Computer Vision Techniques for Autonomic Collaboration between Mobile Robots

Catherine Saunders, Roy Sterritt, George Wilkie
School of Computing and Mathematics
University of Ulster
Northern Ireland
e-mails: {saunders-c2@email.ulster.ac.uk, r.sterritt@ulster.ac.uk, fg.wilkie@ulster.ac.uk}

***Abstract*— Autonomic Computing is concerned with improving the self-management and adaptability in software. In this paper, we propose an Autonomic concept that would allow robots to interact with each other using an alternative communication protocol in the event that direct wireless digital communications fail. The aim of the research is to create enhanced levels of autonomous behavior that enable the robots to still communicate via a more primitive mechanism. The alternative communication protocol uses a form of textual semaphore. The visual means of communication involves processing textual images that contain meaningful messages or instructions. These textual images are transmitted via a display screen mounted on the robot. The robot receiving the message does so by taking a photo and performing image processing. The research focus is on how robots could adapt to a hardware communications failure by using the technique outlined here. Our experiments have proven that a robot can read text from a screen and respond with an action, e.g., to rotate. Results also indicate that further work is needed in the area of image processing in order to improve the accuracy of the character recognition. It is hoped that these results will lead to further research that can build upon the ideas presented and seek to establish a two-way conversation as opposed to a monologue.**

Keywords-Autonomic Computing; Mobile Robot; Optical Character Recognition; Computer Vision

I. INTRODUCTION

The idea of a self-managing artificial entity is not new; it is a concept that has long existed in Science Fiction. As software systems become more complex, they require more management. As a solution, IBM devised Autonomic Computing [1]. Autonomic Computing seeks to remove human involvement and management of a system [2] and instead allows that system to monitor itself and adapt. Adaption is a fundamental aspect to an Autonomic system, being able to modify its state to cope with external or internal changes is vital to ensure it stays operational.

To be Autonomic or self-managing, a software system should possess awareness of its internal self and external environment. It should also be able to make changes to its state, and to adapt and repair itself [3]. Like the Autonomic Nervous System, an Autonomic system should be able to manage itself without outside influence.

An Autonomic system architecture consists of different layers, each with different responsibilities and awareness [4]. An immediate Reaction layer that can respond to imminent threats to the system; a Routine layer where most processing is performed; a Reflection layer that analyzes past and current experience in order to predict and improve future performance of either the routine or reactive activities [4]. A simple example of a process that could occur in the Reaction layer is that of a mobile robot using on-board sensors, such as ultrasonic or infrared to prevent a collision with an object.

Space robotics is an area that could greatly benefit from software that is Autonomic. The distance between Earth and other planetary bodies makes instantaneous teleoperation of rovers impossible, at least for the foreseeable future. Increasing the Autonomic capabilities of robots so that they may collaborate without the necessity of human intervention would be very advantageous. It would allow a greater number of robots to be sent and enable them to explore unaided.

This paper is part of an ongoing research project that aims to assess and explore areas where Autonomic Computing could be applied in Space Exploration. We discuss how image processing could be used to aid collaboration between entities in the event of a communications failure, i.e., as part of a Self-Healing strategy. Specifically, we look at how a mobile robot could observe and infer meaning from text displayed on the screen of another robot. The purpose is to assess whether meaningful communication can take place and how viable a method Optical Character Recognition (OCR) is for re-establishing communication between entities, where there has previously been a breakdown in the normal radio (wireless) communication. A benefit to having a robot display text as opposed to a symbol or QR code is that a human can also read the message.

Section II of this paper explains why Autonomic Computing is needed and how useful it could be for space exploration. Section III looks at related work involving robotics and text processing. Section IV introduces the research robot platform that was used for the experiments; the setup is also explained through the use of a diagram. Section V gives an overview of the image processing that was used in the experiments. Section VI discusses the algorithm that was used to enable a robot to detect text that is being displayed by another robot in its environment. The experimental results are discussed in Section VII. Section VIII summarizes the paper and discusses future work.

