

## Research Article

# Human-Machine Interface for a Smart Wheelchair

**Amiel Hartman and Vidya K. Nandikolla** 

*College of Engineering and Computer Science, California State University at Northridge, CA, USA*

Correspondence should be addressed to Vidya K. Nandikolla; [vidya.nandikolla@csun.edu](mailto:vidya.nandikolla@csun.edu)

Received 10 July 2018; Revised 18 November 2018; Accepted 2 December 2018; Published 2 January 2019

Guest Editor: Ling-Ling Li

Copyright © 2019 Amiel Hartman and Vidya K. Nandikolla. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper describes the integration of hardware and software with sensor technology and computer processing to develop the next generation intelligent wheelchair. The focus is a computer cluster design to test high performance computing for smart wheelchair operation and human interaction. The LabVIEW cluster is developed for real-time autonomous path planning and sensor data processing. Four small form factor computers are connected over a Gigabit Ethernet local area network to form the computer cluster. Autonomous programs are distributed across the cluster for increased task parallelism to improve processing time performance. The distributed programs operating frequency for path planning and motion control is 50Hz and 12.3Hz for 0.3 megapixel robot vision system. To monitor the operation and control of the distributed LabVIEW code, network automation is integrated into the cluster software along with a performance monitor. A link between the computer motion control program and the wheelchair joystick control of the drive train is developed for the computer control interface. A perception sensor array and control circuitry is integrated with the computer system to detect and respond to the wheelchair environment. Multiple cameras are used for image processing and scanning laser rangefinder sensors for obstacle avoidance in the cluster program. A centralized power system is integrated to power the smart wheelchair along with the cluster and sensor feedback system. The on board computer system is evaluated for cluster processing performance for the smart wheelchair, incorporating camera machine vision and LiDAR perception for terrain obstacle detection, operating in urban scenarios.

## 1. Introduction

Assistive technology is essential for elderly and disabled communities to help in daily living activities, socialization, and traveling. It is also known that the robotic application in the medical mobility can provide a better life for the people with both lower and upper extremity impairments. While assistive robotic technology is progressing rapidly to improve the mobility of people, several challenges remain to make this technology truly usable by the humans. One important aspect that requires research development is defining the control protocols between the human and the robot technology. There are different types of wheelchairs including basic, lightweight, folding, multi-function, powered, fully/partially autonomous and so on. And there are many types of control design to manipulate the functionality of the wheelchair, from basic drive to fully controlled wheelchair using brain-controlled interface. However, the power wheelchair users frequently report accidents, therefore our focus is to advocate

the use of robotic technology, in particular sensor-based detection and navigation using smart wheelchairs [1–4].

Smart wheelchair is generally equipped with sensors, cameras and computer-based system as main processing unit to be able to perform specific task. Autonomous smart wheelchairs are controlled by human user interface where the human makes decisions at the highest level of operation and the smart control technology makes the rest of the motion automatic. The advances in autonomous smart wheelchair are embedded with computers and focuses heavily on the computer cluster architecture. The intelligence is added to a wheelchair platform around user control despite their disabilities, which makes the study of the human-machine interface (HMI) between the user and the wheelchair an important assistive robotic field of study. Standard electric powered wheelchair has little computer control with some level of motor control using a joystick. Therefore, researchers are focusing on computer-controlled wheelchair integrating sensors and intelligence to decrease the need of human

intervention. But attempts to build smart wheelchairs mostly lack robustness in motion control, perception and control techniques. This research work describes the development of a smart wheelchair integrated with a HMI and its interaction between sensory feedback and the computer control system [5–8].

The main focus of the paper demonstrates the design and performance of the interface between sensory feedback and the computer-controlled system. The real time data processing is addressed here for a smart wheelchair that functions as a low speed autonomous vehicle. The focus is on the implementation of mobile high-performance computing (HPC) cluster comprised of a multi-computer system connected over a local area network (LAN). A scalable network communication pipeline is developed that is expandable to accommodate additional computers as needed for light or intensive data processing loads. A user interface is developed that runs on a single client computer terminal in the network. It is used to activate and monitor the real time performance status of the software on other server computers in the network. Hardware and software parallelism is implemented to improve data processing. The following sections in this paper focus on: related work, smart wheelchair system, computer cluster network (system architecture), visual interface, computer cluster performance followed with results and conclusion.

## 2. Related Work

Ground vehicles developed for intelligent navigation come in many form factors. When developing a smart wheelchair it is significant to consider the operating conditions for the platform, in regards to the environment and also the user interaction with the system controls. A smart wheelchair platform can be constructed from different approaches including modifying an EPW, converting a manual wheelchair, or adding a seat to a mobile robot. Building a smart wheelchair with electronics added to an EPW results in the smart features being connected to the incorporated EPW controller to access the embedded drive train [10–12]. Connecting smart feature electronics to an EPW motion controller has the added benefit of being compatible with commercially available controllers, both joysticks, and alternative controllers for persons with multiple disabilities.

The type and placement of perception sensors on an EPW for smart wheelchair navigation can have a significant impact on the performance of the system as well as how the platform interacts with the wheelchair user. Some smart wheelchair designs favor sensor placement where the vehicle perception can be optimized but it impacts the EPW height and user access to the platform [10–13]. Newer developments in perception sensor technology can help to negate sensor placement limitations and reduce the impact of sensor placement on the platform accessibility [14, 15]. Furthermore, processing of high-resolution data sets from perception sensors for real time navigation can require high performance computing methods.

Many self-driving wheelchair platforms use a laptop computer for onboard processing [12, 13, 16]. However,

using only a single laptop computer can impose significant constraints on the real time processing capabilities of a smart wheelchair. An alternative method is to use cloud computing to offload real time data processing from the smart wheelchair hardware [12]. In the past, high cost perception sensors have hindered smart wheelchair applications. The advent of low cost LiDAR in recent years, and the abundance of inexpensive machine vision cameras, allows incorporating autonomous navigation capabilities universally into EPW platforms. Currently, high bandwidth, low cost perception sensors can collect large amounts of data in real time, which requires significant computing power for the real time autonomous navigation for the smart wheelchair. While laptops and small form factor computers can be used for real time data processing, but as sensor hardware improves, the large perception sensor data sets will require alternative and more specialized embedded data processing hardware. Our approach is to use four small form factor computers for high performance real time processing on the smart wheelchair. Perceptions sensor data processing is distributed in the computer cluster configuration to improve real time performance. The design of the smart wheelchair uses EPW with computer motion control integrated into EPW joystick controller for motion control of the embedded drive train. Movable sensor mounts and armrests provides desirable sensor placement while allowing access from the front or side without increasing the wheelchair height [8, 9].

