Development of an Optical Autonomous Satellite Identification System

Lochie Ferrier

under the direction of
Dr. Alvar Saenz-Otero
MIT Space Systems Laboratory

Research Science Institute
July 30, 2014

Abstract

Reliable identification of satellites is one of the major challenges hindering the development of autonomous satellite management systems. Electronic identification is unsuitable for this task as it requires electrical power and is relatively difficult to standardize. 3D modelling techniques are also unsuited to this task as the satellite could be damaged over its lifetime. This paper presents an identification system utilising a pre-placed optical marker, which is unpowered and resilient. The marker could be identified with lightweight image-processing systems on an inspecting satellite, making it a promising solution for the problem of satellite identification.

Summary

Large amounts of space debris in orbit today are making space an unsafe environment for satellites and astronauts. Robots that can manage space debris have been proposed as a solution to this problem. Currently, no system has been developed that can reliably identify space debris or inactive satellites, making it impossible for a recycling satellite to classify and reuse space debris independently. The prototype identification system described in this paper consists of a small marker, similar to a sticker, that could be placed on a spacecraft prior to launch. Once the spacecraft fails and becomes debris the marker could then be scanned and used to identify the piece of debris, similar to how grocery scanning in a store. Although the system developed was a prototype, it showed that the concept is promising as a future identification method.

1 Introduction

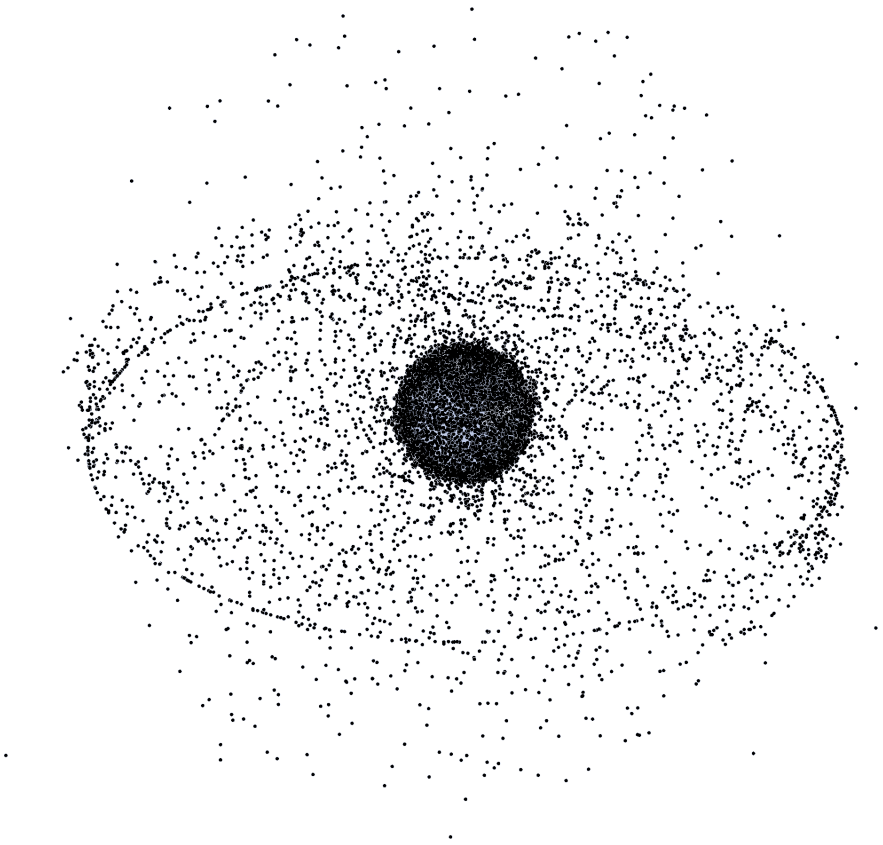


Figure 1: Diagram showing the positions of approximately 19,000 pieces of space debris greater than 10cm in diameter as of July 2009. The outer ring is debris in geostationary high earth orbit and the inner, denser cloud is comprised of debris in low earth orbit [1].

Today, the multitude of active and inactive satellite hardware in geocentric orbit creates a cloud of man-made machinery around the Earth [1]. It is important that this cloud of satellites remains regulated, as many people rely upon it for essential functions such as communication and observation [2]. Several databases are maintained to provide information about the satellites currently orbiting Earth; however, no system currently exists that can use these databases to identify a satellite solely by observation [3, 4]. Electronic systems can identify most satellites, but these systems cease to function when satellites inevitably fail and lose power. These inactive satellites remain in orbit and pose a great danger to

spaceflight operations in geocentric orbit. As most satellites in orbit are inactive, using electronic methods as a solution is not optimal [1].

A variety of solutions are being developed to combat the problem of space debris, including robots equipped with harpoons and vast nets, and ground-based laser removal systems [5, 6, 7]. Many future satellite management programs, such as DARPA's Phoenix satellite recycling program, require the ability to identify satellites reliably [8]. If a satellite cannot be identified, it is difficult to autonomously determine its basic properties and design, making reuse problematic as parts can't be categorised or managed. Generally, interaction with satellites using an autonomous system is almost impossible if the target satellite's properties are not known. Without a simple identification system that can function when a target satellite is inactive, complex and inefficent methods such as identification by comparison with 3D models or by human intervention must be employed to characterise the target.

In the future, in accordance with the Kessler Syndrome, many satellites may be seperated by collisions into smaller, separate parts during their lifetime [9]. The Kessler Syndrome, proposed by the NASA scientist Donald J. Kessler, hypothesizes that the problem of space debris may cascade out of control in the future. As more space debris is created, the probability of collision increases, creating even larger amounts of debris. Already, many satellites have degraded into smaller parts [1]. As a result, both whole satellites or spacecraft (e.g. the Hubble Space Telescope) and satellite sub-systems or parts which may become disconnected in the future (e.g. the Faint Object Spectrograph onboard Hubble Space Telescope) will be referred to as *satellites* in this paper.

An optical identification system could be a simple, low weight solution to this problem. The identification system proposed in this paper, entitled OASIS (Optical Autonomous Satellite Identification System), consists of a visual marker that is placed on the satellite and a computer vision system that runs on an inspecting satellite or other system.

OASIS uses much of the technology developed by MIT's Space Systems Laboratory (SSL). The SPHERES VERTIGO satellite vision prototyping system, developed by SSL, was used as a camera system for OASIS. The original SPHERES system was designed for the development of satellite movement algorithms in a zero gravity or flat floor environment [10, 11]. SPHERES has been used in the past to develop many systems, such as a zero-gravity fuel-slosh control experiment [12, 13].The VERTIGO system, comprised of a stereo camera system and a small image processing computer, added vision capabilities to SPHERES. The open-source library OpenCV and the scripting language Python were used to develop an image processing algorithm to locate and scan the marker. Although an algorithm was not fully developed, advances were made in filtering image noise.

The marker was designed to be visible in a variety of lighting conditions and orientations. Several features were incorporated into the design of the marker for this reason. Retroreflective tape was used to ensure that the marker would be visible to a computer vision system in low light conditions. To make the marker useful to humans (i.e. astronauts), human readable information was incorporated alongside computer vision markers.

2 Visual Marker

At first, design choices were poor, creating designs that did not fulfill the requirements outlined in Section 2.1. As the design was refined, the focus shifted towards functionality, resulting in a design that was clearly visible and useful for the application. The primary requirement for the visual marker was that it should be visible to a computer vision system at a reasonable distance in a variety of lighting conditions. To test this requirement during the design process, each prototype was photographed using the VERTIGO camera system, and then evaluated for its readability. Testing of the prototypes was performed in a similar manner to the testing of the full system as outlined in Appendix A.

Two basic technologies were used in all versions of the visual marker. A QR code was included in all designs, for autonomous identification. However, without a guidance system, a QR code would be difficult to locate at a long distance. For this reason, a retroreflective guidance system comprised of small, square pieces of retroreflective material was included on the marker as shown in Figure 2. As the positions and dimensions of these squares were known, they could be used to locate the marker at long distances.