 ISBN: 978-1-61208-391-9

II. RESEARCH BACKGROUND

As a precursor to an eventual human landing on other planetary bodies, probes in the form of satellites and landers have been sent to explore and report. As early as the 1970s, the Soviet Space Program sent teleoperated rovers to the moon [5].

The Mars rover MSL Curiosity is one of many Martian rovers that are paving the way for an eventual human landing. Their overarching goal is to discover as much as possible in order to minimise the risk that is inherent with human space travel. It is hoped that future incarnations of Mars rovers will possess more autonomous behavior, thus making them less dependent on human operators to provide mission direction and navigational waypoints.

NASA's concept missions highlight their interest in eventually sending multiple spacecraft as opposed to one all-important rover. The current process of having a team micromanage a rover's actions would be untenable for a swarm or perhaps even a smaller cluster. The adoption of more ambitious projects like the NASA concept mission Autonomous Nano-Technology Swarm (ANTS) is reliant on the improvement of self-management techniques and the collaboration between multiple self-managing entities [6] [7]. Within a swarm of Autonomic entities, each would need to follow the Self-CHOP paradigm so that the entire swarm can operate efficiently [6][8]. Self-CHOP stands for Self-Configuring, Self-Healing, Self-Optimizing and Self-Protecting; it represents the key functions that an Autonomic system should possess.

Autonomic Computing is concerned with creating software that can self-manage its internal state without the need for human assistance [4][9]. For a system to be considered Autonomic, it must be Self-Aware, Environment Aware, Self-Adjusting and Self-Monitoring. In order to meet these requirements, it needs to monitor its internal and external state, be able to respond to stimuli and update its state. Autonomic Computing research breaks a software system down in to many components known as Autonomic Elements (AE). Each AE consists of an Autonomic Manager (AM) and a Managed Component (MC) that the manager controls via the MAPE control loop [1]. The MAPE control loop was devised by IBM in 2001, the acronym stands for Monitor, Analyze, Plan and Execute [1][10]. The MAPE control loop enables the AM to monitor and adjust the MC.

The AM also monitors the external environment for messages from other AEs. A communications channel exists between the AEs, this ability to exchange information from one AE to another AE is designed to help to improve the overall performance of a system, reduce failure and increase the ability of a system to self-repair.

Information that could be sent from AE to AE via this channel includes a Heartbeat Monitor (HBM) signal, which by its very presence could indicate whether or not another part of the system is still functioning [11][12]. The absence of an *"I am alive"* signal denotes that there is potentially a problem with the AE that is not transmitting. Within a purely software system, there are avenues available for repair and investigation that do not exist with separate mobile robots. With separation, their only means of collaborating as a whole involves transmitting data or communicating via emergence. If there is a fault with their hardware and they are unable to transmit, there is no way for the other members of a swarm or cluster to receive a status report from a faulty robot.

In this paper, we propose that OCR could be used as a backup communication strategy, with robots being able to glean information by reading the screen of a faulty robot. This proposal assumes that not all hardware on the robot is damaged and that it is still able to display messages on its screen. Other work involving OCR and robots has investigated using a standalone robot to observe text in the environment and mapping its location or following textual signs [13][14][15].

III. RELATED WORK

Great advances have been made in robot mapping, object detection and navigation. However, most robots are not equipped with the ability to read. For a robot to operate successfully in a human centric environment, they should be able to fully understand that environment. Analyzing the world around them should not be limited to mapping walls or navigation, as this does not provide a complete picture of where they are. Other forms of visual clue should be considered, perhaps for guidance or directions. Such forms would include reading QR codes or even reading forms of text. There is a limited amount of research on robots making use of OCR; this section will discuss research that combines robotics and OCR.