## 3. Smart Wheelchair System

The smart wheelchair system is comprised of a Jazzy 600ES electric-powered wheelchair (EPW) with added hardware and software for autonomous navigation and user interface. Mechanical modifications to the EPW incorporate a retrofitted footplate with rotating sensor tower at the front of the platform for mounting oscillating LRF sensors for range sensing and color cameras for terrain detection (Figure 1) [8, 9]. Optical rotary encoders are coupled to the drive train as part of the motion control interface for autonomous navigation. A National Instruments (NI) reconfigurable input/output (RIO) board is connected to the encoders for motion control feedback, the EPW joystick for control commands, and a computer for the real time control technology. A secondary mobile power system for the smart wheelchair is operated in conjunction with the EPW embedded power system to boost platform run time. The secondary power system is built using automotive and marine commercial off-the-shelf (COTS) electronic components incorporating a lithium-ion polymer (Li-Po) battery pack and power distribution panel with component level circuit protection (Figure 1) [8, 9].

The autonomous navigation capabilities of the smart wheelchair including perception, localization, path planning, and motion control, operate in the LabVIEW software environment. The path planning and motion control programs are optimized for the differential drive wheelchair platform. Small form factor (SFF) computers are connected over LAN with an Ethernet switch to explore the benefits of HPC cluster for real time sensor data processing and autonomous



FIGURE 1: Retrofitted power wheelchair showing all the added features [8, 9].

navigation. Network automation is used to enhance control of the LabVIEW programs to simplify the user interface. A touch screen monitor user interface connects to the client computer for software system access. A real time performance monitor is incorporated into the user interface to provide sensor data visual feedback and track the status of the LabVIEW programs on the computer cluster during smart wheelchair operation [8].

#### 4. Computer Cluster Network (System Architecture)

The primary objective of the HPC cluster is to improve real time data processing capabilities of the smart wheelchair's perception and intelligent navigation. The computer cluster system architecture can be seen in Figure 2 that shows the perception and cognition connectivity. The programs are expanded to function on a cluster as distributed computing network. The homogeneous cluster is constructed of four SFF desktop computers with low power components commonly found in laptop computers [9–11, 13, 17]. This allows achieving a small physical computer volume significant for the smart wheelchair mobile platform. Each computer contains a multi-core CPU with eight processing threads [18]. The LAN connecting the cluster computers is configured as an Internet Protocol version 4 (IPv4) class C private network. The scalable cluster network can incorporate additional computers, limited by available mounting space on the smart wheelchair platform.

The smart wheelchair system architecture (Figure 2) is separated into the two primary sections: perception and cognition. The computer cluster layout is designed so that one

computer is designated for cognition and three computers are designated for the perception task. The perception sensor data processing is further separated into 3D point cloud depth data from the LRF sensors and RGB camera vision image processing. Cognition incorporates the decision process for path planning and interfacing with the EPW electronics for execution of motion control. The process of acquiring depth data starts with the LRF sensors that are used to collect a 3D point cloud of depth data. The data is filtered for noise and reduced to a horizontal plane 2D representation for obstacles in front of the smart wheelchair. A binary representation of the data as nearest obstacle edges is used to reduce the amount of data being transferred between the networked computers. This also reduces the amount of data being processed for path planning. The nearest obstacle edge simplified format is a means of combining LRF and camera vision data into useful map of terrain obstacles for cognition.

Two RGB cameras are used in front of the smart wheelchair to collect color image data of the environment. The cameras are connected to a computer through a USB hub for power and data transfer. The RGB data is processed through a series of filters before being converted into a binary representation of extracted terrain obstacles. The RGB data is also used to represent the terrain obstacle data for the user visual interface.

The perception data is transferred from each of the computers designated for the perception task over the Ethernet LAN to the cognition computer. The perception data is combined on the cognition computer and used for path planning determination or user visual interface. A distance histogram of obstacle leading edges is used for high speed real time path planning. The cognition computer is connected to

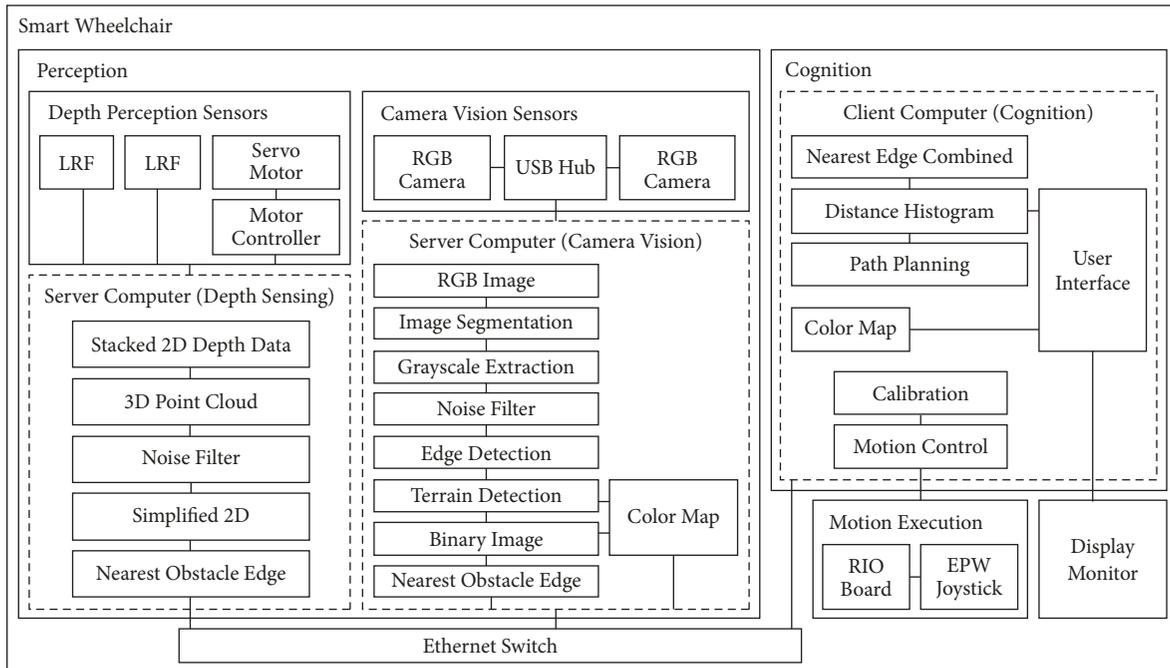


FIGURE 2: Computer cluster system architecture [9].

the EPW joystick using a National Instrument's RIO board for motion control of the drive train.