2.1 Requirements

The visual marker component of the system was designed to meet the following requirements, ordered by priority. The assumptions made here are that the marker will be imaged by the VERTIGO camera system and that the marker will use a version of a QR code.

1. The QR Code should be readable by the VERTIGO camera system at a distance of 30cm in normal and low light conditions.

2. The marker should feature a guidance system to allow an inspecting system to locate the marker in normal and low light conditions.

3. The marker should present information that is readable by a human (i.e. text or graphics) at a distance of 1m.

4. The marker should be no larger than a 10 x 10 cm square, and no thicker than 5mm.

2.2 Marker Design

The development of marker versions 1 and 2 is detailed in Appendix A. Here, the features of the final version of the marker shall be discussed. Previous versions had no formal specifications regarding the size of discrete elements relative to the size of the whole marker, making them difficult to reproduce or standardize. The final marker (version 3) was made to

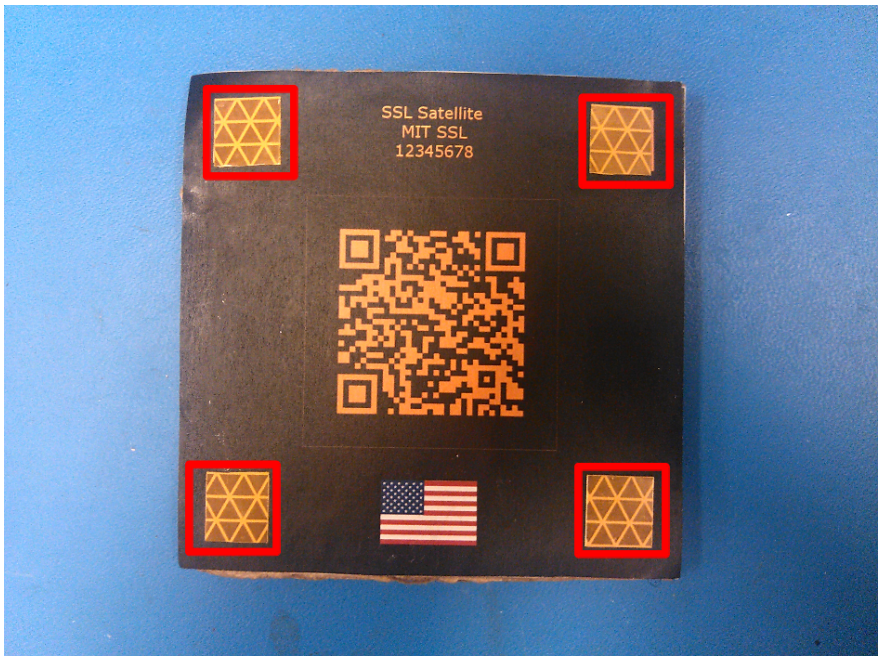


Figure 2: A final version marker with guidance squares highlighted in red.

be more universal, fixing many of the problems found in previous versions. The final design used a ratio based sizing system to allow for scaling, or reproduction of the entire marker without sizing or formatting issues. This ratio-based sizing system also makes it easier for an image-processing algorithm to find the marker as it can use dimensions to characterize the marker. This is shown in the dimensions of the template in Figure 3.

The square shape of the marker allowed it to be easily mounted on a variety of parts. The QR code used here was larger than those of the previous versions and was highly readable as it used a v4 QR code with a medium to low level of error correction. The QR code had a reasonable capacity of 90 characters and was readable by the VERTIGO camera system at the required distance of 30cm as shown in test images 19L-24R. The flag portion was improved to accomodate the two standard aspect ratios of national flags (1:2 and 2:3). The color was also restored to the flag to enable better identification of nationalities.

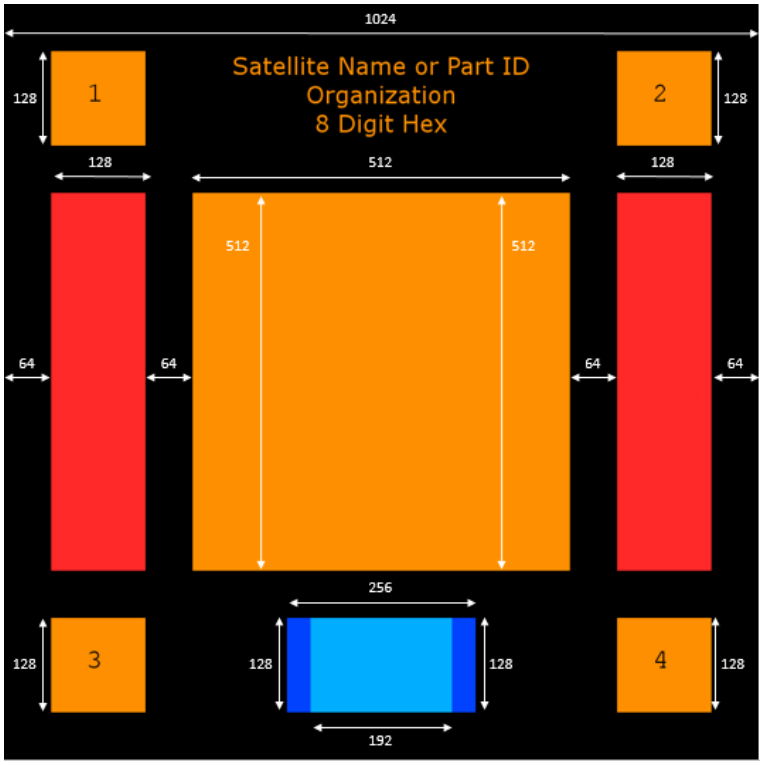


Figure 3: The template for the final marker design with dimensions marked in pixels. QR code is located in the central orange region. The four orange squares are the guidance squares. The blue area at the base of the marker is the space for either a 1:2 or 2:3 flag. The red bars on both sides of the QR code are free space for future expansion. Marker is 10cm in width and height and is printed at scale.

3 Computer Vision

The computer vision system developed for OASIS consisted of two components: a camera system and an image-processing algorithm. The camera system was comprised of a subset of the complete VERTIGO system. To use the images captured by the imaging hardware for locating and identifying the marker, an image-processing algorithm was developed using OpenCV. However, a solution for reliably locating the marker in an image was only partially developed.

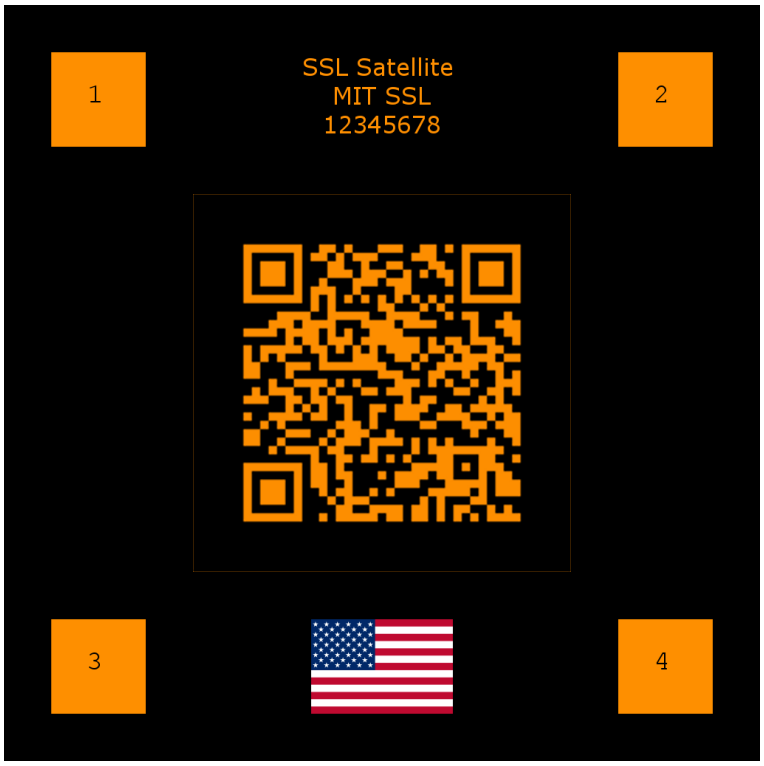


Figure 4: An example of the final marker design. Marker is 10cm in width and height and is printed at scale.