Using OCR to navigate an indoor environment by reading signs and without using Simultaneous Localization and Mapping (SLAM) has been investigated in [13]. SLAM is a research area that aims to solve the problem of how a mobile robot builds a map of an unknown environment and keeps track of its location within that environment. A robot was equipped with an RFID reader and each sign within the environment had an RF card associated with it. When the reader detected an RF card, a photo was taken and analyzed for text.

In [16], the benefits of using landmarks that occur in the human environment as opposed to artificial landmarks deliberately placed for the robot is discussed. The work presents a technique that enables a robot to recognise a door sign using a Histogram of Oriented Gradients [17] classifiers and then reading the text using Google Goggles [18]. The goal of the research is to create richer maps by gathering semantic information.

The idea that reading text can provide useful semantic information has also been explored in [19]. A robot was set the task of traversing a corridor and reading text of different sizes. It used a pan/tilt/zoom camera to zoom in on potential text. The ability to zoom in on the text improved the accuracy three-fold. *"To date robots have not been able to access this source of semantic information and often rely on* *a second tier of informational infrastructure such as QR codes of RFID tags"* [19].

OCR is prone to errors; Poster et al. [20] used probabilistic inference to compensate for detection errors and determine what a detected word was most likely to be e.g., "nqkio" is actually "nokia".

An edge detection algorithm is used by Liu and Samarabandu [21] to better identify characters within an image and allow a mobile robot to navigate. The edge detection algorithm is used to highlight blobs within a photo taken of a sign, bounding boxes are placed around the blobs and used to create sub images, which can then be passed to an OCR engine.

A robot that can read the license plate of a car is discussed in [22], where a mobile robot and Android phone were combined and used a number of image pre-processing techniques to remove noise from the image before OCR recognition took place.

An interesting use of a robot and OCR is presented in [23], the robot operates as a library assistant and can find and fetch books for a user. The user requests a book via a voice command; the robot then uses OCR to locate the book and picks it up using a robotic gripper arm.

Another useful OCR approach is described in [24], where the work involved aiding a blind person by using a shoulder mounted camera system that reads text within the environment and then dictates it to the user. Like the system in [19], it also uses a camera that can zoom in on potential text to get a higher resolution image.

This section has highlighted some examples of research that combines robotics and OCR. The majority of the research in this area tends to focus on improving SLAM and helping a robot navigate an environment by gathering semantic information. This paper presents a novel use for OCR that has not been covered before. We are interested in how two or more robots could communicate via OCR in the event of a communications failure.

IV. ROBOT PLATFORM

The experimentation was completed using the X80 [25] and X80-H [26] models manufactured by Dr Robot. A Samsung Galaxy Tablet was used to display text and mimic a screen that could belong to a robot. The X80 and X80-H are wheeled differential drive style robots that are equipped with a variety of sensors including Ultrasonic and Infrared. For these tests, the Ultrasonic sensors were used for collision prevention with other objects in the environment.

As shown in Figure 1, a separate computer runs the application that connects and controls the robot. The computer is connected via Ethernet to a router and data is sent to and from the robot via Wi-Fi. The application was developed in C# and utilises the Dr Robot API and EMGU CV library [27]. EMGU CV is a .NET wrapper for the widely used Open CV library[28].

The camera included on the X80 is subject to distortion, in future experiments we may opt to use a higher resolution IP WLAN camera that can be placed on the robot. The X80-H robot is equipped with a small mono LCD screen that can display 128x64 pixels Bitmaps.

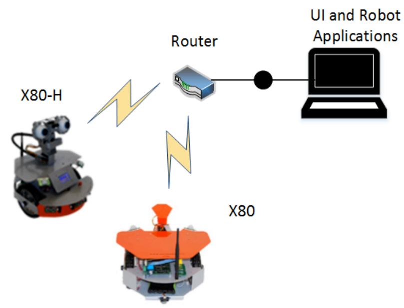

Figure 1. Setup involves robots sending/receiving data via Wi-Fi

We initially planned to use the LCD screen to display text but the X80 was incapable of reading the text due to the dullness of the screen and low quality X80 camera. For this reason, a tablet was chosen instead. Its brighter screen proved easier for the X80 camera to take images, from which text could be detected.