The smart wheelchair software is segmented into separate programs to operate concurrently on multiple computing nodes to reduce the time required for data processing. Data is transferred between the memory spaces of the computers and the RIO board using network published shared variables in LabVIEW. Beneath the client-server model for the computer cluster is the abstraction layer of the LabVIEW shared variable engine. This is hosted on the client computer as a virtual server and used to control the data transfer between computing nodes writing updates to and reading data from the network [19, 20]. The shared variables are configured for message passing without a buffer to transfer the most current data sets for real time autonomous navigation.

**4.1. Network Automation.** The goal of network automation is to simplify control of the autonomous navigation software distributed on the computer cluster network for real time operation of the wheelchair. The network automation subroutines operate in the background on the client computer transparent to the user interface. The subroutines are programmed with three primary functions: activating, monitoring and deactivating in coordination with the cognition program on the client computer. The deactivation of the server computer programs are used to make sure other programs are not running in the background when the data is not being processed by the cognition for navigation. The subroutines monitor the real time status of the server programs as part of the client computer to inform the wheelchair user if the system is operating normally. The details of the network automation architecture are described further.

The computer cluster network as seen in Figure 3 is separated into one client computer and three server computers based on the separation of the perception and cognition tasks. The client computer incorporates the user interface referred as the cognition computer. The client computer communicates with the three server computers used for perception over the LAN. The method of communication between computers on the network is using network variables that function as part of the LabVIEW shared variable engine. The cognition client computer is the host for the shared variable engine and the perception server computers are the subscribers. The cognition program is used to activate background network automation programs on the client computer. The background network automation programs contain the functionality to control and monitor the distributed perception programs on the server computers. They provide the automated functionality of launching the distributed algorithms across the network from the client computer. The network automation programs are also used to monitor the state of the perception and cognition programs to allow for more automated program interaction rather than limited to data transfer.

LabVIEW VI Server functionality is integrated in the client computer cognition program to dynamically call the subroutines to run independently so that the subroutines can respond to the activation and deactivation of the cognition program. Dynamic calling is used within the subroutines to target the server computer programs across the LAN in combination with static IP addressing. The cognition program for each server computer in the cluster activates a separate subroutine. Error handling is used within the network automation subroutines to prevent LabVIEW software failure. The cognition program and network automation

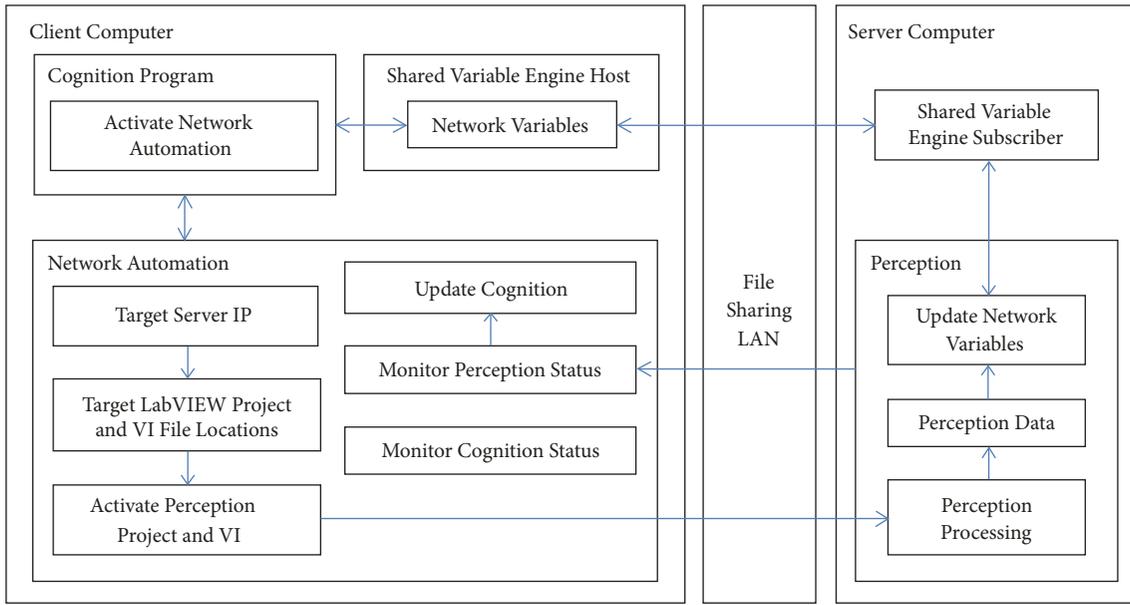


FIGURE 3: Network automation architecture [9].

subroutines work together to reset the status of the server computer programs if the autonomous navigation is reinitialized on the client computer. The reinitialization process is effective at resetting the cluster software from the client computer after correcting for accidental camera sensor hardware disconnect, without the need to access the server computers.

## 5. Vision Enhancement

The purpose of the vision system for the smart wheelchair is twofold: detection of terrain obstacles and integration of detected data into the user interface for visual feedback. The terrain obstacle detection features are expanded to improve processing with computer cluster hardware for higher resolution RGB images and multiple LRFs. Multiple sensors data is collected instead of a single sensor to improve data resolution and scope for feature detection [8, 13, 17, 21–23].