3.1 Imaging Hardware

The VERTIGO camera system was used to capture images of the marker at a reasonable resolution. The complete VERTIGO camera system consisted of many components, which are shown in Figure 5. However, for the purposes of OASIS, a subset of these components was used as shown in Figure 6, to reduce processing time and simplify development. Instead of using the VERTIGO image-processing machine, which had poor performance, a laptop computer was used to gather and process images. Though the number of components was reduced, the relative placements of the components was same as in the complete VERTIGO system. Two 480 x 752 pixel monochrome USB Interface IDS UI-1220LE cameras were arranged in a stereo configuration for imaging. Two 700mA red LEDs were placed in close proximity to the cameras to maximize reflection from the retroreflective markers.

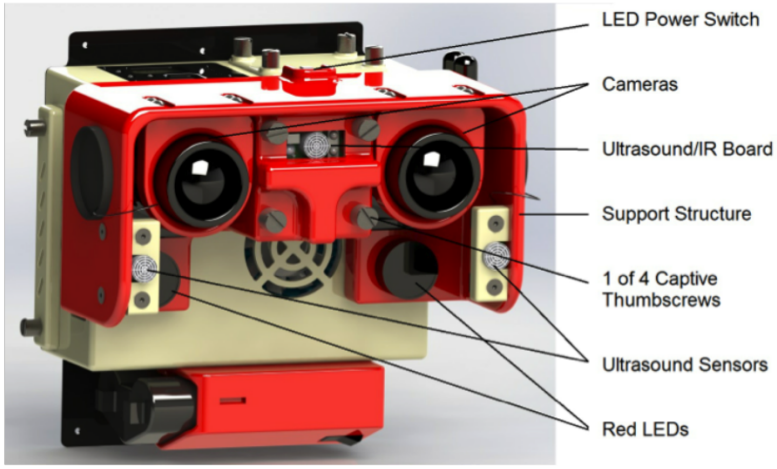


Figure 5: CAD drawing of the complete VERTIGO system with components labelled. Image credit: Tweddle et al. [14].

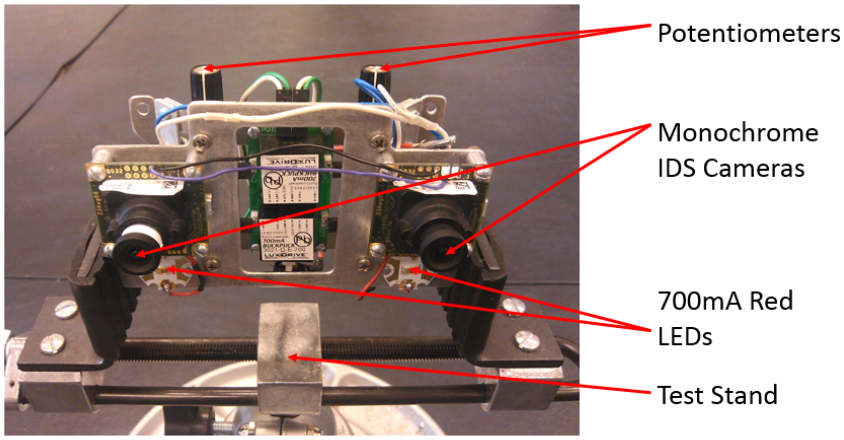


Figure 6: The subset of VERTIGO components used for OASIS. The components were placed on a piece of aluminium which was then mounted in a test stand.

3.2 Algorithm

Several issues prevented the rapid development of an object-detection algorithm that could reliably locate the marker. Extreme lighting conditions in space can make the marker difficult to distinguish from the backing material and glares from reflective coatings can obscure the marker as demonstrated in Figure 7. This made development of a complete algorithm

relatively difficult, as the marker was not always locatable even if it was within the field of view of the camera. In addition, even if the marker was clearly visible, noise could appear in the background which was almost indistinguishable from the marker. Noise sources could include the target satellite, Earth or the sensor in low-light conditions.

Figure 7: Test image 23R, with glare obscuring the upper right guidance square.

To locate the guidance squares within an image, the image-procesing concept of contours was used. Contours within an image are defined as boundaries around areas of similar colour or intensity. In this case, as monochrome images were taken by the camera system, contours were found by intensity alone. The signal desired for this filter was the complete contours of at least three of the retroreflective guidance squares placed around the edge of the marker. The noise was considered to be any contour that was not a contour of the guidance squares. The target performance for this filter was an average noise reduction rate of 50% and a complete signal retention or success rate of 85%. By using edge detection or image segmentation to find the contours within an image, useful information about the shapes within an image could be determined. Once the contours had been found in an image, they could be characterized or evaluated on an individual or group basis to reduce the amount of noise in the image. For

OASIS, all the contours in the image were found with the Suzuki algorithm developed by Satoshi Suzuki and Keiichi Abe [15]. Examples of contours found in images using the Suzuki algorithm are shown in Figures 8 and 9.

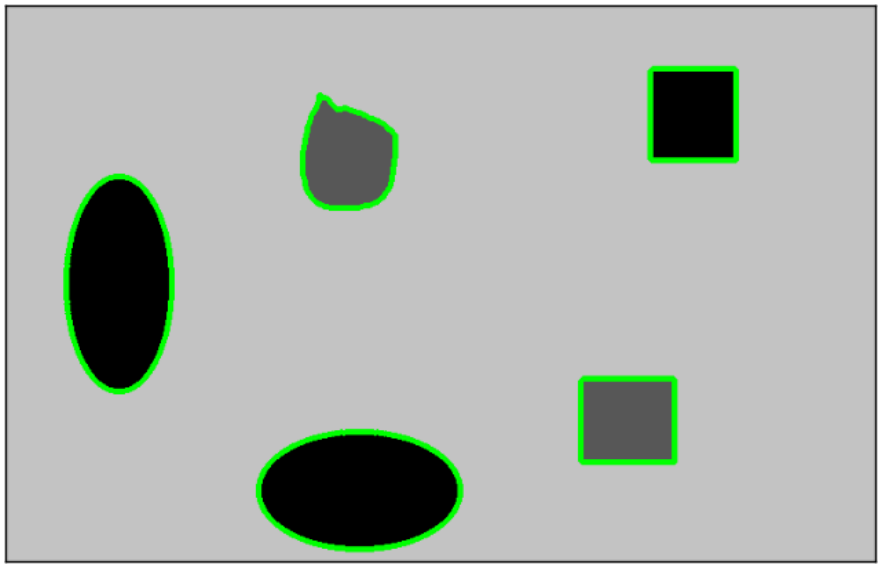


Figure 8: In this image with 5 contours, shapes can be easily identified.

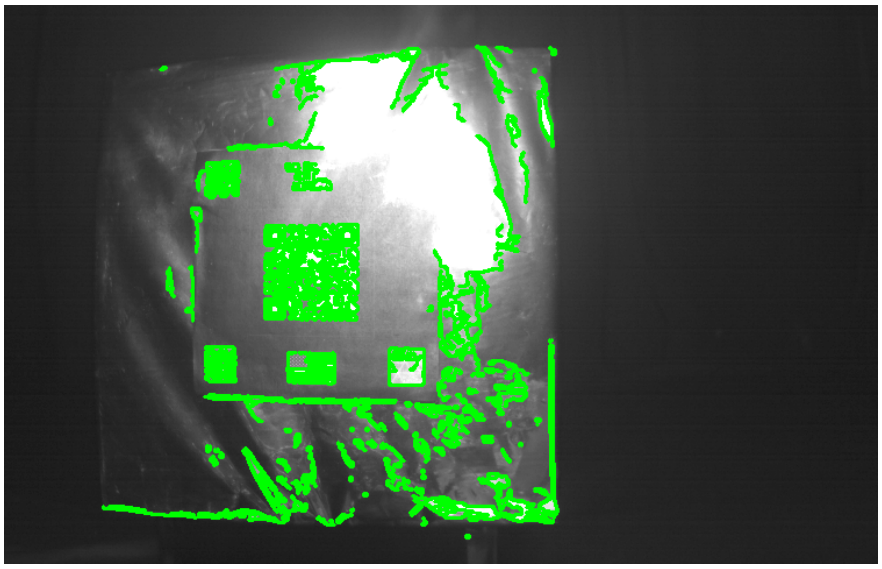


Figure 9: Test image 23L, with 435 contours found. Simple shapes have become difficult to distinguish as a result of the high amount of noise.

