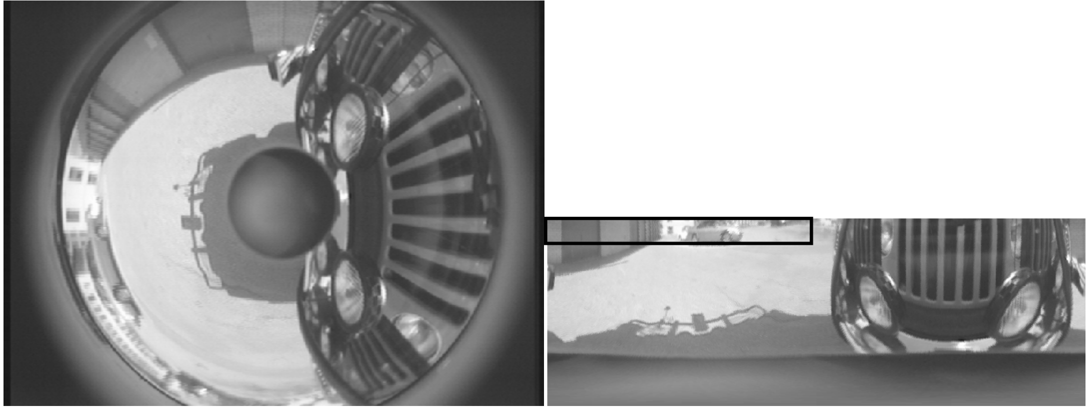

Figure 3: Left: Image from the omni-camera. The vehicle can be seen on the right side of the image. Right: the unwarped image, converted into cylindrical coordinates. The box at the upper left shows the portion of interest, corresponding to a forward-looking image.

A second sensing system we are developing for our vehicles is optical flow based on omnicamera sensing. Since most of the vehicle motion is confined to a plane, the image motion of interest will primarily be horizontal (in the unwarped images; circumferential in the original coordinates). We unwarp only the portion of the image corresponding to looking forward and approximately horizontal. We run a vertical Sobel operator to find vertical edges, then track them from image to image. Figure 4 shows an original image, and a sequence of five Sobel images from successive frames. The motion from image to image is relatively small and consistent, enabling tracking edges across the sequence. Figure 5 shows the vehicle motion and object positions calculated for one sequence.

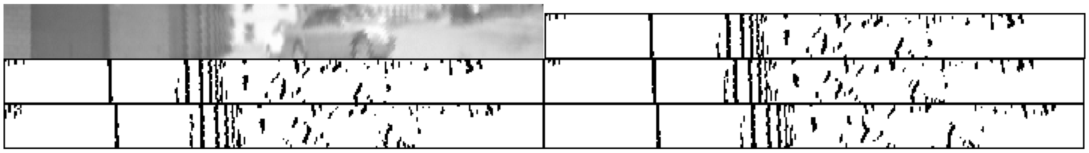

Figure 4: Original image, and sequence of five edge images.

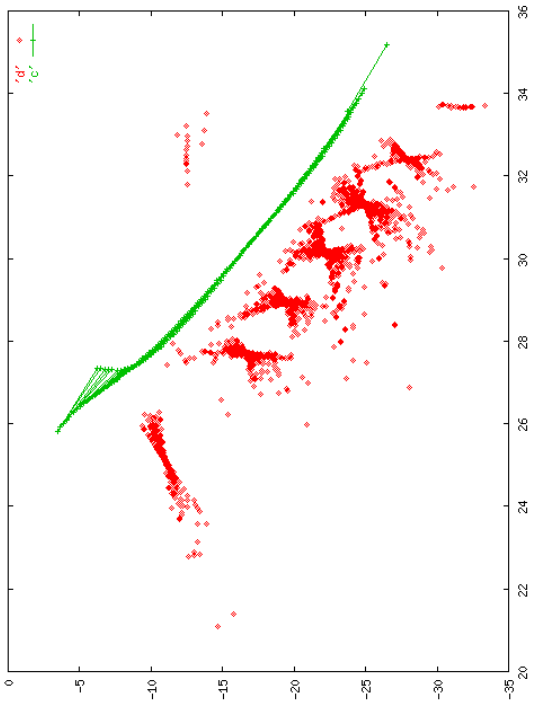

Figure 5: Map built from optical flow. The line segments show the calculated vehicle position and heading at each frame. The points show calculated object positions. Each object position is calculated as the closest intersection of five lines of sight from the optical tracking. The next step is to filter calculated positions over more than five observations, to reduce the scatter around the mean locations.

3 See Everything Again

While the individual sensors are good, none of them can see all the features that we need to detect; and no one sensor is reliable in all circumstances. Thus, a dominant theme of our work is sensor fusion. It is our thesis that:

1) no single sensor will provide complete and affordable coverage, therefore sensor fusion will be required;
2) no single sensor fusion methodology will be suitable for all combinations of sensors.