The color machine vision cameras used on the smart wheelchair are capable of delivering a 1920 horizontal by 1200 vertical (1920x1200) maximum pixel resolution RGB image. The maximum tested resolution for the real time image-processing program is 960x600 for each camera. Higher resolutions produce slow results for real time autonomy of less than 2Hz operating frequency. The minimum tested pixel resolution from each camera is 120x75, maintaining the 16:10 aspect ratio of the camera digital sensor. The number of image segments tested on each computer ranges from one to four. The two side-by-side cameras are connected to each vision server computer to increase the horizontal field of view. Therefore, the tested image resolution for a single vision computer varies from 240x75 to 1920x600 pixel resolution and these dual camera images are split up to four segments. To minimize the network data traffic, the segmented images are combined into a single image on the vision computers for processing before transferring to the client computer over

the LAN. The tested resolutions and image segmentation are based on the combined dual camera image processed on each vision computer.

Multiple cameras fixed at different orientations on the smart wheelchair is challenging as each camera is exposed to different light conditions in outdoors during transit. Each RGB image is processed using a separate image-processing pipeline to adapt to different lighting conditions. Image segmentation is used to improve computer CPU parallel processing capabilities of the vision programs for faster real time image processing. Segmented images and data from multiple RGB sensors are concatenated to produce a combined terrain map.

The RGB data processing pipeline incorporates data reduction for improved processing time. RGB data resolutions are selected for real time transit based on the computer cluster processing capabilities. The cluster processing results for one vision computer is discussed. Testing results indicate the performance is similar on both vision computers and the cluster processing time or CPU usage is unaffected by the changes in the terrain image.

**5.1. Terrain Perception.** The RGB data is converted to multiple gray scale images for different feature extractions. The gray scale data is filtered for noise and enhanced to improve feature contrast. Edge detection is used to enhance feature extraction. Isolated terrain obstacles are converted to binary representations. The terrain data is converted to a nearest obstacle edge histogram (Figure 7) and combined with LRF data for obstacle avoidance and path planning. Multiple, color coated, binary images are combined for the terrain map visual interface (Figures 4 and 5).

The consolidated terrain data is represented in binary image to separate passable and impassable terrain or color coded to represent concrete, dirt, undesired terrain, shadows

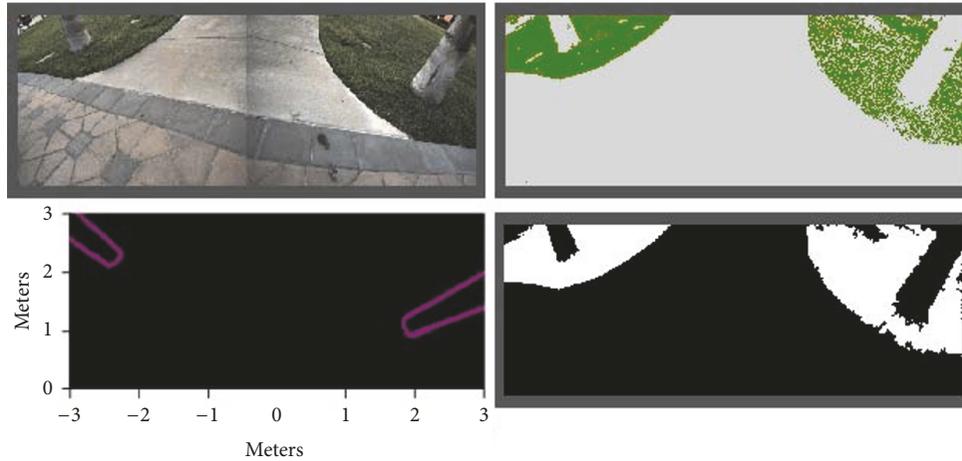


FIGURE 4: Image captured by dual camera, from top left clockwise: source image, color coded terrain, terrain obstacle binary image, LRF distance histogram [9].

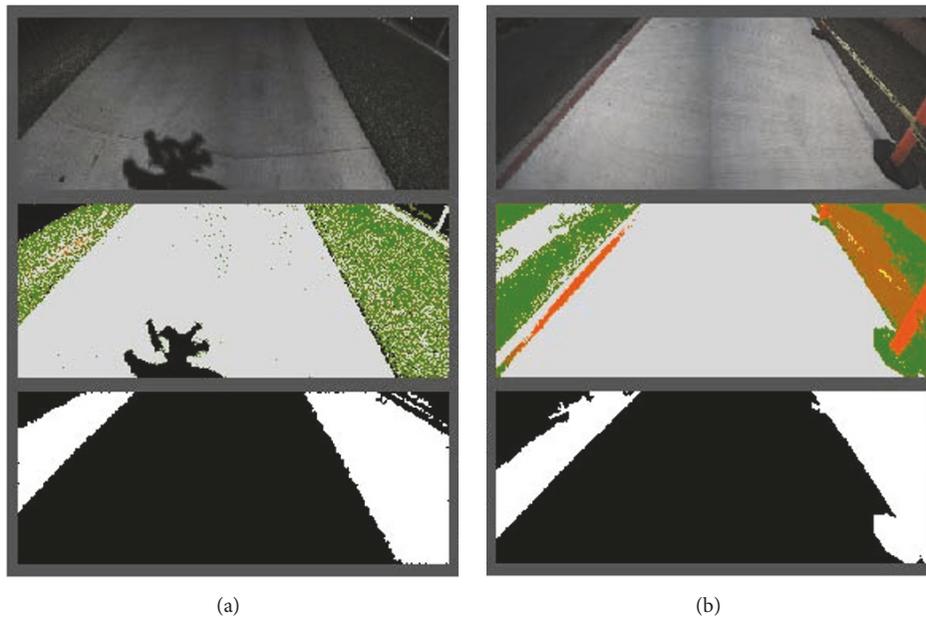


FIGURE 5: Dual camera images (a) shadows (b) construction marker. From top to bottom: source image, color-coded terrain obstacle visual interface, and terrain obstacle binary image [9].

and brightly colored construction markers in the pathway of the smart wheelchair (Figures 4 and 5). Figure 4 shows an outdoor daylight scenario captured by a dual RGB camera configuration. The image array data is used to combine the two images into a wider field of view representation as a single image. The LRF data shows the two tree trunks represented in distance histogram format, which is insufficient information for desirable scene for navigation. Therefore, the vision data is utilized to detect the edges of the grass as undesirable terrain, while filtering the three shades of concrete as a desirable open path. The data from the vision binary image is converted to a distance histogram shown in Figure 7 and combined with LRF histogram to create the nearest obstacle edge histogram

for path planning. Since the trees are located within the grass terrain areas, the vision distance histogram indicates the nearest obstacle edge data for the scene.