The filter operated in a sequential process, discarding noise at each stage. The filter was carefully designed so that, in a wide variety of conditions, an appropriate amount of noise would be discarded at each stage without removing significant amounts of signal.

The first stage of the algorithm found contours in the image using a Canny Edge Detector and subsequently the Suzuki algorithm [16].

A simple individual square-classification method was applied to discard noise from the initial set of contours. An s-value, invented by the author, was calculated for each contour from the product of the contour's aspect ratio (rotation invariant) and its approximated area. The contours were then sorted by s-value using Python's built-in implementation of the Timsort hybrid sorting algorithm [17]. In this sorted set, the signal (the guidance squares) was usually ranked within the top 30% of s-values. The lower 70% of contours were discarded as a result. Of course, for some images, the total number of contours in the set was too low to simply extract the upper 30% of all contours as less than three contours would have been extracted, resulting in a loss of signal. This case was accounted for, and in sets that had fewer than 31 initial contours, the 9 contours with the greatest s-values were extracted.

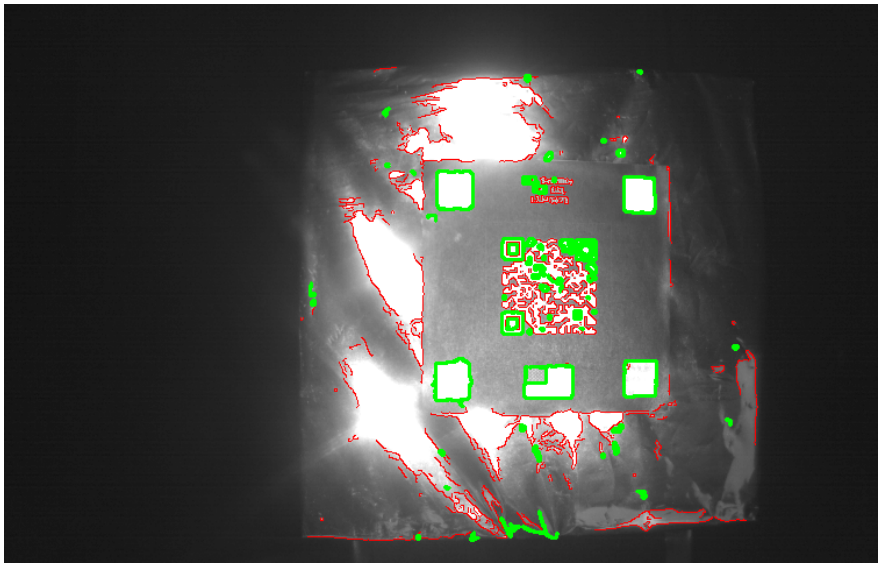


Figure 10: S-value filtering stage applied to test image 24L. Contours that were discarded are drawn in red and contours that were kept are drawn in bolded green.

To further reduce noise and maintain efficiency, it was necessary to consider the signal as groups of three contours, the minimal number of contours required to locate the marker. Groups of four contours were not considered because in many cases, one contour had not been found or had been discarded by the s-value filtering stage. All possible combinations of three contours were iterated through, the most computationally expensive step of the algorithm. The centroids of each contour in the set were found using image moments. The angle between the centroids was calculated and if the difference between this angle and a right angle was greater than 0.2π, then the set of three contours was discarded. This stage normally removed a small amount of extreme noise in sets with a low amount of contours. Each undiscarded three contour set was added to an array.

For each three contour set within the array, four characteristics were calculated and stored in arrays in the same order. The distances between contours, the illuminations of contours, the s-values of contours and the similarity of contours as measured with a shape matching function were used as characteristics.

The distance ratios and angles between contours in the signal three contour set were known as illustrated in Figure 11. This could be used as a characteristic for the three contour set. Firstly, the ratio $\frac{D_1}{D_2}$ was calculated from the centroids of each contour. A skew effect did not change this ratio sufficiently to discard the signal using this characteristic. A second value was required to make the characteristic more discriminative. As it was not computationally efficient to find the side lengths of each contour, the radius of the enclosing circle of each contour was calculated instead. These measurements were then combined to produce a characteristic d as shown in Equation 1.

$$d = |\frac{\sqrt{2}}{12} - \frac{D_1 r}{(D_2)^2}| \tag{1}$$

The intensity of each retroreflective marker could be used as another metric as it was

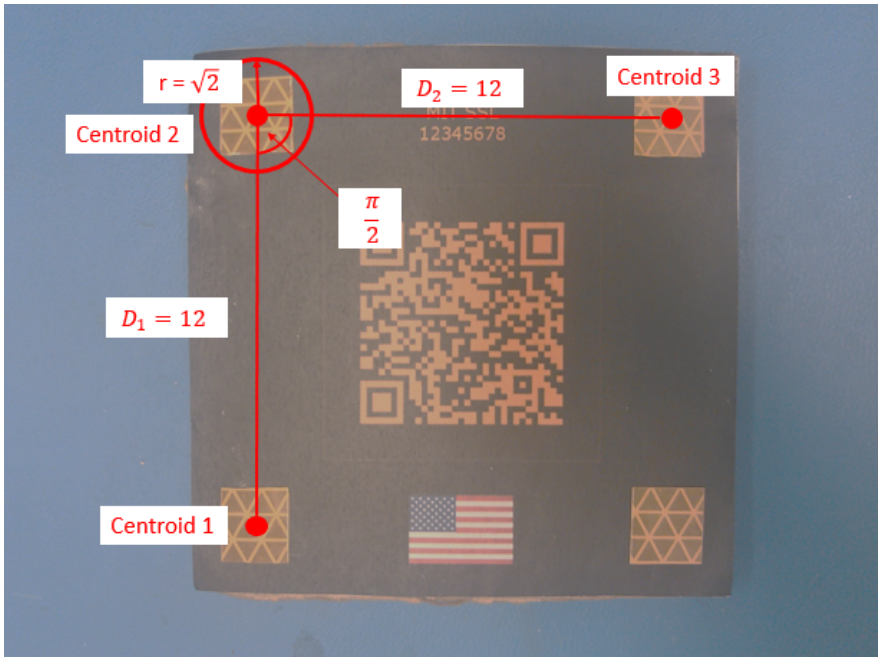


Figure 11: Marker annotated with the dimensions used by final filtering stage.

similar in most situations. The mean intensity of the image mask of each contour in the set was calculated. Then the standard deviation of these mean intensities was used as a characteristic to determine how similar the set was in intensity.

The s-value calculation was reused to characterize the set. The standard deviations of the s-values of the three contours were calculated and used as a characteristic.

A shape-matching function built into OpenCV was also used as a characteristic. All combinations of two shapes were compared using this shape-matching function, and then the standard deviation of the resultant three values was calculated.

When combined, these four characteristics could describe the three square markers, largely irrespective of lighting conditions. A simple way to combine these characteristics would be to multiply them and find the contour set with the minimal product. However, this approach propagated noise for smaller values. A noise set often had one or two very strong characteristics (perhaps caused by the inclusion of one or two squares), implying that a product could be far smaller than that of a signal set. Therefore, a different approach had

to be used that gave preference to good performance in many characteristics.

For each three contour set, the ranks of the set in each characteristic was calculated. These ranks were then summed to create a final metric. The rank of the set according to this final metric was then calculated. The sets ranked in the top 50% according to this metric were retained and the sets in the lower 50% were discarded.

3.3 Testing of Filtering Algorithm

(The full details of the testing process are given in Appendix B.) A successful result was defined as a set of final contours which included three or more of the marker's retroreflective guidance squares. Three different sets of images were considered to evaluate the effectiveness of the filtering algorithm. The value s_i refers to the initial number of guidance squares found by the Suzuki algorithm.