V. IMAGE PROCESSING

Image processing is a vast area of research within mobile robotics; its primary focus is to help with SLAM. However, the idea that robots could communicate by reading each other's screens is not a topic that receives much attention. This is because it is a relatively inefficient way to communicate compared to more traditional methods such as wireless communication, Infrared and Bluetooth. It is, however, interesting to discuss different approaches like OCR, or even speech recognition were messages broadcast from a speaker on a robot could be processed for words by another robot, because such means of communication may provide a useful backup system in the event that the primary mode of communications fails – thereby fitting our goal of creating adaptability within our robot capabilities.

For the experiments, we used the Tesseract library, which was initially developed by HP Labs and is currently maintained by Google [29]. Processing and analyzing the data within each video frame is computationally inefficient. Using color filtering and feature based matching reduces the amount of processing required. By only checking for characters within a Region of Interest (ROI), this eliminates the reading of extraneous data that may occur in the environment. To create a ROI, such as a rectangle, we can place a red rectangular frame around text; everything except the color red can then be filtered out. The next step is to look for contours with 4 edge points, draw a bounding box around the contour, and clip the X, Y, Width and Height of this bounding box from the original frame. The Tesseract OCR Engine is then used to look for text within this clipping and not the entire frame. By only processing the characters that appear within the red bounding box, this helps to cut down on the amount of false data that could be read.

VI. PROTOYPE SYSTEM

We devised an algorithm that enables a robot to check on another robot if the *"I am alive"* signal has not been received. The experiment is conducted under the premise that if the X80 robot has not received the signal, it then moves towards the X80-H in order to investigate.

The flow diagram in Figure 2 shows the steps taken if the signal is not received. The robot that has noticed the signal is absent moves to the last known co-ordinates of the robot that is not transmitting. When encountering an object <0.3m it processes a frame of its video feed and looks for red contours with 4 points, if it finds a contour, it looks for text within the bounding box of those 4 points. The purpose of this is to test whether two robots could still communicate information if they no longer have access to the more usual means of wireless communication due to a hardware fault or atmospheric conditions. We used red contours but other shapes and colors could be used to help recognise the screen of the robot that is not transmitting. Another idea would be to train a Haar classifier and have the robot look for the exact image of the non-transmitting robot in its video feed.