Figure 5 shows two outdoor scenarios where the scene on the left side of the figure is a source image with shadow cast onto the terrain in front of the smart wheelchair. The color and gray scale histogram analysis used to filter the image detects the shadow contrast from the concrete path. The detected shadow is removed from the binary image representation and obstacle distance histogram. The right side in Figure 5 shows a sidewalk scene with road on one side and dirt with construction markers on the other. The red curb is detected along with the orange and yellow construction

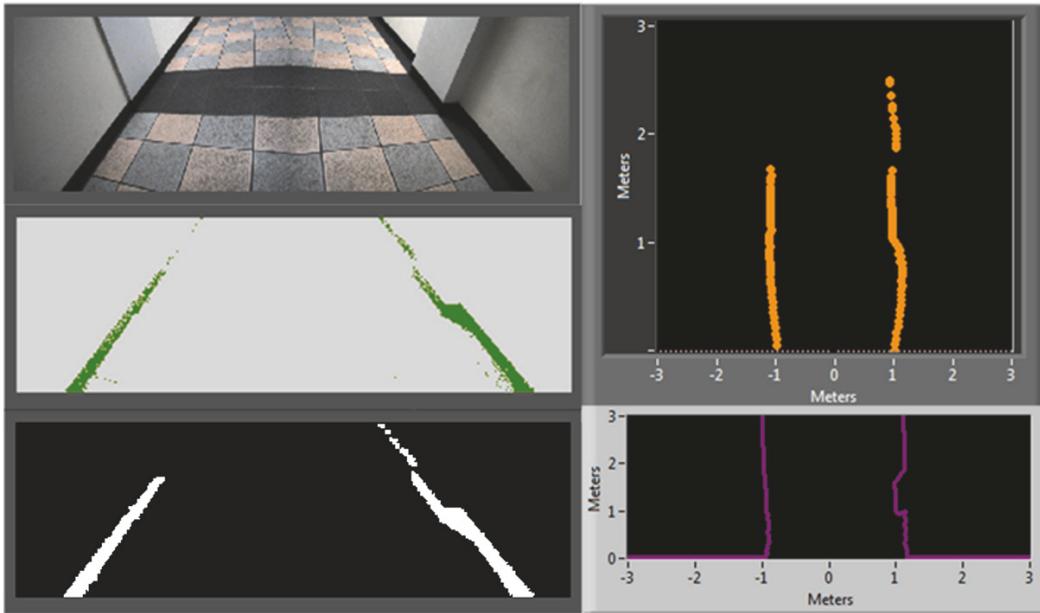


FIGURE 6: Camera images for indoor corridor and front panel display [9].

marker. In this scene, for real time operation the red curb and yellow construction tape are removed from the filtered binary image due to low resolution data. The dirt and base of the orange construction post is detected as side of the walkway along with the road and represented in the terrain binary image as seen in Figure 5.

**5.2. Data Reduction Process.** The smart wheelchair perception system uses 3D point cloud depth data and RGB images. The LRF hardware sensor is used for 3D depth data collection and data processing for real time operation. The raw 3D depth data processing has little effect on the real time performance of the system compared to RGB image processing (Figure 7). Consequently, data reduction is primarily applied to the data collected from RGB sensors. The RGB image data is converted to multiple gray scale images for filtering and feature extraction. Converting the data into binary image data further reduces the useful gray scale image. Additional filtering is applied to the binary images that contain smaller data set. Binary image data is further reduced to a distance histogram.

**5.3. RGB and Depth Data Fusion.** The RGB image data is combined with the LRF data for obstacle avoidance and path planning. The 3D point cloud depth data collected using the LRF sensors is filtered and converted to a stack of 2D depth data horizontal planes. The depth data planes are then compared and reduced to a single 2D nearest obstacle edge histogram for efficient path planning. The RGB image data is converted to high contrast gray scale and reduced to binary images in the data reduction process. Multiple binary images are combined to compile the terrain obstacle detection data. The stacked binary images are reduced to a nearest obstacle edge distance histogram to combine with the LRF

distance histogram data. The camera distance histograms are calibrated to compensate for camera lens barrel and mounting angle perspective distortions. Distortion correction is incorporated into the distance histogram process to keep processing time low. The result is less powerful but faster calibration than image pixel mapping. The combined camera and LRF histogram extracts the closest edges of detected terrain obstacles.

**5.4. Indoor and Outdoor Operation.** Indoor and outdoor light conditions differ significantly. Outdoor daylight scenarios can often involve direct sunlight exposure, while indoor lighting conditions are typically low light in comparison. The LRF sensors are unaffected by light exposure from outdoor direct sunlight or indoor low light conditions. Bright and low light conditions have a more significant effect on RGB camera vision. Different calibration techniques for RGB data are required for indoor, low light and outdoor sunlight scenarios. Calibration of image contrast on RGB and gray scale images is used to compensate for changing light conditions.

Detecting the continuous smooth walls of the inside corridor shown in Figure 6 can be accomplished with a 2D LRF. The vision distance histogram shows the detection of the corridor walls compared to the multicolored tiles on the ground plane. Comparing the vision histogram to the LRF distance histogram shows the similarity in detection of the walls across multiple sensors. More complex object detection indoors is addressed in a similar fashion to outdoor scenarios with the combination of filtered RGB and 3D depth data.