Image Set	Number of Images in Set	Success Rate	Average Contour Reduction Rate
$s_i \geq 3$	37	78.38%	71.58%
$s_i = 3$	12	50.00%	69.42%
$s_i = 4$	25	92.00%	75.62%

For $s_i = 3$, the success rate of the algorithm was far lower than the target value of 85%. However, this could be improved by taking multiple images and performing comparisons to detect more signal. The average contour reduction rate compared to the target rate of 50% was excellent, particularly for image set $s_i = 4$.

4 Conclusions and Suggestions for Future Work

The optical autonomous satellite identification system outlined in this paper demonstrated the potential of an optical approach as a future method for identifying unknown objects in space. The markers developed functioned in a variety of lighting conditions and the foundations of an image processing algorithm were proved. However, many improvements could be made to the image-processing component of the system. Further work could accurately determine the position of the marker by using more modern classification techniques such as machine learning. The marker could also be improved by implementing shields and other features to minimize glare effects.

5 Acknowledgements

I would like to thank Dr. Alvar Saenz-Otero who mentored me in MIT's Space Systems Laboratory. I am especially thankful for the freedom he gave me to pursue this project. I am very thankful for my tutor Dr. Jenny Sendova, for her terrific advice on academic writing and other aspects of research. Bennett Amodio and Katerina Velcheva assisted with many technical issues regarding LaTex. Ava Chen was a fantastic editor refining much of the language in this paper. Georgia Murray, Jason Cui and Marie Herring also assisted with proofreading and editing. For sponsorship, my sincere thanks go to Fyshh iPhone Development, Rotary Club of Canberra South and Canberra Grammar School. Without the fantastic help of MIT and CEE, RSI would not have been possible. The Australian science organization NYSF also provided an enormous amount of logistical support. Special thanks go to Dr. Andrew Shelley, the science teacher who initially encouraged me to apply to this program.

References

[1] NASA. NASA IOTD Space Debris. Available at http://earthobservatory.nasa.gov/IOTD/view.php?id=40173. Accessed: 2014-07-07.

[2] U. of Concerned Scientists. What Are Satellites Used For? Available at http://www.ucsusa.org/nuclear_weapons_and_global_security/solutions/space-weapons/what-are-satellites-used-for.html. Accessed: 2014-07-16.

[3] U. of Concerned Scientists. Satellite Database. Available for download at http://www.ucsusa.org/nuclear_weapons_and_global_security/solutions/space-weapons/ucs-satellite-database.html. Accessed: 2014-07-11.

[4] C. Peat. Satellite Database. Available at http://www.heavens-above.com/Satellites.aspx. Accessed: 2014-07-11.

[5] ESA. Whale of a Target: Harpooning Space Debris. Available at http://www.esa.int/Our_Activities/Space_Engineering/Clean_Space/Whale_of_a_target_harpooning_space_debris. Accessed: 2014-07-28.

[6] ESA. How to Catch a Satellite. Available at http://www.esa.int/Our_Activities/Space_Engineering/Clean_Space/How_to_catch_a_satellite. Accessed: 2014-07-28.

[7] M. T. Review/NASA. NASA Studies Laser for Removing Space Junk. Available at http://www.technologyreview.com/view/423302/nasa-studies-laser-for-removing-space-junk/. Accessed: 2014-07-28.

[8] DARPA. Phoenix - Developing Technologies for More Flexible, Cost-Effective Satellite Operations in GEO. Available at http://www.darpa.mil/Our_Work/TTO/Programs/Phoenix.aspx. Accessed: 2014-07-12.

[9] D. Kessler. The Kessler Syndrome. Available at http://webpages.charter.net/dkessler/files/KesSym.html. Accessed: 2014-07-09.

[10] A. S. Otero. The SPHERES Satellite Formation Flight Testbed. Available at http://ssl.mit.edu/publications/theses/SM-2000-Saenz-OteroAlvar.pdf. Accessed: 2014-07-10.

[11] NASA. ISS Utilization: SPHERES. Available at https://directory.eoportal.org/web/eoportal/satellite-missions/i/iss-spheres. Accessed: 2014-07-10.

[12] S. Chintalapati, C. Holicker, R. Schulman, B. Wise, G. Lapilli, H. Gutierrez, and D. Kirk. Update on SPHERES Slosh for Acquisition of Liquid Slosh Data aboard the ISS. Available at http://arc.aiaa.org/doi/pdf/10.2514/6.2013-3903. Accessed: 2014-07-28.

[13] D. Hayhurst and A. Saenz-Otero. Development and Implementation of the SPHERES-Slosh Experiment. Available at ssl.mit.edu/files/website/theses/SM-2014-HayhurstDustin.pdf. Accessed: 2014-07-25.

[14] B. Tweddle, E. Muggler, A. Saenz-Otero, and D. Miller. The SPHERES VERTIGO goggles: Vision based mapping and localization onboard the international space station. Available at http://ssl.mit.edu/spheres/library/tweddle-isairas12.pdf. Accessed: 2014-07-10.

[15] S. Suzuki and K. Abe. Topological Structural Analysis of Digitized Binary Images by Border Following. Available at http://www.sciencedirect.com/science/article/pii/0734189X85900167. Accessed: 2014-07-20.

[16] OpenCV. Canny Edge Detector. Available at http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html. Accessed: 2014-07-25.

[17] D. R. MacIver. Understanding timsort, Part 1: Adaptive Mergesort. Available at http://www.drmaciver.com/2010/01/understanding-timsort-1adaptive-mergesort/. Accessed: 2014-07-24.

[18] DENSO ADC. Information capacity and versions of the QR Code. Available at http://www.qrcode.com/en/about/version.html. Accessed: 2014-07-18.

Appendix A Development of Marker

Version 1

Figure 12: An example of a version 1 marker. Width: 5cm, height: 3cm. Printed at scale.

The first version of the marker was designed for the purposes of demonstrating the technology and understanding how the basic elements of the design would work together. There were several aspects of the design which conflicted with the requirements. The dimensions of the v2 QR code were far too small to be read by the VERTIGO system computer at a short distance of approximately 30cm. Also, the use of v2 QR code with a medium to low level of error correction meant that only 38 characters could be encoded, a relatively low amount [18]. The placement of the QR code in the corner of the marker was found to be easily obscured, causing problems with locating the marker. A disproportionate amount of space in this design was used to present text, a design choice that did not reflect the requirements. There were two main issues with the guidance system on this design. The first was that only 3 retroreflective squares were included in the design. If a single retroreflective square was not detected, the guidance system would not be usable by the image-processing algorithm. Since the VERTIGO camera system is monochrome, the use of colours in the retroreflectors was unjustified and perhaps even counter-productive, as it would be more difficult for an algorithm to categorize the markers as a set based on intensity.

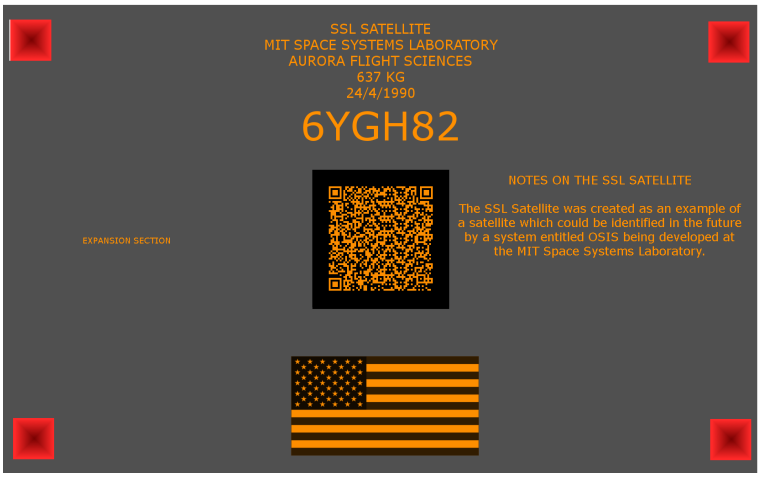


Figure 13: An example of a version 2 marker. Width: 10cm, height: 6cm. Printed at scale.