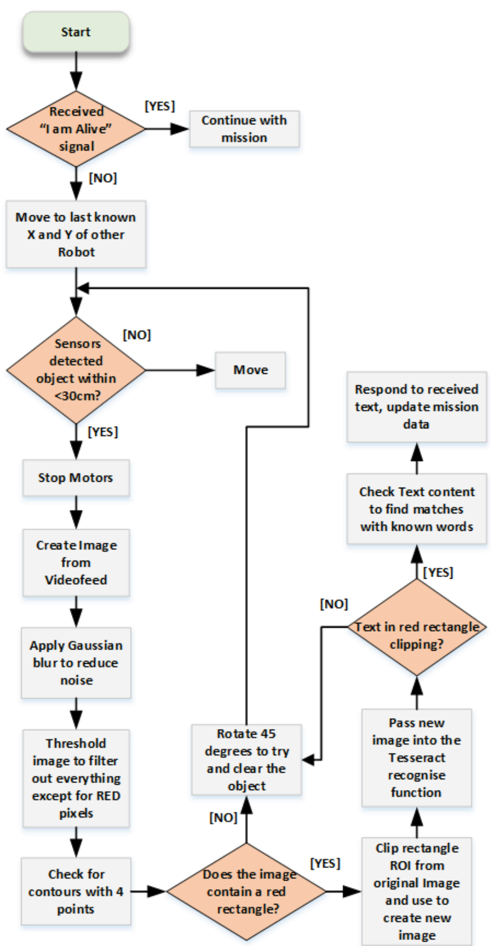

Figure 2. Flow diagram of OCR communication

The Algorithm in Figure 3 shows the steps taken when a robot does not receive the *"I am Alive"* heartbeat signal from another robot within a cluster. The robot moves to the last known co-ordinates and takes a photo, checks the image or text and then follows the command associated with the text.

```
Algorithm: If Heartbeat signal is not received
Robot autonomously moves around the environment
if "I am Alive" signal has not been received then
    Move to last known co-ordinates of other robot
        if sensor detects object within 30cm then
            Stop movement
            Take snapshot of video frame
            Look for red rectangles within the snapshot
        end if
        if red rectangles exist in snapshot then
            if rectangle contains text then
                if text matches a command word then
                    Follow Command e.g., Danger
                end if
            end if
        end if
end if
```

Figure 3. Algorithm used when a heartbeat signal is not received

VII. EXPERIMENTAL RESULTS

The tests involved an X80 and an X80-H robot, with the latter being assigned the role of sender. A Samsung Galaxy Tablet was partially covered in red card and placed on the X80-H, this was used to display the word "DANGER". The X80 was then tasked with approaching the X80-H, stopping when it detected an object <0.3m and using image processing techniques to find a red rectangle and looking for text within it.

In Figure 4, we can see the X80 (right hand robot) has stopped approximately 0.3m from the X80-H (left hand robot in the picture). The X80-H robot has the tablet positioned at the front of its base. The tablet has text displayed; the red rectangle ensures that only the text within it will be processed by the X80

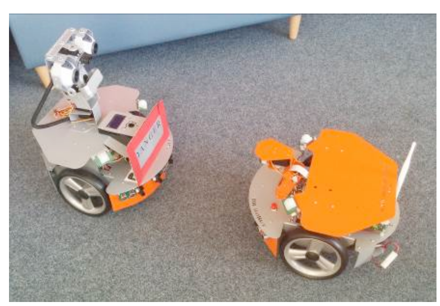

Figure 4. Experiment using an X80-H, X80 and Tablet

In Figure 5, the 1st row from left to right displays the Robot Camera video feed, the 2nd image is of a screenshot taken of the video feed, the blue bounding box highlights a

 ISBN: 978-1-61208-391-9

detected red rectangle. The Red Filter image box is a gray scale representation of the screenshot, a threshold filter has been applied so that only red pixels will display in white, non-red pixels are shown in black. The 2nd row shows the OCR output with the word "DANGER" detected; the Rectangle's image box gives a clearer indication of the size of the rectangle detected. The final image is of the Clipped Rectangle bounding box, it is this image that is used in conjunction with the Tesseract library. Only this image is checked for text.

We can see that only the inner rectangle has been detected, the outer rectangle was occasionally detected; however this made no difference to whether the text was correctly identified. The Red Filter image box shows that the inner rectangle has a more even shape due to the white inner box. It has been noticed that the camera tends to classify non-background pixels as red; this then results in a somewhat fuzzy appearance of the outer box. Due to the irregular shape the outer box has more than 4 contour points and is therefore not detected as a rectangular object.

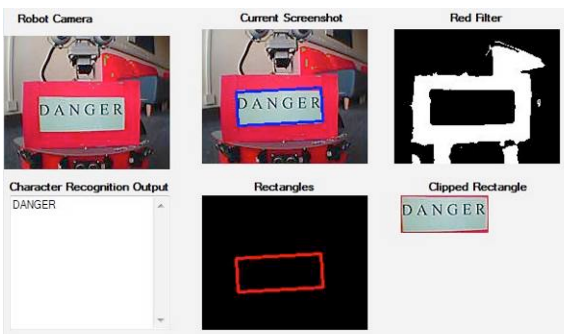

Figure 5. GUI showing processed image and detected rectangle

A. *Experiment 1 Results*

Table I shows the results of 10 tests where the receiver robot was approximately 0.29 metres from the sender robot. The sender robot or X80-H is shown on the left in figure 3. The test was performed under optimum lighting conditions with artificial and natural light. The red rectangle was detected 100% of the time; the OCR output included the word "DANGER" 40% of the time. In 2 cases, the OCR failed to detect any characters, the other 40% included variations on the word, with "7ANGER" being detected 30% of the time.