## 6. Cluster Processing Results

The performance of the cluster processing is evaluated to determine the benefits of increased parallel computing for

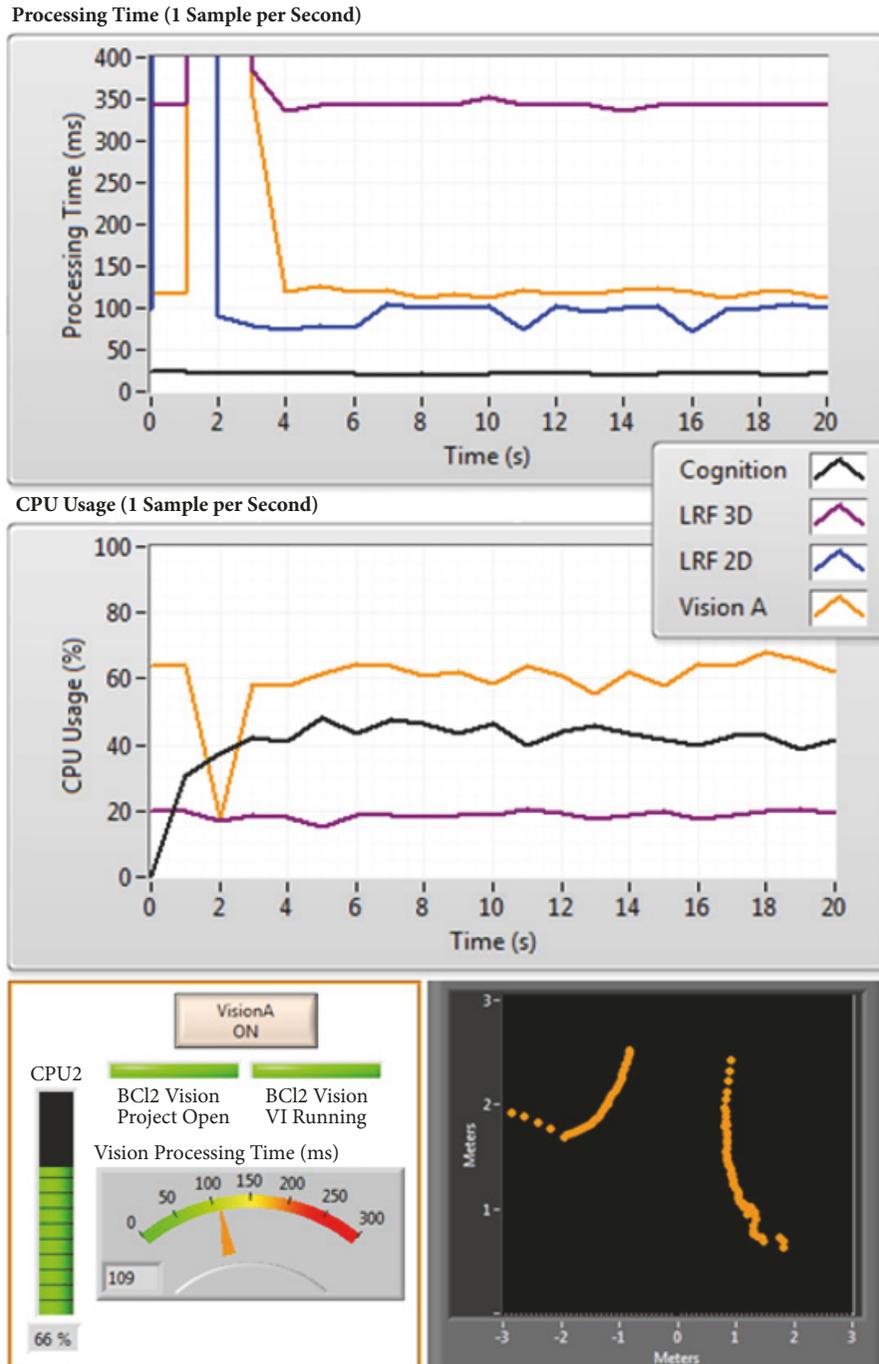


FIGURE 7: Performance monitor and user interface front panel display for three computers. Top: program processing time trace, Middle: computer CPU usage trace, Bottom left: simplified user interface with program status and Bottom right: camera vision distance histogram [9].

real time operation. Each portion of the distributed programs on a single computer is considered independently. The system as a whole is tested for the limiting factors in the overall performance of the cluster processing. The test results indicate that the client computer program performance and the dynamic 2D planer scan LRF program operates above 10Hz. Dynamic 2D refers to the scanning and processing of a single planer scan from the LRF. The LRF scanning plane is

rotating around an additional pitch angle during the scanning process. The robot vision program is not able to operate above 10Hz, which is the maximum hardware resolution. Therefore, additional testing is conducted to determine the benefits of image segmentation on vision performance. Furthermore, the 3D LRF scanning is limited to less than 10Hz by the LRF hardware. The 3D LRF scanning process refers to the combination of a full sweep of single plane scans, and the

