5

Stereo Vision Utilizing Parallel Computing for the Visually Impaired

Pichaya Tandayya, Thanathip Limna and Nikom Suvonvorn
Intelligent System (iSys) Research Team, Department of Computer Engineering
Faculty of Engineering, Prince of Songkla University
Thailand

1. Introduction

Many technologies have been developed to help the visually impaired. For example, a walking stick with an ultrasonic sensor can detect whether there are obstacles ahead, but cannot judge the distance, size, and position of the objects. A Differential Global Position system (DGPS) (Hashimoto et al., 2001 and Helal et al., 2001) with a mobile phone Electronic Sensory System for the visually impaired (ESSVI) (Ando, 2003) can find a destination, but says nothing about the safety of the surrounding area.

Ultrasonic or infrared devices can be attached to walking sticks or glasses, to detect obstacles in the environment and calculate distances, but do not provide size and position information for continuing a journey.

Stereo vision technology can help by finding the distance between an object and the stereo cameras, and also its size and position.

In a stereo vision system, two cameras represent the 3D view as seen by human eyes. Stereo disparity and depth estimation are determined by the difference in the positions of two corresponding points in the stereo images.

Stereo matching algorithms can be classified into two categories: Intensity-based Stereo Matching (ISM) using the intensity profile of each image for finding the disparity (Owens 2009), and Feature-based Stereo Matching (FSM) for finding the disparity using features in each image, such as edges, lines, and corners (Owens 2009). Generally, FSM cannot provide a proper disparity for images with featureless surface objects such as whiteboards, doors, and television monitors, while ISM fares much better. The tradeoff is that ISM requires intensity information for each pixel, which demands longer processing time.

The Electro Neural Vision System (ENVS) is a stereo vision application which presents obstacles and distances via signal alerts sent to the fingers (Meers & Ward 2004). If an object is close by then a signal with a high frequency is sent, while a signal with a lower frequency is sent for a distant object. However, ENVS employs FSM which cannot detect objects with featureless surfaces.

The ISM technique can detect objects with featureless surfaces, but applying the technique in real time requires complex and time consuming image processing, especially when accuracy is required. However, PCs are getting more powerful, often having 2-core or 4-

core CPUs. Unfortunately, sequential processing cannot fully utilize the maximum performance of multi-core CPUs, and parallel processing techniques are therefore necessary. This chapter describes a stereo vision system for the visually impaired, employing ISM to detect objects with almost any kind of surface.

Most image processing algorithms for stereo vision can be categorized as Single Instruction Multiple Data (SIMD), often utilizing threads and processes. With threads, it can be very difficult to predict the order of tasks and the number of run-time tasks. Alternatively, parallel programming using a Message Passing Interface (MPI) has better process management, follows mature standards, and offers robust implementations (MPI standard, 2009).

MPICH is an MPI implementation using shared memory which naturally maps to the Symmetric Multiple Processors (SMPs) architecture (MPI CH, 2009) used in our stereo vision system. The code can be easily recompiled on other device configurations and architectures. Our work is novel in that parallel computing has never previously been combined with the ISM technique to reduce computing time. This chapter investigates the computing time reductions possible when applying the parallel computing approach to a Depth Discontinuities Pixel-to-Pixel stereo (P2P) algorithm (Birchfield & Tomasi, 1998), running on a 2-core PC and an 8-core server.

Our work applies parallel computing using MPI, ISM techniques, and off-the-shelf multi-core computers to reduce computing time. We also investigate the calibration of low-cost stereo cameras using webcams.

The next section will present background on stereo vision, depth discontinuities algorithms, and the message passing interface. Section 3 describes the system design, including a system overview, low-cost stereo cameras, object distance estimation, and enhanced P2P using MPI. Experiments are described, and results analyzed.

2. Background

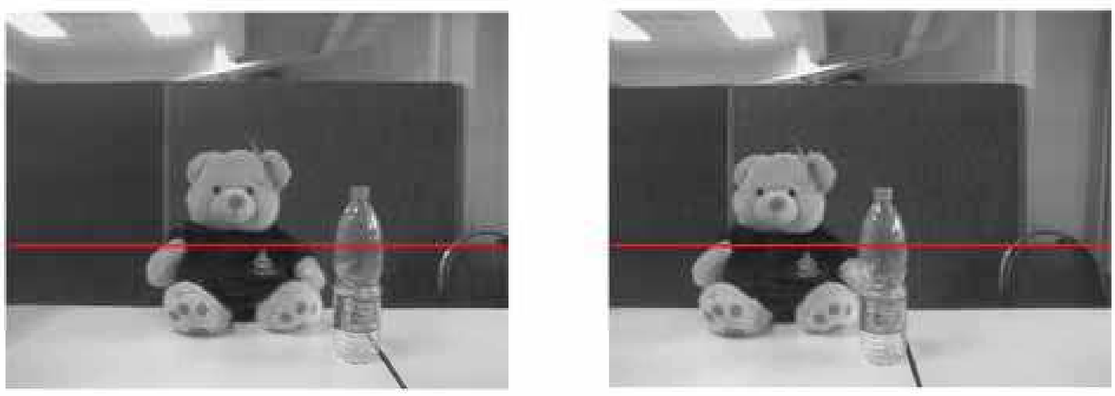
We provide some background on stereo vision using ISM, and parallel computing using MPI. Also, the sequential computing performance of P2P is investigated.

2.1 Stereo vision using intensity-based stereo matching (ISM)

In FSM, features of the matching process are applied to features extracted from the stereo images. In ISM, the matching process is directly applied to the intensity profiles of two images.

We are interested in the ISM technique since it can find a disparity even in featureless surface objects. In particular, we employ Depth Discontinuities by Pixel-to-Pixel Stereo (P2P), developed by Birchfield and Tomasi, and available within OpenCV, an open source framework for computer vision (CVAUX, 2009).

Fig. 1. Example of the P2P scan line matching - matching pixels from left and right images (Birchfield & Tomasi, 1998).