There are many techniques that can be used for sensor fusion: evidence grids, Kalman filters, sensor co-location, etc. Our philosophy is not to use a single kind of fusion for all applications, but rather to use the most appropriate fusion method for each set of sensors. Accordingly, in our existing design, we use different kinds of fusion at different points in the system. Figure 6 illustrates the overall system design and the opportunities for fusion.

3.1 Multi-Sensor Fusion

Simple radars by themselves measure range but not bearing. Optical flow processing by itself produces bearing information but no true range data. Combining the two produces much richer data than either one could independently.

The most compelling example is in the case of moving objects. A simple radar will produce a range measurement, but no information of object size or bearing. Optical flow processing essentially tracks a feature as the vehicle moves, and therefore generates a set of rays pointing towards the tracked object, one ray for each vehicle position. In a static world, if the vehicle motion is well known, the rays will all intersect at a single point, which is the object's location. But if the object is moving at the same time as the vehicle (e.g. another moving car, a bicyclist, etc) then the intersection of the rays will not be the correct location of the object. In effect, optical flow gives a linear relationship between object velocity, distance, and size, rather than any absolute measurement. A single range measurement, for example from a simple radar, will fix the distance to the object, and thus allow unambiguous determination of velocity and size.

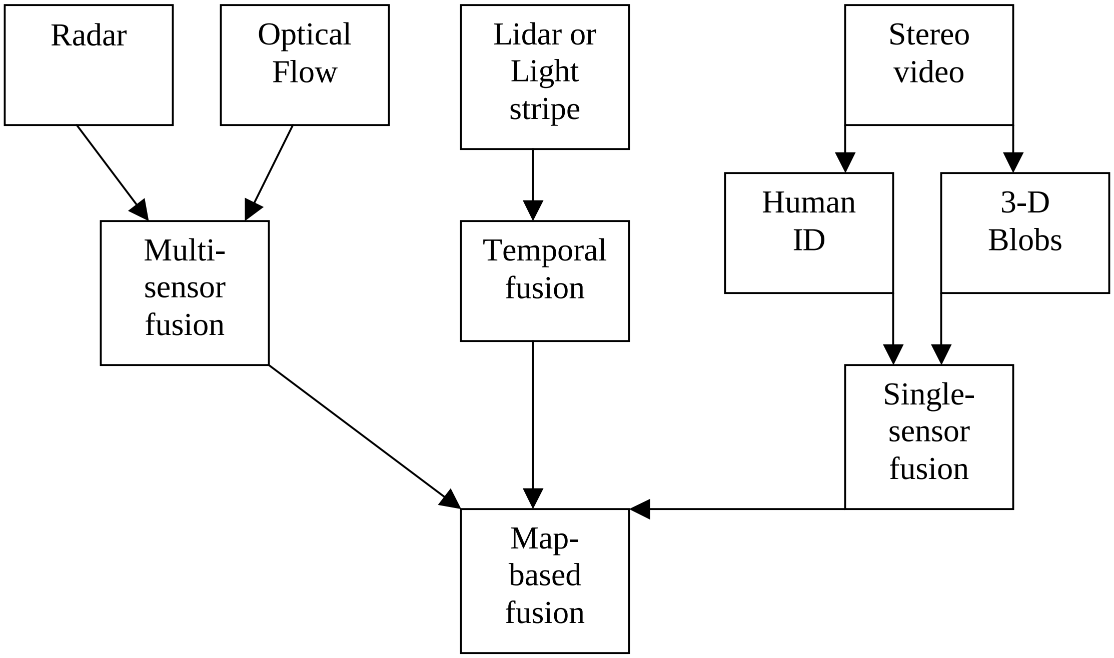

Figure 6: Fusion methods at various places in the data flow.

3.2 Temporal Fusion

Lidar and light-stripe ranging give range from a single viewpoint, but are subject to occlusions and give no direct measurement of velocity information. The fusion processing used for them is time-based, watching the same sensor over multiple scans as the vehicle moves and finding matching data form scan to scan; details are given in section 4.

3.3 Single-Sensor Fusion

The easiest case of fusion is doing different kinds of processing on the same data, then combining the results. The stereo processing, described below in section 5, processes stereo images to calculate range data and processes grayscale images to detect people. Since the underlying data come form the same sensor, fusion is straightforward.