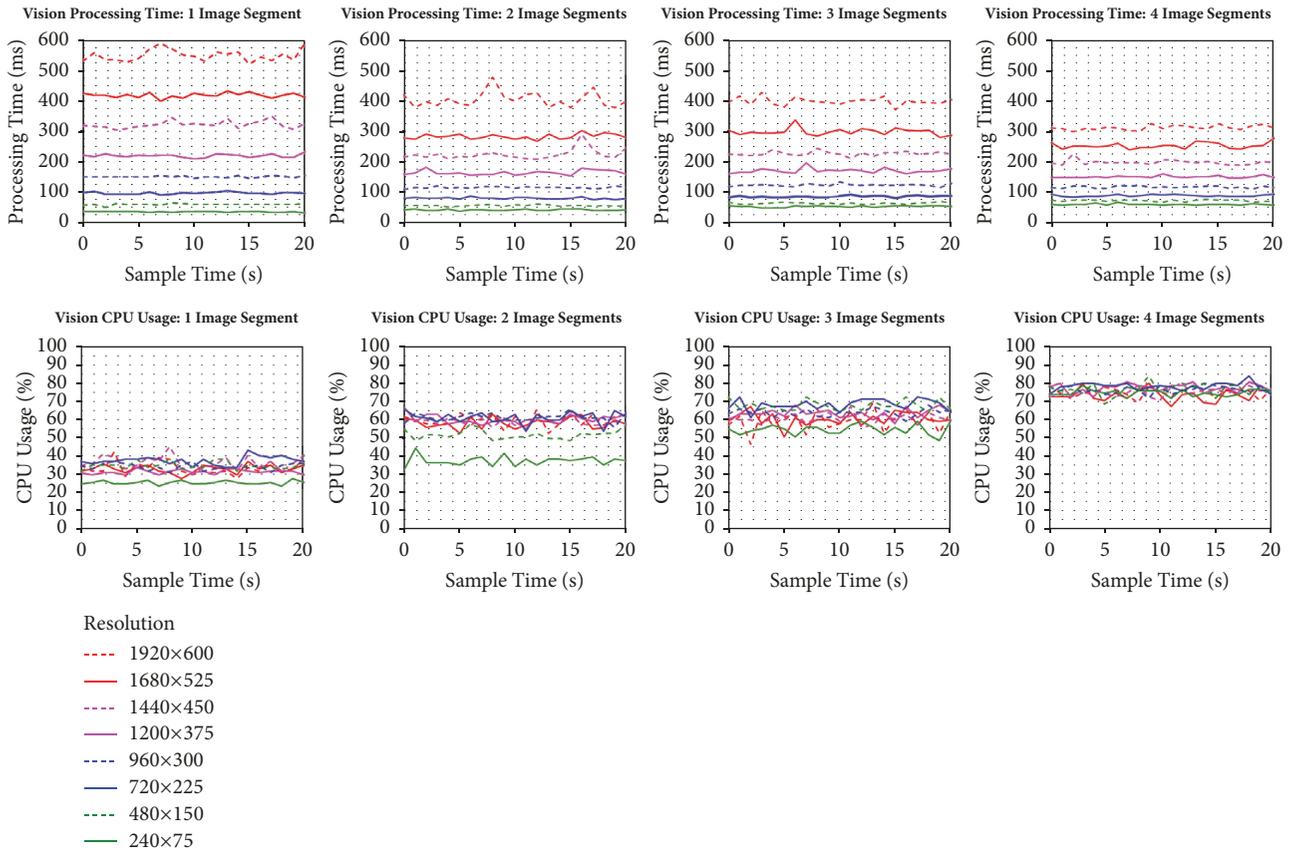


FIGURE 8: Vision program processing time (top) and vision computer CPU usage (bottom) of 20 sec samples [9].

stacking and processing of the combined data sets into a 2D representation of nearest obstacle edges.

**6.1. Evaluation Metric.** The real time performance of the cluster is considered for program processing time and computer CPU usage. The processing time includes the capture of a frame from the perception sensors, combining perception sensor data into a local map, instantaneous path determination and calculating necessary wheel speeds for motion control execution. Low processing time is critical for smooth real time operation of the smart wheelchair. The desired processing time for the smart wheelchair with an operating speed up to 4 MPH is 100 milliseconds (ms), or ten program iterations per second.

The CPU usage of each computer in the cluster is analyzed to determine the processing requirements of the programs. Increased CPU usage can represent improved task parallelism, while decreased CPU usage indicates less data processing or improved program efficiency. Accurate reading of the computer CPU usage for the methods used in LabVIEW is limited to one sample per second. While the processing time can be sampled effectively at a much higher frequency, the processing time sample rate is limited to one sample per second to correlate to the CPU usage.

**6.2. Performance Monitor.** A performance monitor is designed to show the status of computer cluster processing

in a consolidated display for smart wheelchair real time operation. A LabVIEW front panel is created to show CPU usage, processing time and software activity of the individual computers in the cluster. The performance monitor is used for system evaluation during testing and development. Also the portions of the monitor are integrated into the user interface.

**6.3. Startup Delay.** There is a measurable startup delay between activating the distributed LabVIEW programs on the cluster and receiving updated information across the LAN. The delay using network automation and activating autonomous navigation from the client computer interface is within four seconds (Figure 8). After four seconds, normal runtime operation of the computer cluster is expected for autonomous navigation. The startup delay on the individual computers in the cluster is tested to be less than two seconds. For the start to stop operation of the smart wheelchair and switching between autonomous and manual control modes, the programs are kept running in the background to reduce the startup delay to a negligible amount. This comes with the cost of power consumption and therefore, reduced smart wheelchair operating time.

**6.4. Runtime Performance.** The runtime performance of the cognition, LRF and vision programs vary considerably (Figure 8). The cognition program on the client computer

TABLE 1: Vision processing near 10Hz operating frequency.

Image Segments	720x225			960x300		
	CPU Usage (%)	Processing Time (ms)	Operating Frequency (Hz)	CPU Usage (%)	Processing Time (ms)	Operating Frequency (Hz)
1	37	98	10.2	35	153	6.6
2	62	81	12.3	62	116	8.6
3	69	88	11.4	64	124	8.1
4	78	90	11.1	77	118	8.5

including path planning, motion control, and the user interface, is able to achieve a processing time of approximately 20ms. The cognition program uses approximately 40% of the client computer CPU during operation. Figure 8 shows the vision program processing time and vision computer CPU usage of 20 second samples for one, two, three, and four image segments at eight different dual camera resolutions for one vision computer.

The 2D LRF processing time is approximately 90ms, achieving 11Hz operating frequency, and the 3D LRF processing time is approximately 350ms, achieving only 3Hz operating frequency. The 2D LRF data is processed as part of the 3D LRF functionality and the 3D LRF utilizes 20% of the LRF computer CPU. The processing time and CPU usage of the vision program varies depending on image resolution and the number of parallel image segments. The general trend for the vision program resolutions ranging from 240x75 to 1920x600 is, an increase in resolution increases the processing time and an increase in image segments decreases the processing time. The benefit of reduced processing time from image segmentation is most significant at the highest tested resolution of 1920x600. At this resolution one image segment requires over 0.5 seconds to process a single image frame. Increasing the number of image segments to four, results in a decrease of approximately 43% in the average processing time to about 300ms. However, this processing time is well above the desired 100ms for smooth smart wheelchair operation.

At lower resolutions, the benefits of image segmentation decrease completely. At the lowest three tested resolutions, 720x225 and below, increased image segmentation negatively which impacts the processing time performance. This resulted in more image threads increasing the processing time. This presumably is due to the effect of processing overhead of segmenting and combining the images in addition to the rest of the image-processing pipeline. This can result in the minimum processing time of 37ms on average occurring at lowest resolution of 240x75 with only one image thread.

The CPU usage varies from about 35% to about 75% where the overall resolution has little effect on the CPU usage. Isolating the resolutions for processing time around 100ms yields the two resolutions of 720x225 and 960x300 (Table 1). With a resolution of 960x300 the processing time averages close to 120ms for two image segments (Figure 8), with one image segment performing significantly slower. Increasing the image segments above two shows no significant improvement. The 960x300 resolution configuration results

in a 0.27 megapixel resolution operating at 8.5Hz for each vision computer, or 0.54 megapixel image processing using two vision computers in the cluster.