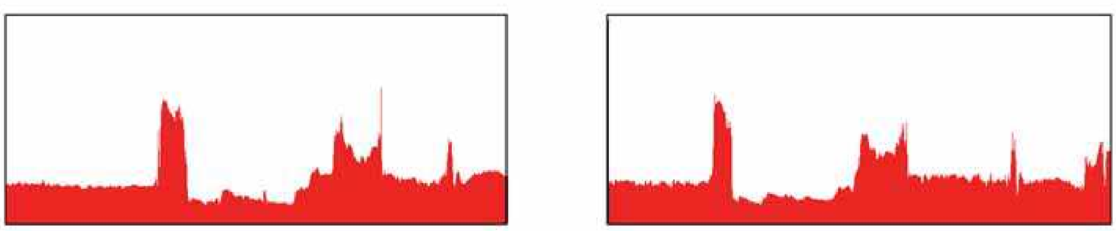


Fig. 2. Intensity of a pair of scan lines.

The P2P algorithm processes left and right images projecting on the same scan lines (Figure 1) in such a way that the processing of each pair of scan lines is independent and also relatively easy to parallelize. Figure 2 shows an example of an intensity comparison of a pair of scan lines. However, the post-processing algorithm needs to combine data from both rows and columns, which is harder to parallelize. P2P matches pixels on paired scan lines by applying a cost function (Birchfield & Tomasi, 1998) (given as Equation 1) to find an M sequence that represents the scan line being considered.

$$\gamma(M) = N_{occ} K_{occ} - N_m k_r + \sum_{i=1} d(x_i, y_i) \tag{1}$$

K_{ooc} is an occlusion penalty constant, k_r the match reward constant, $d(x_i, y_i)$ the distance between pixel x_i and pixel y_i, N_{occ} is the occlusion, and N_m is the number of related pairs.

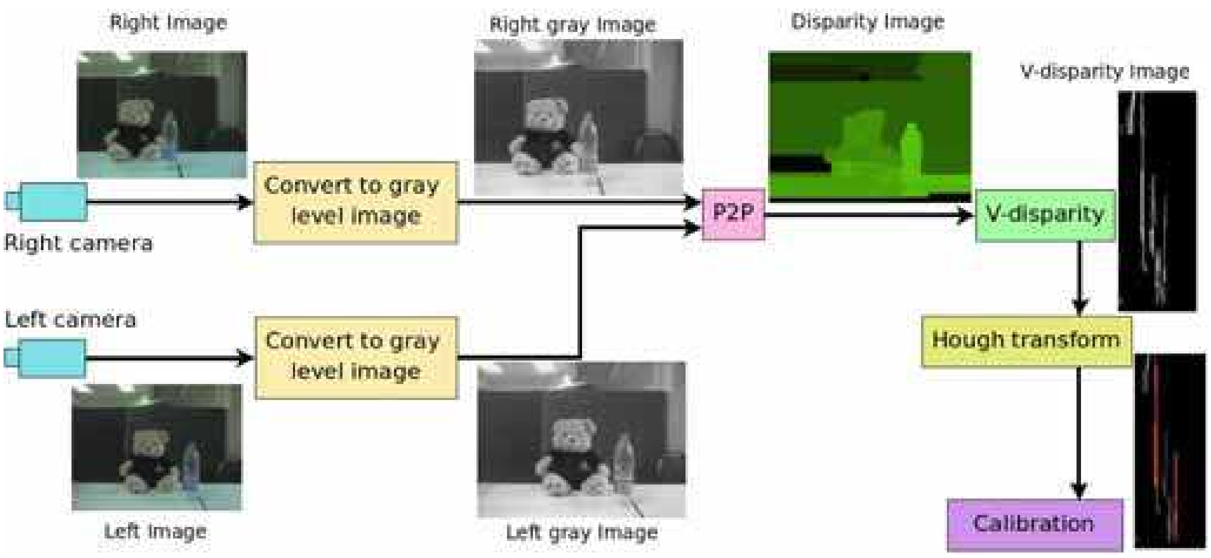


Fig. 3. Stereo vision processes (Limna et al., 2009).

Figure 3 shows the stereo vision processes. First, the color images from the left and right cameras are transformed into gray-level images to speed up subsequent processing and remove unnecessary information. The disparity image created by the P2P algorithm is a binary image containing pixels that define a relationship between the left and right images. The disparity image (see Figure 4) is used to calculate a summary of the disparity values in each scan line (called a V-disparity image). Then, depth lines are extracted from the V-disparity (Labayrade et al., 2002) using a Hough transform, and compared with the V-disparity to obtain object distances.

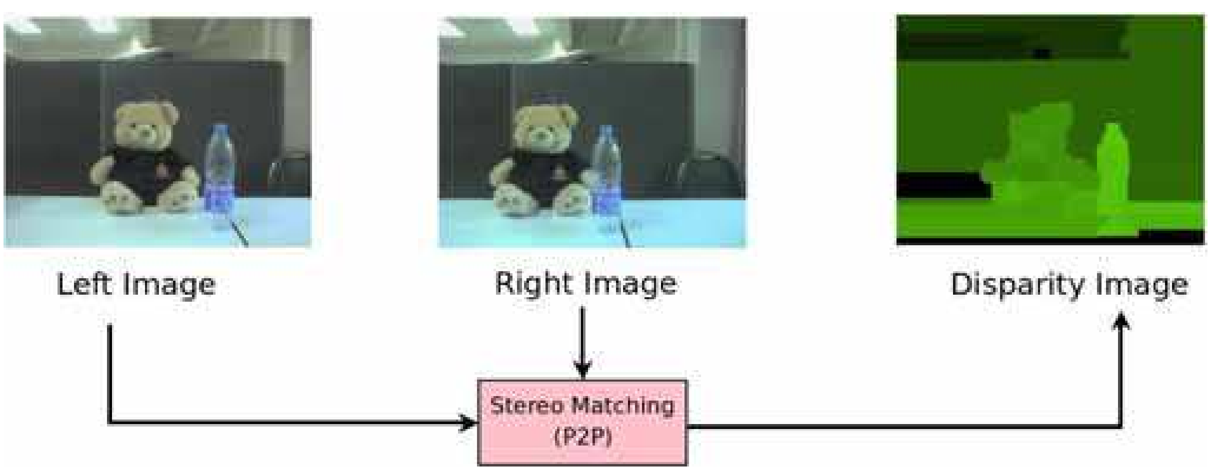


Fig. 4. Example left, right, and disparity images. (Limna & Tandayya, 2009)

Parallel computing can help reduce the processing time at several stages, such as for edge detection and scan line matching.