Version 2

The second version of the marker design attempted to fix many of the issues found in the first version whilst introducing new features. The capacity of the QR Code was expanded to 311 characters by using a v10 QR Code with a medium to low level of error correction. However, this design change meant that the QR Code could no longer be reliably read using the VERTIGO camera system at the required range as the QR code pixels were indistinguishable. The colour of the text and the QR Code was standardized to be orange (hex triplet F8F00) in order to make the marker thermally consistent. The flag was also modified to a monochromatic scale for the same purposes. However, this presented a problem as several flags are only distinguishable by colour, resulting in ambiguity. Similar to the problems with sizing in version 1, this meant that there was little space left for the QR code, although the central placement of the QR code meant that it was less likely to be obstructed. The rectangular dimensions of the marker were not desirable as they meant that dimensions based characterization could not be used by the image processing algorithm.

Appendix B Testing

Methods

To test the performance of the system in a variety of lighting conditions and applications, an experiment was designed to simulate the environmental conditions of space. The following variables were modified:

1. Position or pose of the camera system relative to the marker. Five different positions and poses were tested as marked on Figure 16. The camera system was rotated 60 degrees in some positions to test different perspectives.

2. Ambient lighting conditions. Tests were conducted with and without 100W spotlight aimed at marker.

3. Satellite coating materials. Two 20cm cubes were constructed, one was painted white and one was covered in a reflective thermal material. More details in Figure 14.

4. Illumination of retroreflectives. The LEDs on the camera system were switched on and off to test the effectiveness of the retroreflective guidance system.

Tests were conducted in the order of the results table, obtaining the left and right camera images simultaneously. Tests in which the LEDs were turned off and the ambient lighting was dark were not performed and are not included in the results. Four tests were also performed without the test cubes, to check for background influence on the experiment (31L-34R). 34 tests in total were performed, producing 68 images.

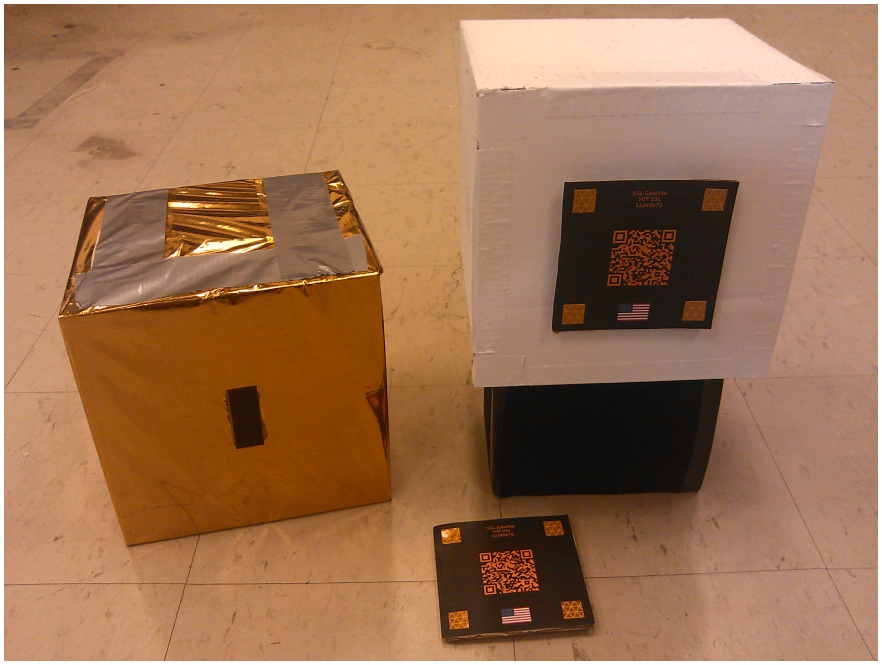


Figure 14: Test cubes and markers. Each marker was attached to a small piece of cardboard and then affixed to a test cube with Velcro®. Note that the non-reflective white cube is resting on a small riser box covered in black fabric. This riser was designed to center the marker in the camera's vertical field of view as shown in Figure 15

.

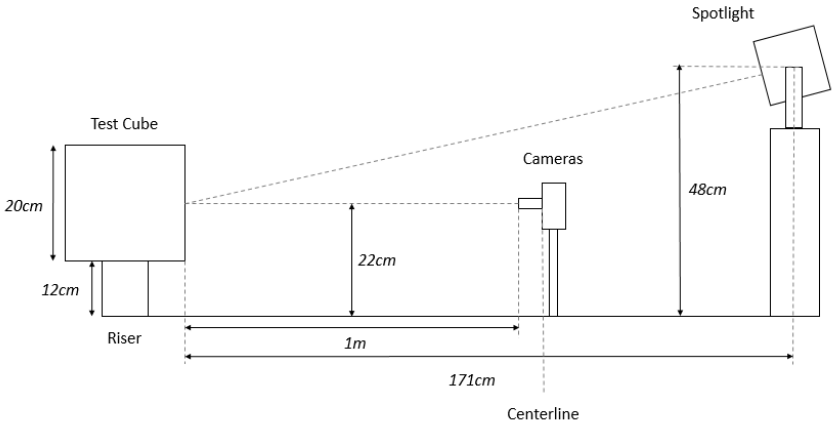


Figure 15: Side view of the testing setup.

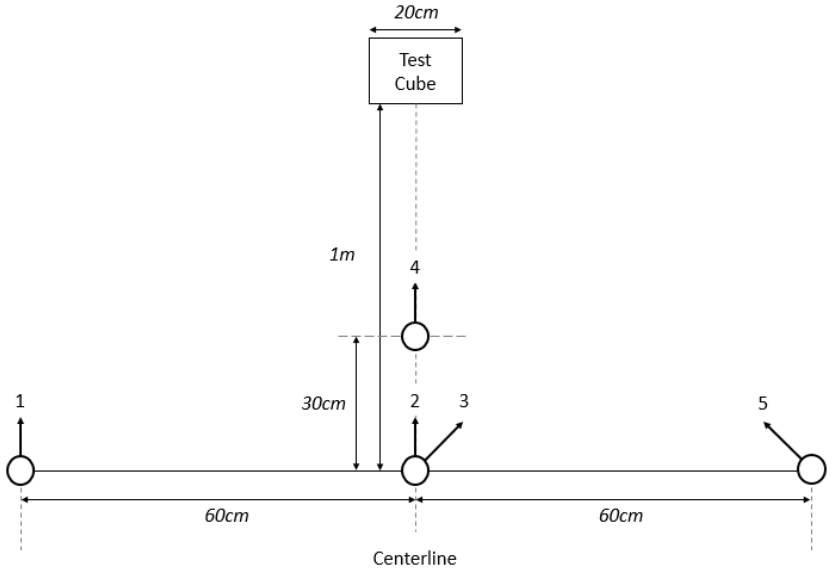


Figure 16: Top view of testing setup. Positions are indicated by circles with arrows for direction.

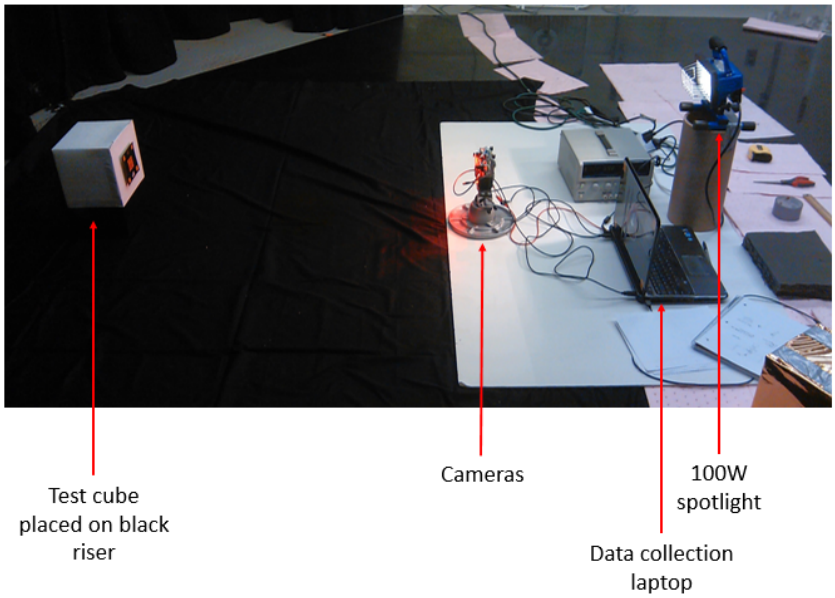


Figure 17: Image of the testing setup with room fluoroescent lights on. Note that these lights were not on during testing.