The X and Y co-ordinates of the inner rectangle are displayed in the table, as are the Width and Height, the readings were very consistent. The distance between the two robots was the same throughout each test, the light in the room varied slightly from one test to the next. The reason for the failed detections as the tests go on seems to lie with the slight variation in lighting. More work is required in order to improve character detection under poor lighting conditions.

TABLE I. RESULTS FROM EXPERIMENT 1

	Detect Rectangle	**OCR Output**	**X**	**Y**	**W**	**H**
1	Yes	DANGER	36	61	92	42
2	Yes	---	36	61	92	42
3	Yes	DANGER	36	61	92	42
4	Yes	DANGER	36	62	92	41
5	Yes	DANGER	36	62	92	41
6	Yes	7ANGER	37	60	91	43
7	Yes	7ANGER	37	60	91	43
8	Yes	7ANGER	37	60	91	43
9	Yes	---	37	61	91	41
10	Yes	7ANOER	37	60	91	43

B. *Experiment 2 Results*

For this experiment, the robot was programmed to read the text within a red rectangle and then respond with an action. If the OCR output contained the word "DANGER" or variations of it, the robot would rotate 360°, results of the experiment are displayed in Table II. The variations observed in Experiment 1 were incorporated in to the code for Text checking in Experiment 2. This test was undertaken to show that the robot could interpret detected text and respond with an action.

TABLE II. RESULTS FROM EXPERIMENT 2

	Detect Rectangle	**OCR Output**	**X80 Action**
1	Yes	%^NGER	**NONE**
2	Yes	DANGER	**ROTATED**
3	Yes	DAWN	**NONE**
4	Yes	---	**NONE**
5	Yes	DANGER	**ROTATED**
6	No	---	**NONE**
7	Yes	DANGER	**ROTATED**
8	Yes	DANGER	**ROTATED**
9	Yes	DANGER	**ROTATED**
10	Yes	DANGER	**ROTATED**

Checking whether a misspelt detected word is similar to a correctly spelled word was explored in [20], where the researchers were able to improve text detection accuracy by using probabilistic inference. Our experiments have shown that more work is required in image pre-processing to correct distortion before an image is passed to the OCR engine. The experiments were conducted at a distance of approximately 0.29 metres; this was due to limitations with the camera being used. In [19][24], the researchers used a pan/tilt/zoom camera to zoom in on a region of interest. In

 ISBN: 978-1-61208-391-9

future we would like to try a similar approach as this would allow for experiments to be conducted at a greater distance. Future experiments will also seek to include more meaningful instructions and responses.

C. Further Experiments

Further experiments were carried out that tested the X80 reading the screen at various angles, the rectangle was repeatedly detected but unfortunately the word could not be recognised unless the robot was facing the screen straight on. The variation observed when lighting conditions are sub optimal can be seen in Figure 6; in this case, too much sunlight has caused red pixels to be detected as green pixels. The Red Filter is unable to detect a rectangle and therefore cannot look for text. However, this is also due to the X80's camera; it has proven to be of low quality and regularly experiences distortion. In future, similar experiments could be performed with a higher resolution camera equipped with a zoom capability.