2.2 Parallel computing using message passing interface (MPI)

The Message Passing Interface (MPI) is a specification which facilitates parallel processing by distributing tasks and data amongst distributed processing units or processors (MPI standard, 2009). MPI is applied widely in computational sciences, where the efficient analysis of high quantities of data is required; e.g., finding the relationships between base sequences in human DNA, preparing a drug formula for destroying cancer cells, animation, and image processing.

MPICH is an MPI implementation (MPICH Document, 2009) that supports several different computer architectures by providing a variety of devices:

- a ch_p4 device for Workstation Networks;
- a ch_p4mpd device for Workstation Networks and Clusters;
- a ch_shmem device for Shared Memory Processors;
- a globus2 device for Grids.

The most suitable device for our work is ch_shmem for multi-core CPUs which share memory, including PCs, laptops, and multi-core servers.

2.3 Sequential computing performance of P2P

There are two processes in the P2P algorithm: scan line matching and post-processing. Scan line matching matches corresponding pixels in the left and right images on the same scan lines. In each scan line, processes match corresponding pixels independently from one another. Post-processing exchanges data between the scan lines in order to select the best disparity image. This requires data across rows and columns, and is not suitable for parallelization. We tested the sequential P2P algorithm on an Intel® Core™ 2 Dual 6320 1.86 GHz 1010.7 MB RAM, running Linux kernel 2.6.26. With a maximum disparity of 100 and an image size of 320x240 pixels, the average computing times was 1.168 seconds for scan line matching and 0.165 seconds for post-processing. Figure 5 **Fig.** shows the experimental results of the sequential P2P algorithm. The scan line matching requires about 70% of the computing time, so its parallelization should benefit the P2P algorithm.

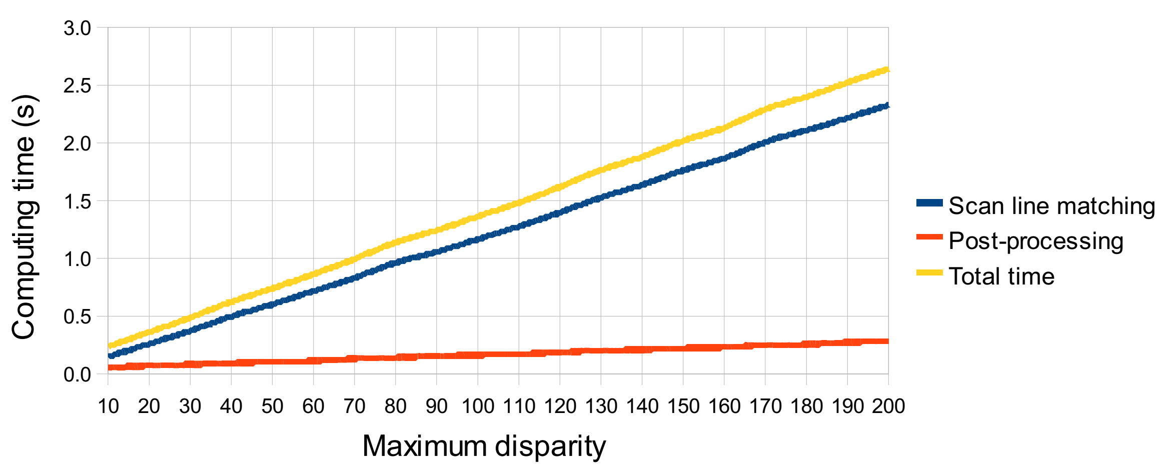


Fig. 5. The relationship between the computing time and the maximum disparity of the sequential computing P2P algorithm, running on a 2-core computer. The image size is 320x240 pixels (Limna & Tandayya, 2009).

3. System design and implementation

This section describes our stereo vision system including system overview, low-cost stereo cameras, and object distance estimation with V-disparity.

3.1 System overview

Our object detection system should issue a warning when there are obstacles 10 meters ahead, as shown in Figure 6**Fig.** . However, the most critical range is between 1 and 6 meters, which is outside a walking stick's range. Our aim is not to replace the walking stick but to augment it.

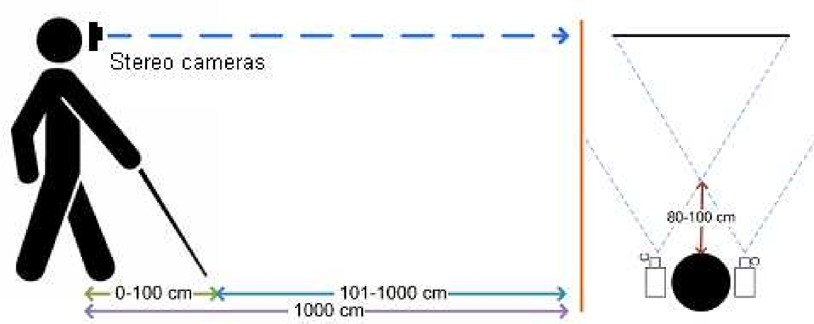


Fig. 6. System usage (Limna et al., 2009).

3.2 Low-cost stereo cameras

Normal stereo cameras are rather expensive, so we utilized cheaper web-cams instead- Logitech QuickCam web-cams for Notebooks Pro (Clark, 2009). On the downside, they are less accurate than normal cameras, and may produce more noise.
We plan to attach the cameras to a helmet, so their base line was 12 centimetres, which is an average head diameter. The design of the stereo cameras is shown in Figures 7 and 8.