Results

A readable square was defined as one with a complete contour drawn around its perimeter by the Suzuki algorithm. Contours with extended lines or incomplete curves were ignored. The process of determining the values of s_i and s_f for each image was completed by the author using images with contours drawn at 1 pixel width. Definitions of values in the results table are as follows:

- s_i : Number of guidance square contours found by the Suzuki algorithm.
- s_f : Number of guidance square contours after filtering.
- c_i : Total number of contours (including noise) found by the Suzuki algorithm.
- c_f : Total number of contours (including less noise) after filtering.
- c_{3f} : Total number of three contour sets after filtering.

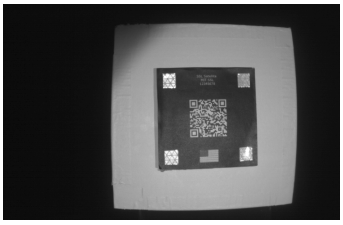
Img No.	Position	Ambient Lighting	Cube Material	LED Status	s_i	s_f	c_i	c_f	c_{3f}
1L	1	Dark	White	LED On	4	3	21	8	534
1R	1	Dark	White	LED On	4	4	38	10	1060
2L	1	Dark	Reflective	LED On	4	4	26	8	540
2R	1	Dark	Reflective	LED On	4	4	48	13	2373
3L	1	Light	White	LED Off	2	1	13	8	534
3R	1	Light	White	LED Off	3	1	25	8	524
4L	1	Light	White	LED On	0	0	17	8	526
4R	1	Light	White	LED On	3	1	38	10	1040
5L	1	Light	Reflective	LED Off	3	1	33	8	530
5R	1	Light	Reflective	LED Off	2	2	43	11	1426
6L	1	Light	Reflective	LED On	4	4	30	8	540
6R	1	Light	Reflective	LED On	4	4	72	20	8672
7L	2	Dark	White	LED On	3	3	24	8	528
7R	2	Dark	White	LED On	4	4	32	8	540
8L	2	Dark	Reflective	LED On	2	2	40	11	1444
8R	2	Dark	Reflective	LED On	3	2	71	20	8726
9L	2	Light	White	LED Off	3	2	28	8	520
9R	2	Light	White	LED Off	3	1	31	8	522
10L	2	Light	White	LED On	0	0	11	8	548
10R	2	Light	White	LED On	0	0	21	8	546
11L	2	Light	Reflective	LED Off	1	1	45	12	1876
11R	2	Light	Reflective	LED Off	1	2	42	11	1425
12L	2	Light	Reflective	LED On	0	0	9	8	361

Img No.	Position	Ambient Lighting	Cube Material	LED Status	s_i	s_f	c_i	c_f	c_{3f}
12R	2	Light	Reflective	LED On	0	0	23	8	541
13L	3	Dark	White	LED On	3	3	21	8	528
13R	3	Dark	White	LED On	2	2	22	8	544
14L	3	Dark	Reflective	LED On	2	2	54	15	3674
14R	3	Dark	Reflective	LED On	1	1	37	10	1092
15L	3	Light	White	LED Off	3	3	19	8	532
15R	3	Light	White	LED Off	1	1	21	8	544
16L	3	Light	White	LED On	1	1	19	8	526
16R	3	Light	White	LED On	0	0	23	8	520
17L	3	Light	Reflective	LED Off	1	1	30	8	532
17R	3	Light	Reflective	LED Off	0	0	31	8	534
18L	3	Light	Reflective	LED On	0	0	12	8	553
18R	3	Light	Reflective	LED On	0	0	19	4	514
19L	4	Dark	White	LED On	4	4	144	42	80728
19R	4	Dark	White	LED On	3	3	145	42	80299
20L	4	Dark	Reflective	LED On	4	3	346	102	1173306
20R	4	Dark	Reflective	LED On	3	3	358	106	1282021
21L	4	Light	White	LED Off	1	1	115	33	38414
21R	4	Light	White	LED Off	1	1	114	33	38222
22L	4	Light	White	LED On	4	1	137	40	69397
22R	4	Light	White	LED On	4	3	140	41	75091
23L	4	Light	Reflective	LED Off	3	3	404	120	1905957
23R	4	Light	Reflective	LED Off	2	1	435	129	2364562
24L	4	Light	Reflective	LED On	4	4	294	87	725971

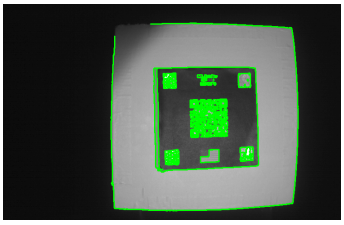
Img No.	Position	Ambient Lighting	Cube Material	LED Status	s_i	s_f	c_i	c_f	c_{3f}
24R	4	Light	Reflective	LED On	4	3	319	94	918098
25L	5	Dark	White	LED On	4	3	31	7	522
25R	5	Dark	White	LED On	4	3	33	8	539
26L	5	Dark	Reflective	LED On	4	4	33	8	527
26R	5	Dark	Reflective	LED On	4	4	42	11	1405
27L	5	Light	White	LED Off	4	3	38	10	1028
27R	5	Light	White	LED Off	4	3	51	14	2926
28L	5	Light	White	LED On	4	1	35	9	783
28R	5	Light	White	LED On	4	3	41	11	1457
29L	5	Light	Reflective	LED Off	4	4	33	8	530
29R	5	Light	Reflective	LED Off	4	4	39	10	1065
30L	5	Light	Reflective	LED On	4	4	43	11	1397
30R	5	Light	Reflective	LED On	4	4	47	13	1848
31L	4	Dark	No Cube	LED Off	0	0	0	0	0
31R	4	Dark	No Cube	LED Off	0	0	0	0	0
32L	4	Dark	No Cube	LED On	0	0	11	0	343
32R	4	Dark	No Cube	LED On	0	0	14	8	540
33L	4	Light	No Cube	LED Off	0	0	9	7	222
33R	4	Light	No Cube	LED Off	0	0	15	8	552
34L	4	Light	No Cube	LED On	0	0	13	8	540
34R	4	Light	No Cube	LED On	0	0	25	8	551

Test Images 19L to 24R

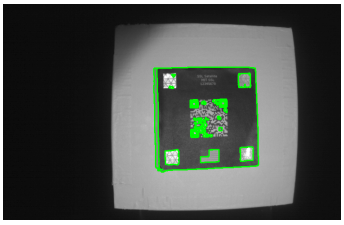
To clearly show various lighting conditions and the performance of the algorithm, the images taken in position 4 were selected. For each test and camera (i.e. 23L and 23R), three images are shown. The first image shows the original image as captured by the camera system. The second image shows the contours found by the Suzuki algorithm. Note that the contours were drawn with a width of 3 pixels for readability. The reader may wish to refer to the results table if they are unsure about the number of square contours in an image. The final image shows the contours remaining after the filtering algorithm has been applied.



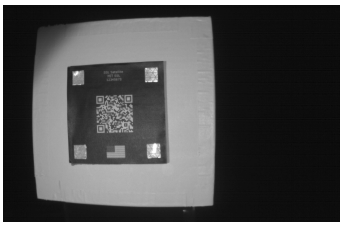
(18.1) 19L original.



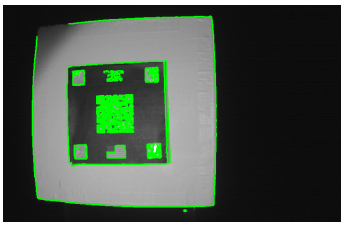
(18.2) 19L initial.