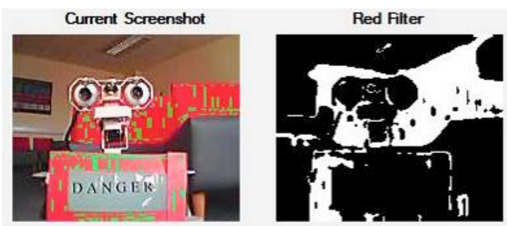

Figure 6. X80 camera distortion, red pixels appear green

Numerous other tests were conducted that involved having a robot read printed words. Using printed text as opposed to an electronic screen greatly improved the rate of detection; this is because the Tesseract OCR has been developed for use with printed text. These tests involved the X80 robot moving around a room and checking for text when it encountered an object < 0.3m away. The text contained various words that could be translated in to simple commands such as Rotate, Move Right, Reverse and Stop. Figure 7 shows the results from a test where the robot was instructed only to process text contained within a red rectangle.

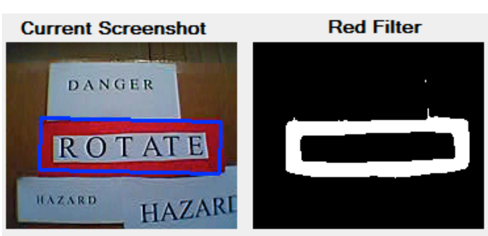

Figure 7. Finding a red rectangle

Figure 8. Camera distortion due to lighting conditions

Despite the higher success rate observed with the printed word test, camera distortion still caused a high failure rate. Figure 8 shows an example of the level of distortion the camera is prone to experiencing. The noise tends to distort the shape of the characters and prevent a word being detected accurately.

VIII. SUMMARY AND FUTURE WORK

In this paper, we proposed a concept that would allow mobile robots to communicate using a form of textual semaphore. By locating an object using Ultrasonic sensors, a robot can analyze whether there is a red rectangle within its field of view. The red rectangle frames a screen used to display textual commands. The contents of a detected rectangle are processed by the responding robot. If the rectangle contains text, then the robot has found the non-transmitting robot. The experiments showed that lighting conditions and the low quality camera greatly affected the results; future experimentation would look at how to compensate for this so that the text can still be read. Improving image noise by using pre-processing techniques will also be explored in future.

Future experimentation may involve instructing one robot to follow another and read the directions on the screen. The robots could be placed within a text-rich environment to test how well the method works. Further work is also needed to see if the text detection rate can be improved and poor lighting conditions compensated for. We would also like to test different screens to ascertain whether some are easier to read at an angle than others. Performing future experiments with a high resolution IP camera instead of the video camera supplied with X80 would help eliminate unwanted image distortion.

The end goal of this research is to develop a system that allows robots to converse via screens with changeable text. Our results to date are encouraging and demonstrate that the principle of communicating in this manner has some potential.

REFERENCES

[1] IBM, "An Architectural Blueprint for Autonomic Computing," IBM, 2003.

[2] R. Sterritt, "Towards Autonomic Computing : Effective Event Management," in Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02), 2003, pp. 40–47.

[3] R. Sterritt and D. Bustard, "Towards an autonomic computing environment," in Proceedings of the 14th International Workshop on

Database and Expert Systems Applications (DEXA'03), 2003, pp. 694–698.

[4] R. Sterritt, M. Parashar, H. Tianfield, and R. Unland, "A Concise Introduction to Autonomic Computing," in Advanced Engineering Informatics, 2005, vol. 19, no. 3, pp. 181–187.

[5] L. Matthies et al., "Computer Vision on Mars," in International Journal of Computer Vision, 2007, vol. 75, no. 1, pp. 67–92.

[6] R. Sterritt, C. Rouff, J. Rash, W. Truszkowski, and M. Hinchey, "Self-* Properties in NASA Missions," in 4th International Workshop on System/Software Architectures (IWSSA'05) in Proceedings of the International Conference on Software Engineering Research and Practice (SERP'05), 2005, pp. 66–72.

[7] R. Sterritt and M. Hinchey, "Engineering Ultimate Self-Protection in Autonomic Agents for Space Exploration Missions," in 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2005, pp. 506–511.