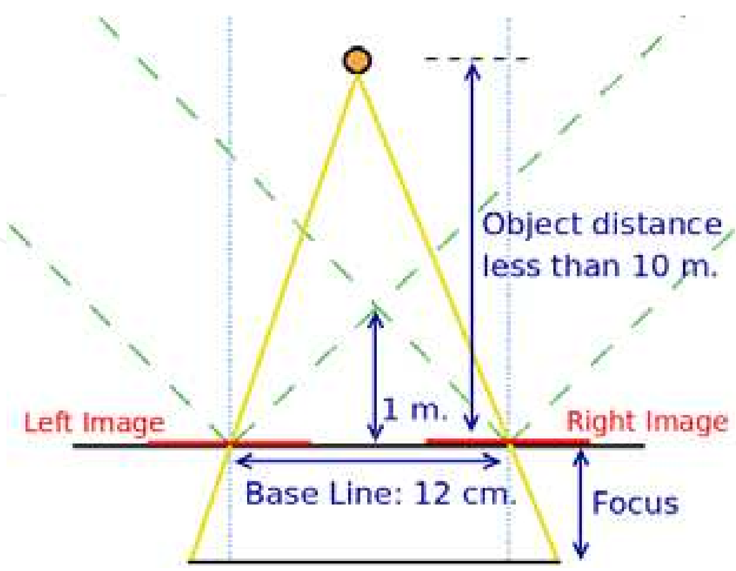


Fig. 7. Design of our stereo cameras (Limna et al., 2009).

Fig. 8. Our stereo web-cams.

3.3 Object distance estimation with V-disparity

An object's depth information can be calculated from slight offsets of projections of the object in both images (the disparity), the relative positions of both cameras (the base line), and the image resolution focal length.

The disparity image created by the P2P algorithm is a binary image that contains corresponding pixels from the left and right images. The V-disparity image is calculated from the summation of the disparities in each scan line. Each vertical straight line in the V-disparity image represents a depth distance. A Hough transform is used to find the depth lines in the V-disparity image, and the distance information in the depth lines are then compared with the V-disparity to find the object distance.

4. Parallel P2P algorithm

The scan line matching algorithm is suitable for parallelism because it independently computes each scan line. Figure 9 shows an example of MPI data distribution in the scan line matching. In each image frame, we divide the left and right image into two segments and distribute them to two processes. The first process computes the top parts of the left and right images, and the second process computes the bottom halves of the images. We then combine the outputs and create a disparity image.

The program is designed to run on a computer with more than two processors, and the user can specify the number of processes at run time. The MPI Scatter command is used for distributing portions of the left and right images to the processes, and each process manages the same number of scan lines. For example, if an image size of 320x240 pixels is assigned to 4 processes, then each process will match 80 rows of scan lines from the left and right images. After the matching is finished, the disparity image and depth discontinuities will be gathered together in process 0 before moving to the next step of the calculation.

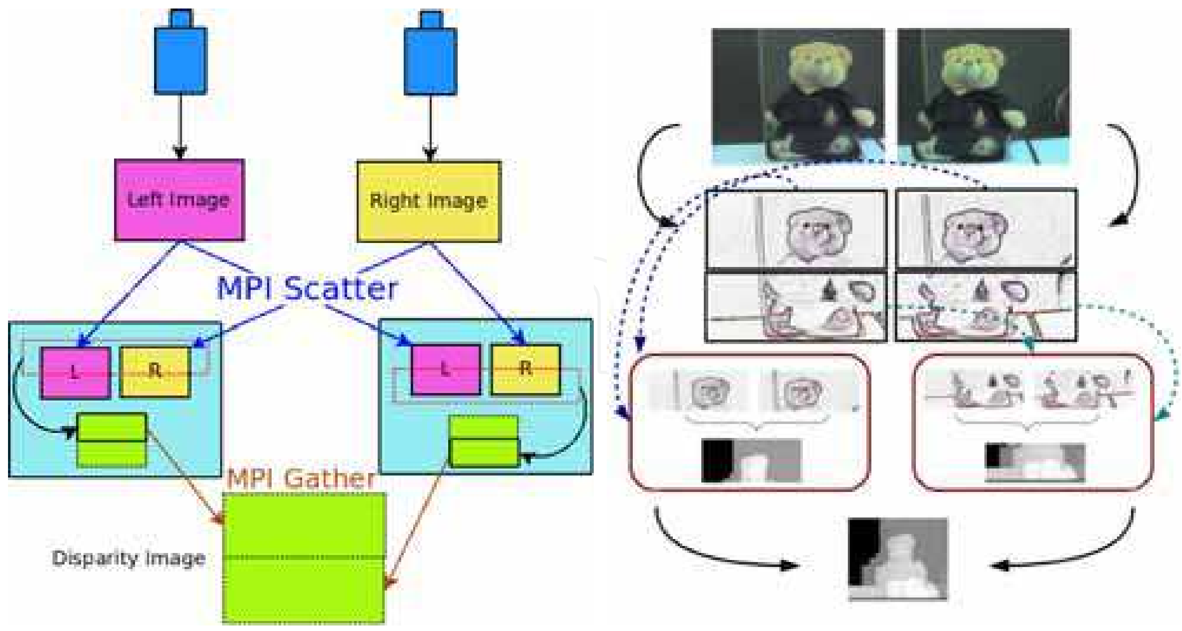


Fig. 9. Data distribution using parallel computing across two processes. (Limna & Tandayya, 2009)

We compared the sequential and parallel algorithms by testing them both on a 2-core PC and an 8-core server. The 2-core machine is an Intel® Core™ 2 Dual 6320 1.86 GHz 1010.7 MB RAM running Linux kernel 2.6.26. The 8-core server is an Intel® Xeon© E5440 2.83 GHz 8200 MB RAM also running Linux kernel 2.6.26.
The pseudo code for the parallel algorithm (Limna & Tandayya, 2009):

```
ROWS : number of rows in the image
id : process id
numProcs : number of processes
startIndex := ROWS*id/ numProcs;                    //starting scan line
endIndex := ROWS*(id+1)/numProcs;                   //finishing scan line

//scatter left and right images at Process 0 to other processes from startIndex to ROWS/
numProcs
MPI::Scatter(imgL[startIndex],COLS*ROWS/numProcs,...);
MPI::Scatter(imgR[startIndex],COLS*ROWS/numProcs,...);

for (scanline = startIndex; scanline < endIndex; scanline++)
scan line matching code

//gather disparity images and depth discontinuities images from all the processes into
Process 0
MPI:: Gather(disparity_map[startIndex],...);
MPI:: Gather(depth_discontinuities[startIndex],...);
```

The MPI Scatter and Gather commands handle data distribution and collection (MPI 2009). MPI Scatter divides an array into smaller parts equal to the number of processes and sends each to a process. MPI Gather collects data stored in all the processes into a receiving array.