3.4 Map-Based Fusion

The output of each subsystem is a symbolic abstraction of the world from the point of view of that subsystem. These symbolic abstractions of object positions, speeds, shapes, and classifications can be painted into a dynamic environment map, which then forms a common, although abstracted, representation of the world. Inevitably there will be significant information lost in the abstraction process, but the abstracted information is vastly more tractable, and can be dealt with in a more general manner. We have structured our world map management system as a "hypothesis pool," in which abstract information about objects from the measurement fusion subsystems form, support, and weaken hypotheses about the world. We have structured the map management system to be able to experiment with a wide variety of hypotheses. For example, if the measurement subsystems produce estimates of object positions and velocities, it is straightforward to assign sensed objects to object hypotheses using simple nearest neighbor data association and maintain hypotheses of object positions and velocities with a linear Kalman filter. Beyond such simple, generic map-based fusion, we can build more complicated hypotheses which actually reason about the measurement subsystem capabilities in order to form a more accurate, although still generic, model of the environment. The fused map information becomes the system's model of the world which can be displayed to the user or propagated to situational awareness modules.

4 Detect Motion And Predict Future Motion

Detecting objects is necessary but not sufficient; the question we really want to answer is not only "where are nearby objects now?" but "where will objects move to in the future?". We are using ladar processing to find moving objects, and to perform some limited object classification. Range data processing has been investigated for mining[3]; off-road driving and mapping [4]; indoor navigation [5] [6] [7]; and other applications. But only a small amount of this work has investigated using lidar data to find and track moving vehicles[8]: we are in essence tackling two problems at once: SLAM (Simultaneous Localization And Mapping)[9] and DTCMO (Detection, Tracking and Classification of Moving Objects).

We mounted a Sick ladar on the side of our Navlab 8 vehicle, drove through the CMU campus and around nearby streets, and collected range data. For each frame of range data, we first segment the range map into connected objects. For each object, we check for a match against the ongoing "moving objects" list, and separate the points associated with moving objects. In the second step, we use an Iterative Closest Points (ICP) algorithm to register non-moving points with the local map, built from the previous scans, in order to compute relative vehicle position. The idea is to use the ICP rule to establish correspondences between points in the current scan and the local map and then solve the point-to-point least-squares problem to compute the relative pose of the current scan and the local map. We use a Kalman filter to predict the pose of the vehicle.

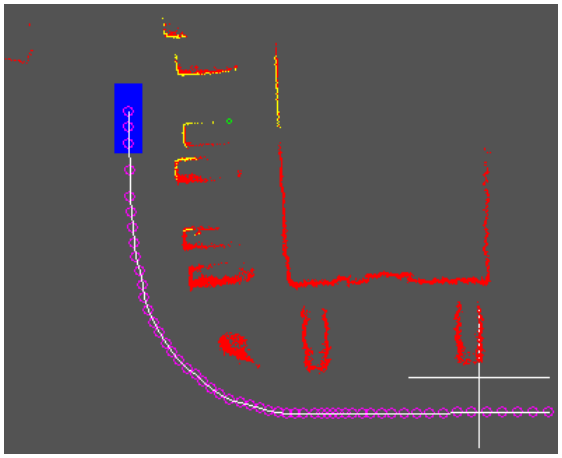

Figure 7: Localization and Mapping

After the registration between the two frames is found, the moving object detection algorithm uses the calculated vehicle pose to separate any new moving objects from stationary objects. From previous scans we know which areas should not be occupied. If we find an object in these areas, that means this object is moving. Then the whole process iterates. The results of relative localization and mapping are shown in Figure 7.

The results of the moving object detection are shown in Figure 8 and Figure 9. Our algorithms found both moving pedestrians and cars successfully.

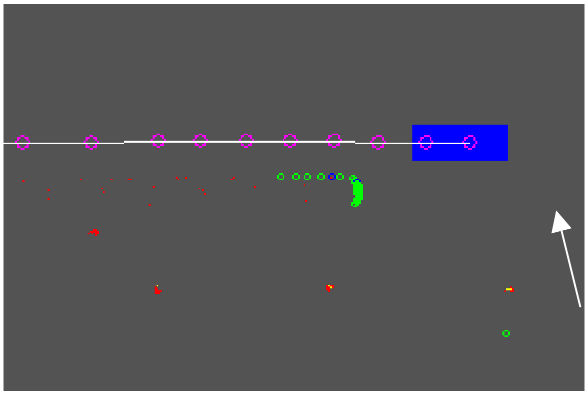

Figure 8: Vehicle Detection: the points shown by the arrow have the shape and motion of a moving car