[8] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "NASA's Swarm Missions: The challenge of Building Autonomous Software," IEEE IT Pro, 2004, vol. 6, no. 5, pp. 47–52.

[9] A. G. Ganek and T. A. Corbi, "The Dawning of the Autonomic Computing Era," IBM Systems Journal, 2003, vol. 42, no. 1, pp. 5–18.

[10] B. Jacob, R. Lanyon-hogg, D. K. Nadgir, and A. F. Yassin, A Practical Guide to the IBM Autonomic Computing Toolkit. IBM, 2004.

[11] M. G. Hinchey and R. Sterritt, "Self-Managing Software," IEEE Computer, 2006, vol. 39, no. 2, pp. 107–109.

[12] R. Sterritt and D. F. Bantz, "Personal autonomic computing reflex reactions and self-healing," in IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 2006, vol. 36, no. 3, pp. 304–314.

[13] S. N. Shaikh and N. V Londhe, "OCR Based Mapless Navigation Method of Robot," International Journal of Inventive Engineering and Sciences, 2013, vol. 1, no. 8, pp. 6–12.

[14] C. Case, B. Suresh, A. Coates, and A. Y. Ng, "Autonomous sign reading for semantic mapping," IEEE International Conference on Robotics and Automation, 2011, pp. 3297–3303.

[15] D. Létourneau, F. Michaud, J. Valin, and C. Proulx, "Making a Mobile Robot Read Textual Messages," In Proceedings IEEE Conference on Systems, Man, and Cybernetics, 2003, vol. 5, pp. 4236-4241.

[16] J. G. Rogers, A. J. B. Trevor, C. Nieto-Granda, and H. I. Christensen, "Simultaneous localization and mapping with learned object recognition and semantic data association," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 1264–1270.

[17] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," in Computer Vision and Pattern Recognition, CVPR., 2005, vol. 1, pp. 886–893.

[18] "Google Goggles." [Online]. Available: http://en.wikipedia.org/wiki/Google_Goggles. [Accessed: 24-Jan-2015].

[19] M. L. Wyss and P. Corke, "Active Text Perception for Mobile Robots," in unpublished, 2012.

[20] I. Posner, P. Corke, and P. Newman, "Using text-spotting to query the world," in IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings, 2010, pp. 3181–3186.

[21] X. Liu and J. Samarabandu, "An Edge-based Text Region Extraction Algorithm for Indoor Mobile Robot Navigation," in IEEE International Conference Mechatronics and Automation, 2005, pp. 701–706.

[22] H. F. Chen, C. Y. Chiang, S. J. Yang, and C. C. Ho, "Android-based patrol robot featuring automatic license plate recognition," in Computing, Communications and Applications Conference, 2012, pp. 117–122.

[23] R. Ramos-Garijo, M. Prats, P. J. Sanz, and A. del Pobil, "An autonomous assistant robot for book manipulation in a library," in SMC'03 Conference Proceedings. IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance, 2003, vol. 4, pp. 3912–3917.

[24] N. Ezaki, M. Bulacu, and L. Schomaker, "Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons," in 17th International Conference on Pattern Recognition (ICPR), 2004, vol. 2, pp. 683–686.

[25] "Dr Robot X80." [Online].
Available:http://www.drrobot.com/products_item.asp?itemNumber=X80. [Accessed: 24-January-2015].

[26] "Dr Robot X80-H." [Online].
Available:http://www.drrobot.com/products_item.asp?itemNumber=X80-H. [Accessed: 24-January-2015].

[27] "EMGU CV." [Online].
Available:http://www.emgu.com/wiki/index.php/Main_Page. [Accessed: 24-January-2015].

[28] "OpenCV (Open Source Computer Vision)." [Online].
Available: http://opencv.org/. [Accessed: 24-January-2015].

[29] Google, "Tesseract-OCR." [Online].
Available:https://code.google.com/p/tesseract-ocr/. [Accessed: 24-January-2015].