5. Result and discussion

This section provides experimental results, a discussion of calibration and distance estimation, and describes the effectiveness of the P2P enhancement using MPI.

5.1 Calibration and distance estimation

This subsection concerns stereo camera calibration and object distance estimation using the V-disparity image.

5.1.1 Stereo camera calibration

The V-disparity image obtained from the disparity image contains several straight lines. Each line's length is the height of the associated object from the disparity image. Also its distance from the left edge is an inverse variation of the distance from the stereo cameras to the object. We can find the distance to the object from a V-disparity straight line by comparing it with the distances of the object measured in prior experiments.

Object distance (meters)	Pixel Position		
	Minimum	Medium	Maximum
1	42	43	44
1.5	31	32	34
2	26	26	27
2.5	21	23	24
3	20	21	21
3.5	19	19	20
4	18	18	18
4.5	17	17	17
5	16	16	16
5.5	16	16	16
6	15	15	15
6.5	15	15	15
7	15	15	15

Table 1. Matches between real object distances and pixel positions from the V-disparity by finding the medium (Limna et al., 2009)

Table 1 and Figure 10 show the relationship between the object distances and V-disparity line pixel positions (minimum, medium and maximum). Each distance range refers to the prior testing dataset, and the data is processed to find the medium for distance estimation. The matches between object distances and pixel positions in the V-disparity can be estimated by the 6^{th} order polynomial equation shown in Equation 2.

Figure 11 shows a graph plot of Equation 1 using the medium dataset. In Equation 2, d is the distance to the obstacle and x the length of the vertical line from the x-axis in the V-disparity.

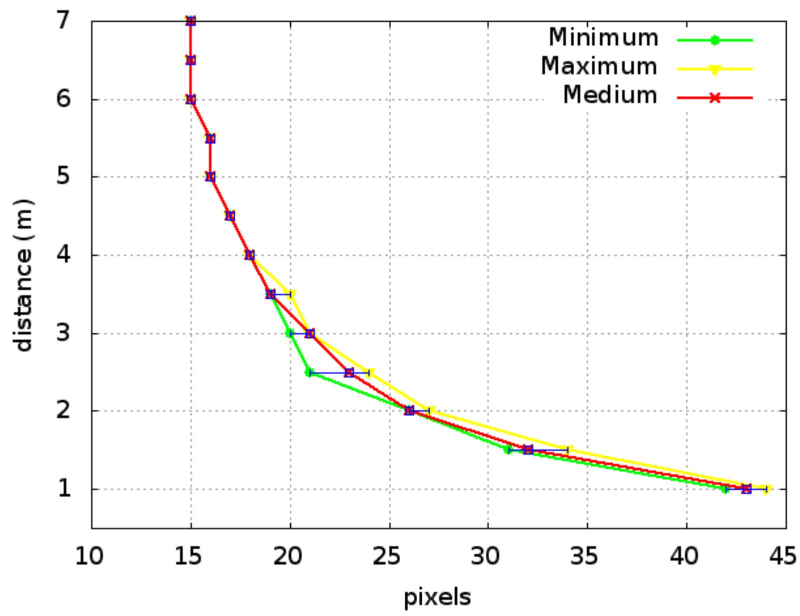


Fig. 10. V-disparity results.

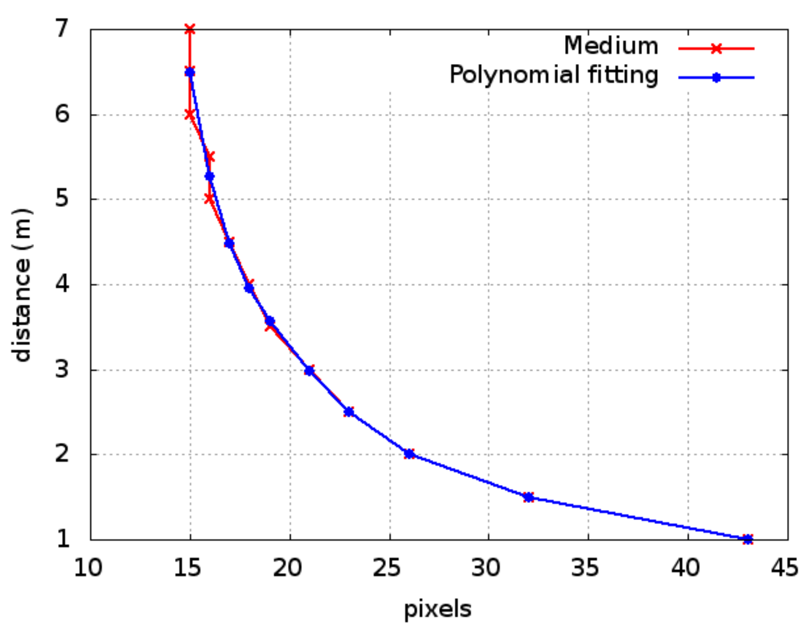


Fig. 11. Medium dataset fits for a 6th order polynomial equation. (Limna et al., 2009)

$$\begin{aligned} d = 1.7938\times10^{-6}x^{6} - 2.9101\times10^{-4}x^{5} + 1.9294\times10^{-2}x^{4} \\ -0.67064\times x^{3} + 12.925x^{2} - 131.57x + 559.10 \end{aligned} \quad (2)$$

5.1.2 Object distance estimation

We use the V-disparity from the P2P algorithm to find straight lines by employing an OpenCV function, cvHoughLine. It returns two sequence pairs, which are the beginning and ending points of the straight line. The x-axis value represents the object depths, and replaces the x in Equation 2 to obtain the object distance. Currently, our system can find object distances within 5 meters of the stereo cameras.

5.2 Enhancing P2P using MPI

The results for the sequential algorithm in Figure 5 shows that scan line matching takes up most of the computing time. The results for the parallel algorithm using MPICH with a shared memory device are shown in Figures 12 and 13.

5.2.1 Running on a 2-core computer

The parallelized algorithm reduces the total response time to 0.79 seconds at a maximum disparity of 100, applied to 320x240 pixels using two processes on a 2-core PC. By comparison, the total time of the sequential algorithm is 1.365 seconds per frame, indicating that the parallelized version can reduce the time by about a half.