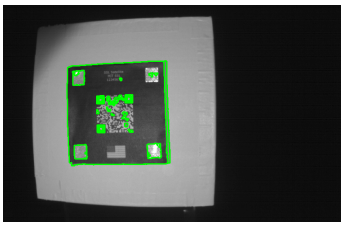
(18.3) 19L filtered.



(18.4) 19R original.

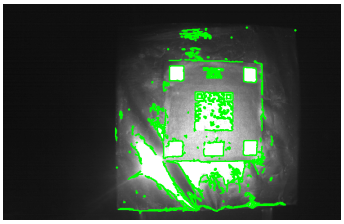


(18.5) 19R initial.

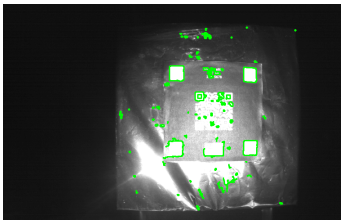


(18.6) 19R filtered.

(18.7) 20L original.

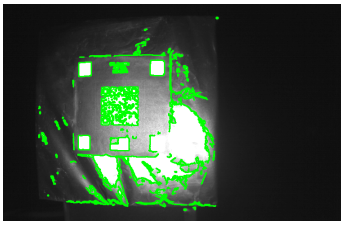


(18.8) 20L initial.

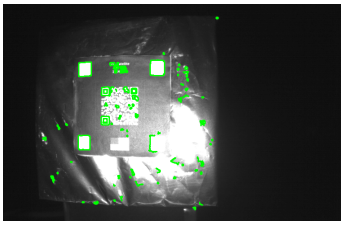


(18.9) 20L filtered.

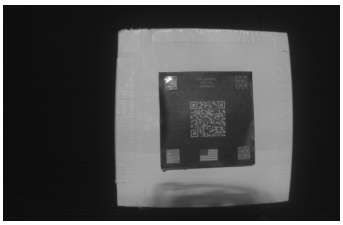
(18.10) 20R original.



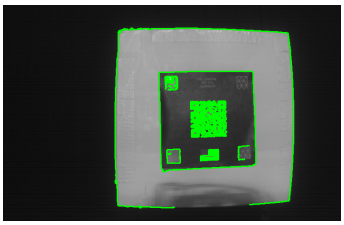
(18.11) 20R initial.



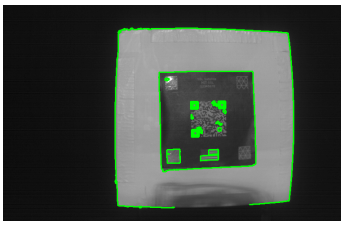
(18.12) 20R filtered.



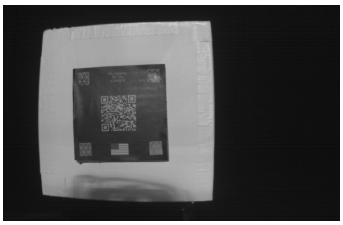
(18.13) 21L original.



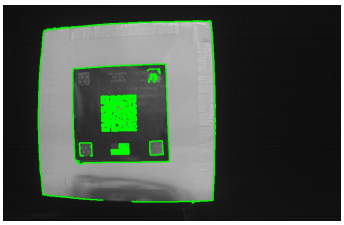
(18.14) 21L initial.



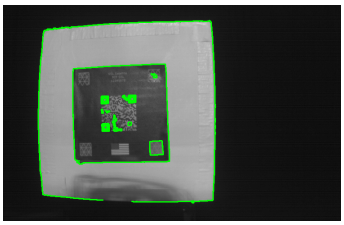
(18.15) 21L filtered.



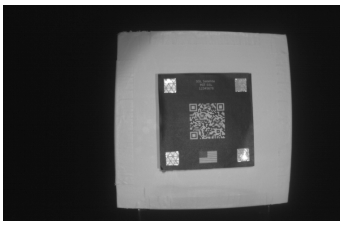
(18.16) 21R original.



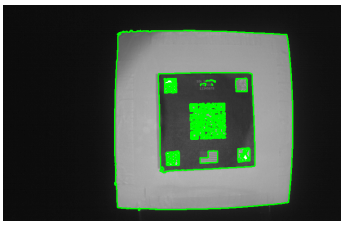
(18.17) 21R initial.



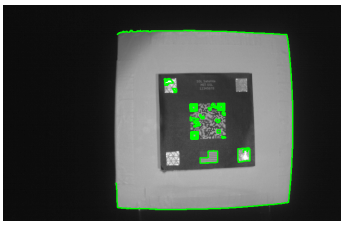
(18.18) 21R filtered.



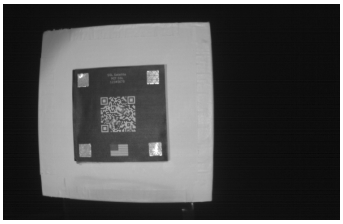
(18.19) 22L original.



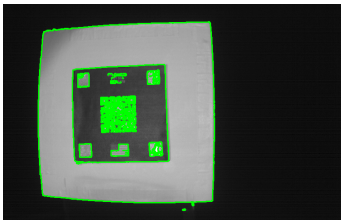
(18.20) 22L initial.



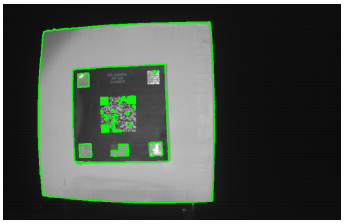
(18.21) 22L filtered.



(18.22) 22R original.

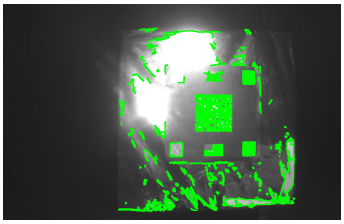


(18.23) 22R initial.

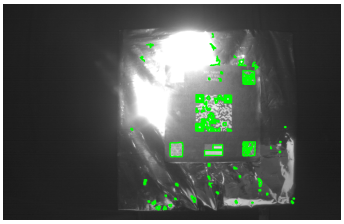


(18.24) 22R filtered.

(18.25) 23L original.

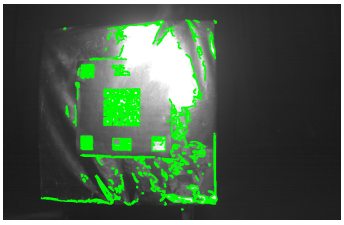


(18.26) 23L initial.

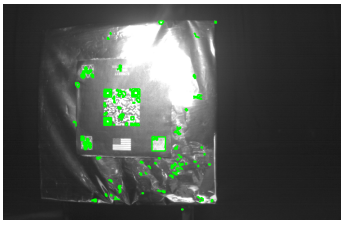


(18.27) 23L filtered.

(18.28) 23R original.

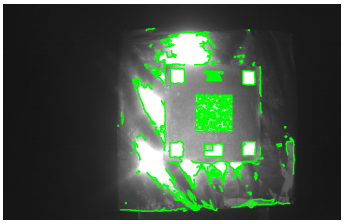


(18.29) 23R initial.

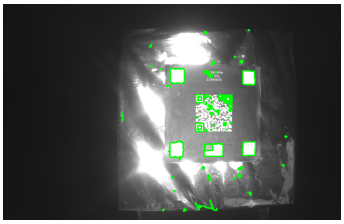


(18.30) 23R filtered.

(18.31) 24L original.

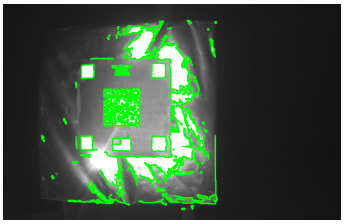


(18.32) 24L initial.

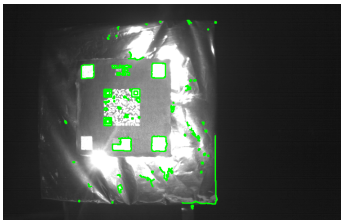


(18.33) 24L filtered.

(18.34) 24R original.



(18.35) 24R initial.



(18.36) 24R filtered.