To achieve an operating frequency of at least 10Hz, the lower resolution of 720x225 is used for the current smart wheelchair operation. For a resolution of 720x225, the fastest average processing time of 81ms for the sample test is achieved with 2 image segments. Since a higher number of image segments produces no improvement on the processing time but does increase the computer CPU usage, two image segments is preferable. This configuration achieves 0.3 megapixel resolution image processing between two vision computers at over 12Hz operating frequency.

## 7. Conclusion

Advances are made on the technology of smart wheelchairs with sensors and driven by intelligent control algorithms to minimize the level of human intervention. The presented vision-based control interface allows the user to adapt and command the system at various levels of abstraction. The cognition motion control algorithms and the master remote control of the drive train is integrated in the control interface. The cluster consists of the distributed algorithms, the performance monitor and the network automation code that are executed from a single computer. With the research presented, the real time image processing is the limiting factor for processing speed in the current cluster configuration. From the results of the parallel processing capabilities of LabVIEW and the eight processing threads on the Intel I7 hyper-threading CPU, the task parallelism for the vision system can improve the CPU usage up to 80%. Currently in the demonstrated design, a dedicated computer is utilized for the LRF data processing cluster configuration that can be optimized. From the simulation results, it is predicted that combining the LRF algorithms onto the already fast performing cognition client computer can reduce the four computers to three. This modification is likely to achieve the same processing time results across the cluster. To improve the task parallelism of the vision algorithms or to provide space in the cluster for new algorithm development, the fourth computer could be utilized.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

The authors would like to show sincere gratitude to Dr. C. T. Lin for his continuous support, motivation, immense knowledge and enthusiasm in supporting the intelligent wheelchair research at California State University Northridge. This project would have been difficult to complete without his contribution and dedication.

## References

- [1] H. Soh and Y. Demiris, "Learning assistance by demonstration: smart mobility with shared control and paired haptic controllers," *Journal of Human-Robot Interaction*, vol. 4, no. 3, p. 76, 2015.
- [2] H. Yuen, J. Pineau, and P. Archambault, "Automatically characterizing driving activities onboard smart wheelchairs from accelerometer data," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5011–5018, Hamburg, Germany, September 2015.
- [3] M. Ghorbel, J. Pineau, R. Gourdeau, S. Javdani, and S. Srinivasa, "A decision-theoretic approach for the collaborative control of a smart wheelchair," *International Journal of Social Robotics*, vol. 10, no. 1, pp. 131–145, 2018.
- [4] C. Huang, Z. Wang, G. Chen, and C. Yang, "Development of a smart wheelchair with dual functions: Real-time control and automated guide," in *Proceedings of the 2017 2nd International Conference on Control and Robotics Engineering (ICCRE)*, pp. 73–76, Bangkok, Thailand, April 2017.
- [5] F. Utaminingrum, M. A. Fauzi, R. C. Wihandika et al., "Development of computer vision based obstacle detection and human tracking on smart wheelchair for disabled patient," in *Proceedings of the 2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–5, Dubai, United Arab Emirates, August 2017.
- [6] S. A. Sheikh and D. R. Rotake, "An evolutionary approach for smart wheelchair system," in *Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCSP)*, pp. 1811–1815, Melmaruvathur, India, April 2015.
- [7] F. Utaminingrum, T. A. Kurniawan, M. A. Fauzi et al., "A laser-vision based obstacle detection and distance estimation for smart wheelchair navigation," in *Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pp. 123–127, Beijing, China, August 2016.
- [8] A. Hartman, R. Gillberg, C. T. Lin, and V. K. Nandikolla, "Design and development of an autonomous robotic wheelchair for medical mobility," in *Proceedings of the 2018 International Symposium on Medical Robotics (ISMR)*, pp. 1–6, IEEE, Atlanta, GA, USA, March 2018.
- [9] A. Hartman, *Design of a smart wheelchair for hybrid autonomous medical mobility [Thesis]*, 2018, <http://scholarworks.csun.edu/handle/10211.3/201000>.
- [10] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey," *Journal of Rehabilitation Research and Development*, vol. 37, no. 3, pp. 353–360, 2000.
- [11] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development*, vol. 42, no. 4, pp. 423–435, 2005.
- [12] J. Leaman and H. M. La, "A comprehensive review of smart wheelchairs: past, present, and future," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–489, 2017.
- [13] C. T. Lin, C. Euler, P. Wang, and A. Mekhtarian, "Indoor and outdoor mobility for an intelligent autonomous wheelchair," in *Proceedings of the 13th ICCHP, International Conference on Computers Helping People with Special Needs*, 2012.
- [14] C. A. Rockey, E. M. Perko, and W. S. Newman, "An evaluation of low-cost sensors for smart wheelchairs," in *Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering, CASE 2013*, pp. 249–254, USA, August 2013.
- [15] S. Desai, S. S. Mantha, and V. M. Phalle, "Advances in smart wheelchair technology," in *Proceedings of the 2017 International Conference on Nascent Technologies in Engineering, ICNTE 2017*, pp. 1–7, Navi Mumbai, India, January 2017.
- [16] R. C. Simpson, D. Poirot, and F. Baxter, "The hephaestus smart wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 118–122, 2002.
- [17] P. Wang, N. R. Keyawa, and C. Euler, "Radial polar histogram: obstacle avoidance and path planning for robotic cognition and motion control," in *Proceedings of the SPIE 8301, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, Burlingame, Calif, USA, 2012.
- [18] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," RFC Editor RFC1918, 1996, Available: <https://tools.ietf.org/html/rfc1918>.
- [19] <http://www.ni.com/white-paper/4679/en/>.
- [20] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2014.
- [21] D. An and H. Wang, "VPH: A new laser radar based obstacle avoidance method for intelligent mobile robots," in *Proceedings of the WCICA 2004 - Fifth World Congress on Intelligent Control and Automation*, pp. 4681–4685, Tsinghua University, China, June 2004.
- [22] J. Gong, Y. Duan, Y. Man, and G. Xiong, "VPH+: An enhanced vector polar histogram method for mobile robot obstacle avoidance," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, pp. 2784–2788, Beijing Institute of Technology, China, August 2007.
- [23] National Instruments, "Reference example of a system monitor displaying CPU and memory usage," Available: <http://www.ni.com/example/31299/en/>, 2011.