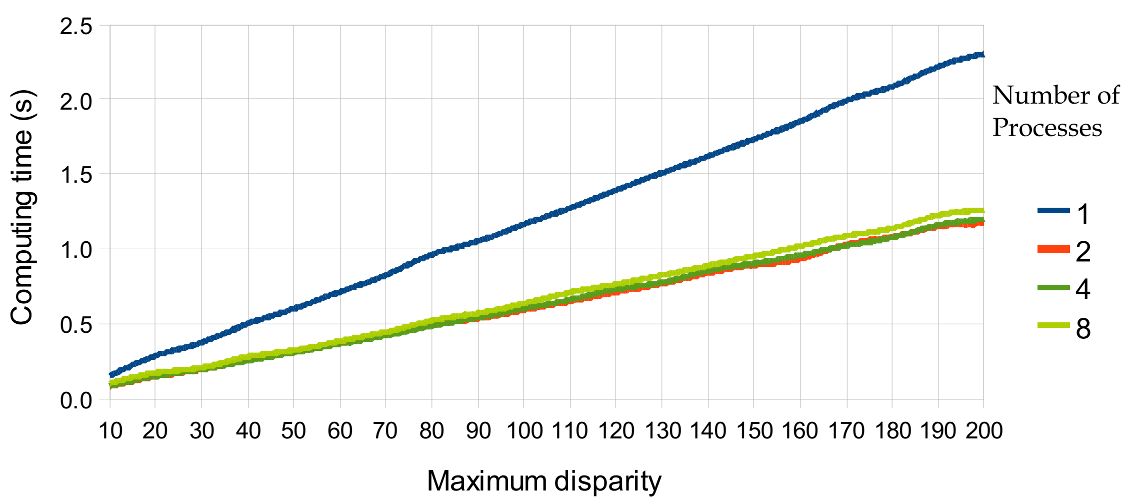


Fig. 12. The relationship between the computing time and the maximum disparity. The graph lines are the number of processes employed in the parallel computing scan line matching of the P2P algorithm using MPICH and a 2-core computer. The image size is 320x240 pixels. (Limna & Tandayya, 2009)

Figure 12 shows that data distribution on the scan line matching algorithm reduces the computing time if more than one process is used. However, when there are more than two processes, the computing time becomes longer than that for two processes. Using more processes than the number of CPU cores does not increase the speed of the parallel algorithm. The computing time for scan line matching at a maximum disparity of 100 is significantly reduced by 49.23%, or 0.575 seconds, which makes the algorithm suitable for use at run time.

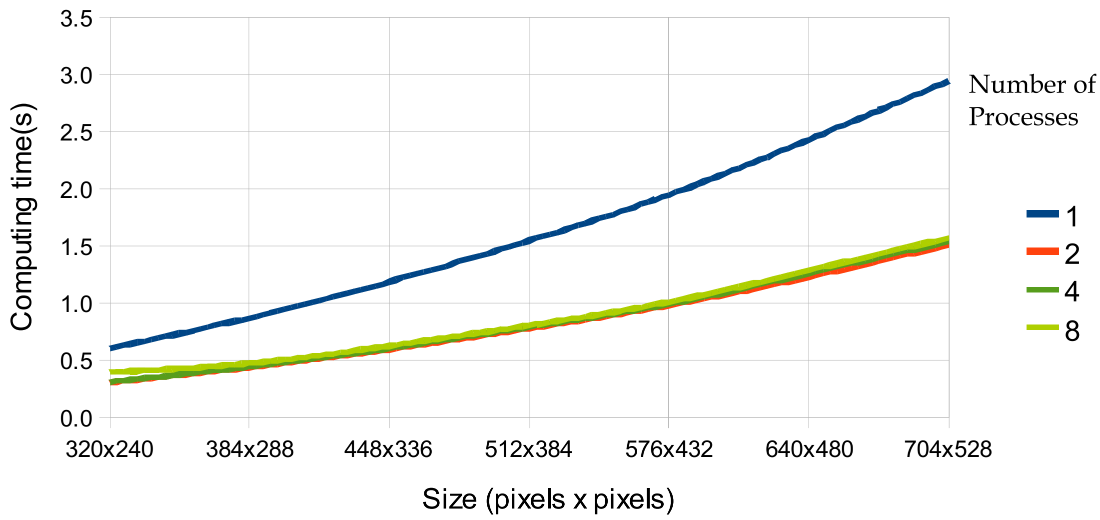


Fig. 13. The relationship between the computing time and the size of images. The graph lines are the number of processes employed in the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server (Limna & Tandayya, 2009).

In Figure 13, we vary the image size and number of processes and investigate the computing time when the image size and number of processes increases. The results show that the bigger the image size, the better the efficiency, and that the relationship between the speedup and number of processes remains the same.

5.2.2 Running on an 8-core computer

The results when running the sequential algorithm on the 8-core machine are shown in Figure 14. The results are similar to those on the 2-core computer - scan line matching is the main computing load. Figures 15 and 16 show the results for the parallel algorithm. The computing time is less when the number of processes is increased. From Table 2, the computing time for the parallel algorithm at a maximum disparity of 100 is reduced by 0.7792 seconds, or 88.21%, compared to the sequential algorithm on two processes.

Type of computer	Computing time (s)			
	Sequence algorithm	Parallel algorithm with		
		2 processes	4 processes	8 processes
2-core	1.1683	0.5932	0.6028	0.6369
8-core	0.8833	0.1041	0.0554	0.0285

Table 2. Average computing time of the serial and parallel algorithms at a maximum disparity of 100 on a 2-core computer and an 8-core server (Limna & Tandayya, 2009).

On the 2-core computer, at a maximum disparity of 100 and an image size of 320x240 pixels, the average sequential computing time of scan line matching is 1.168 seconds, while the average parallel computing time with two processes is 0.593 seconds. For the 8-core server, the average sequential computing time of scan line matching is 0.883 seconds, while the average parallel computing time with two processes is 0.104 seconds. Increasing the number of processes above two processes in the 2-core computer does not reduce the computing time. By contrast, increasing the number of processes in the 8-core server up to 8 significantly reduces the computing time.