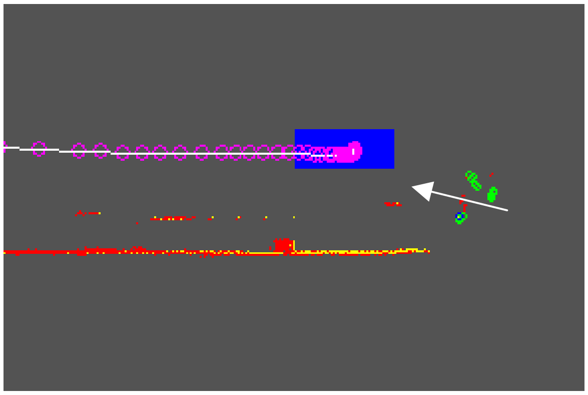

Figure 9: Pedestrian Detection: the group of points designated by the arrow have the shape and motion characteristics of a walking pedestrian

5 Detect People As A Special Case

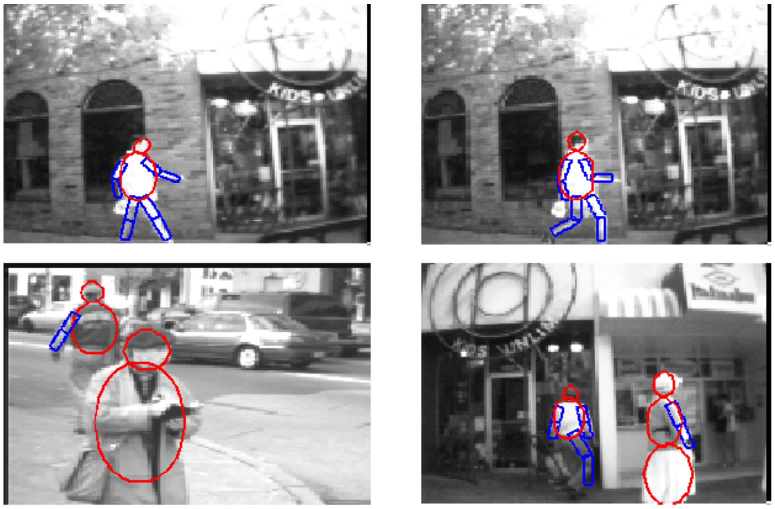

Figure 10: Human detection

In addition to finding objects and predicting motion, we would like to pay special attention to people. We begin with stereo vision, using various commercially-available stereo vision systems to give us 3-D information. We segment the scene into 3-D blobs, which are directly useful for obstacle detection. Then, for each 3-D blob, we look at the size and rough shape: if the size and shape could be a human, we use the outline of the 3-D blob as an initial segmentation in the original gray-scale image. The region segmented from the gray-scale image is then examined to see if its appearance looks like a human.

Early versions of this process used a neural net, trained on examples of humans. The current version uses a parts-based segmentation, driven by anthropometric models. The examples shown in Figure 10 show the body parts that are found in each detected person.

6 Conclusion

Perception is still an important challenge for outdoor intelligent vehicles, operating in the vicinity of other vehicles and people. Solving the perception challenge will take a number of innovations: new sensors, new sensor processing, and new fusion systems. We are working on each of those areas.

We are motivated in our work both by scientific interest and by the practical challenges of building safe, reliable, and dependable vehicles.

7 Acknowledgements

This work is partially supported by a grant from SAIC, by a subcontract from the DARPA MARS program through SAIC, by Robert Bosch Corporation, by funding from the Federal Transit Agency and PennDOT, and by a JSPS fellowship. The views and conclusions in the document are those of the authors, and not necessarily those of the sponsors.

8 References

[1] G. Bootsma and R. Koolen, What Moves People?, in Traffic Technology International, April/May 2001, UK & International Press

[2] Buses C. Mertz, S. McNeil, and C. Thorpe. Side Collision Warning Systems for Transit Buses in Proceedings of IV 2000, IEEE Intelligent Vehicle Symposium, October, 2000

[3] G. Dissanayake, M. Hebert, A. Stentz, and H. Durrant-Whyte, Map Building and Terrain-Aided Localisation in an Underground Mine, Proceedings of the Field and Service Robotics Conference, 1997.

[4] D.F. Huber, O. Carmichael, and M. Hebert , 3D map reconstruction from range data, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA ’00), Vol. 1, April, 2000, pp. 891 - 897

[5] J.-S. Gutmann and C. Schlegel, AMOS: Comparison of scan matching approaches for self-localization in indoor environments, Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots, IEEE Computer Society Press, 1996

[6] Lu, F. and Milios, E., Globally consistent range scan alignment for environment mapping, Autonomous Robots, 4, 333--349, 1997.

[7] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, Robust Monte Carlo Localization for Mobile Robots Artificial Intelligence Journal, 2001.

[8] L. Zhao and C. Thorpe, Qualitative and Quantitative Car Tracking from a Range Image Sequence, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, June, 1998, pp. 496-501.

[9] J. Gonzalez and R. Gutierrez, Direct motion estimation from a range scan sequence, Journal of Robotics Systems 16(2), pp 73-80, 1999.