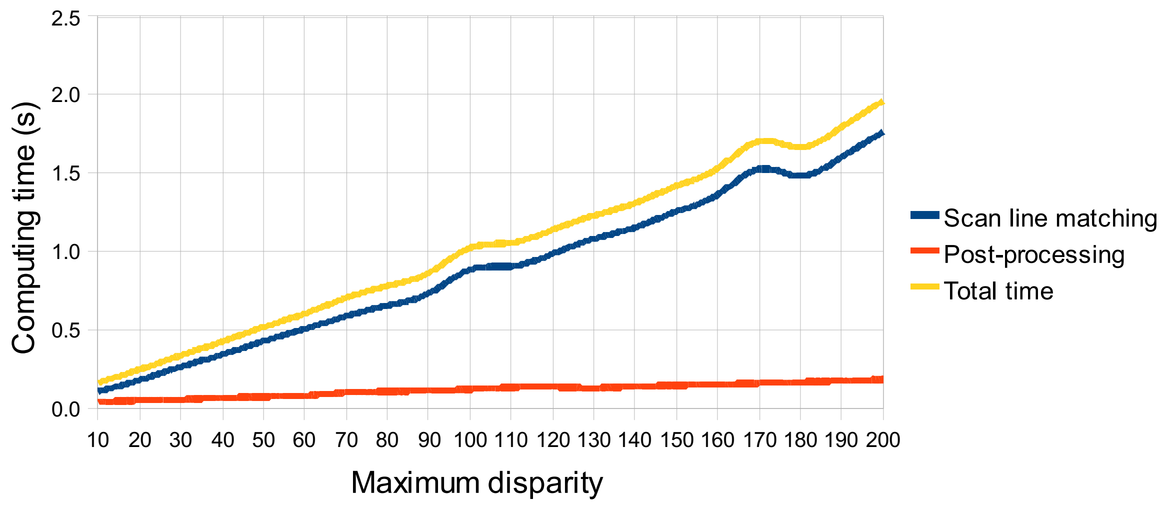


Fig. 14. The relationship of the computing time and maximum disparity of the sequential computing P2P algorithm on an 8-core server. The image size is 320x240 pixels (Limna & Tandaya, 2009).

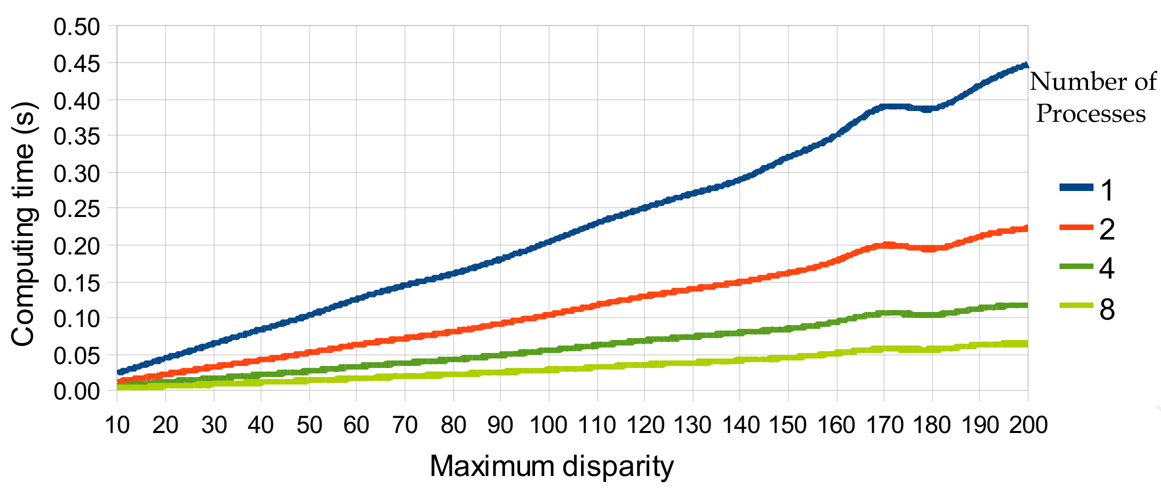


Fig. 15. The relationship between the computing time and the maximum disparity. The graph lines are the number of processes employed in the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server. The image size is 320x240 pixels (Limna & Tandaya, 2009).

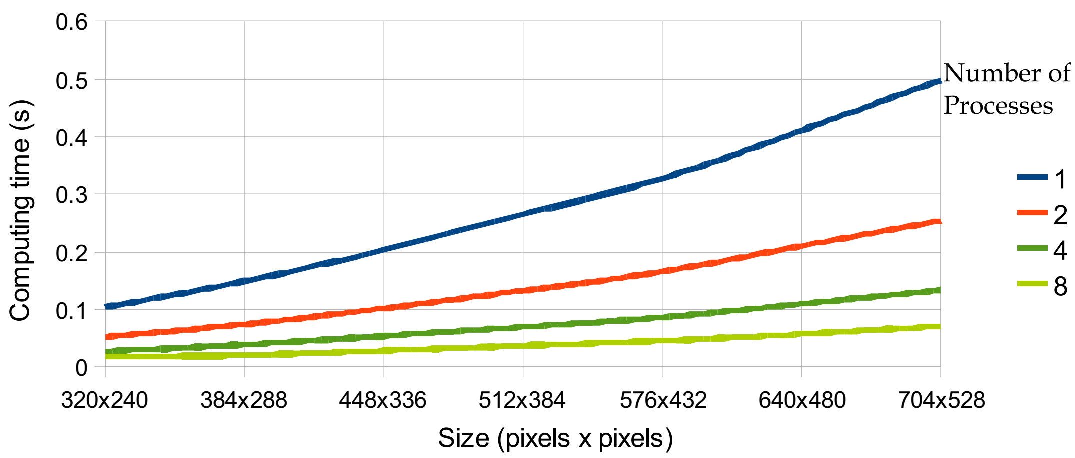


Fig. 16. The relationship between the computing time and the image size. The graph lines are the number of processes employed in the parallel computing scan line matching of the P2P algorithm using MPICH and an 8-core server (Limna & Tandaya, 2009).

5.3 Result analysis

In order to apply the parallel P2P algorithm in real-time, there are three factors to consider: the response time, the image size, and the disparity. A response time of less than 1 second is preferable. To detect obstacles within 1-6 meters of the cameras requires that the disparity is in the range of 100-150 so it can be computed within that response time. The best image size for this disparity lies between 320x240 and 512x384 pixels.

The results show that the number of processes run on a multi-core computer should not exceed the number of cores, corresponding to parallel computing theory.

5.4 Limitations

The P2P algorithm employs ISM techniques, which requires multiple parallel images so it can accurately detect obstacles. Our prototype utilizes stereo web-cams that have a maximum base line diameter of about 12 centimeters so they can be attached to a helmet (or glasses). Consequently, the prototype can accurately detect obstacles within a 5 meter range. For distances beyond 5 meters, the system's accuracy is variable. It can detect an object of 65x65 centimeters at a distance of 10 meters, but the image is so small that it may be lost amongst noise. Better quality cameras would increase the detecting distance and accuracy.

The application response time is 0.790 seconds for an image of 320x240 pixels at the maximum disparity of 100. It can detect slowly moving objects such as people walking, but cannot reliably detect quickly moving objects, such as cars, in real time.

6. Conclusion

Our obstacle detection system for the visually impaired uses stereo vision and parallel computing to reduce the response time of a depth discontinuities P2P stereo algorithm, by re-implementing scan line matching using MPI on a 2-core computer and an 8-core server. The object distances are found by employing V-disparity.

Our system can accurately detect slowly moving objects within a 5 meter range, using 12-centimeter base-lined low-cost web-cams. More development and detailed experiments will improve these results.
Our work shows that parallel computing using MPI on multi-core computers significantly reduces the computing time of the P2P depth discontinuities ISM algorithm, making it possible to detect obstacles in real time.
Planned future work includes a suitable interface for the visually impaired user, and a proximity warning system for dangerous objects. Applying stereo vision with pattern recognition can provide more details about the environment. When prices allow, increasing the number of cores on the computer, to 4 or 8, will reduce the computing time even further.

7. Acknowledgements

The authors are grateful for the support of the PSU Grid Center of Prince of Songkla University, the Thai National Grid Center, and the Intel Corporation. The authors are also thankful for Dr. Andrew Davison for proof reading.

8. References

Ando, B. (2003). Electronic Sensory Systems for the Visually Impaired. *Instrumentation & Measurement Magazine*, IEEE Vol. 6, Issue 2, (June 2003) page numbers (62-67)

Birchfield, S. & Tomasi, C. (1998). Depth discontinuities by pixel-to-pixel stereo, *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pp. 1073-1080, Mumbai, India, Jan. 1998

Birchfield, S. & Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 401-406, Apr. 1998

Clark, S. (2009). Logitech QuickCam for Notebooks Pro. *http://www.everythingusb.com/logitech_quickcam_for_notebooks_pro.html*, Available on 4 Jan. 2009

CvAux, *http://opencv.willowgarage.com/wiki/CvAux#FindStereoCorrespondence*, Available on 4 Jan. 2009

Hashimoto, H.; Magatani, K. & Yanashima, K. (2001). The development of the navigation system for visually impaired persons. *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* vol. 2, pp. 1481 - 1483, Istanbul, Turkey, Oct. 2001

Helal, A., Moore, S.E., Ramachandran & Drishti B., (2001). An integrated navigation system for visually impaired and disabled. *Proceedings of the Fifth International Symposium on Wearable Computers*, pp. 149 - 156, Zurich, Switzerland, Oct. 2001

Labayrade, R., Aubert, D. & Tarel, J.-P., Real time obstacle detection in stereo vision on non flat road geometry through "v-disparity" representation. *Proceedings of IEEE Intelligent Vehicle Symposium*, vol.2, pp. 646- 651, June 2002

Limna, T., Tandayya P. & Suvanvorn N., Low-cost Stereo Vision System for Supporting the Visually Impaired’s Walk, *Proceedings of 3rd international Convention on Rehabilitative Engineering & Assistive Technology (i-CREATe 2009)*, Singapore, Apr. 2009

Limna, T. & Tandayya P., Enhancing the Pixel-to-Pixel Depth Discontinuities Algorithm Using MPI, *Proceedings of 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009)*, Phuket, Thailand, May. 2009

Meers, S. & Ward, K. (2004). A vision system for providing 3D perception of the environment via transcutaneous electro-neural stimulation. *Proceedings of the Eighth International Conference on Information Visualisation*, pp. 546 - 552, ISBN 0-7695-2177-0, London, UK, July 2004, IEEE Computer Society

MPICH-A Portable Implementation of MPI, *http://www-unix.mcs.anl.gov/mpi/mpich1/*, Available on 4 Jan. 2009

MPICH Documents, *http://www-unix.mcs.anl.gov/mpi/mpich1/docs.html*, Available on 4 Jan 2009

Nicolescu, C. & Lecture, P. (2000). Parallel low-level image processing on a distributed-memory system, *Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing*, pp. 226 - 233, Cancun, Mexico, May 2000, Springer

Owens, R., (2009). Stereo, *http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/lect11.html*, Available on 4 Jan. 2009

Stereo vision start with two views, *http://www.vision3d.com/stereo.html,* Available on 4 Jan. 2009

The Message Passing Interface (MPI) standard. *http://www-unix.mcs.anl.gov/mpi/*. Available on 4 Jan. 2009

Rehabilitation Engineering

Edited by Tan Yen Kheng

ISBN 978-953-307-023-0
Hard cover, 288 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

Population ageing has major consequences and implications in all areas of our daily life as well as other important aspects, such as economic growth, savings, investment and consumption, labour markets, pensions, property and care from one generation to another. Additionally, health and related care, family composition and life-style, housing and migration are also affected. Given the rapid increase in the aging of the population and the further increase that is expected in the coming years, an important problem that has to be faced is the corresponding increase in chronic illness, disabilities, and loss of functional independence endemic to the elderly (WHO 2008). For this reason, novel methods of rehabilitation and care management are urgently needed. This book covers many rehabilitation support systems and robots developed for upper limbs, lower limbs as well as visually impaired condition. Other than upper limbs, the lower limb research works are also discussed like motorized foot rest for electric powered wheelchair and standing assistance device.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821