University of Rhode Island

DigitalCommons@URI

Mechanical, Industrial & Systems Engineering Faculty Publications

Mechanical, Industrial & Systems Engineering

4-1-2018

A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops

Nicholas M. Difilippo
University of Rhode Island

Musa Jouaneh
University of Rhode Island, jouaneh@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/mcise_facpubs

Citation/Publisher Attribution

Difilippo, Nicholas M., and Musa Jouaneh. "A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops." *IEEE Transactions on Automation Science and Engineering* 15, 2 (2018): 887-895. doi: 10.1109/TASE.2017.2679720.

This Article is brought to you by the University of Rhode Island. It has been accepted for inclusion in Mechanical, Industrial & Systems Engineering Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu. For permission to reuse copyrighted content, contact the author directly.

A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops

Keywords

Automated disassembly; computer vision; disassembly tooling; electronic waste (e-waste); robotic disassembly

The University of Rhode Island Faculty have made this article openly available. Please let us know how Open Access to this research benefits you.

This is a pre-publication author manuscript of the final, published article.

Terms of Use

This article is made available under the terms and conditions applicable towards Open Access Policy Articles, as set forth in our Terms of Use.

This article is available at DigitalCommons@URI: https://digitalcommons.uri.edu/mcise_facpubs/428

N. DiFilippo, and M. Jouaneh. "A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops." IEEE Transactions on Automation Science and Engineering, Vol. 15, No. 2, pp. 887-895, 2018.

A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops

Nicholas M. DiFilippo and Musa K. Jouaneh, *Senior Member, IEEE*

***Abstract*— This brief investigates the performance of an automated robotic system, which uses a combination of vision and force sensing to remove screws from the back of laptops. This robotic system uses two webcams, one that is fixed over the robot and the other mounted on the robot, as well as a sensor-equipped (SE) screwdriver. Experimental studies were conducted to test the performance of the SE screwdriver and vision system. The parameters that were varied included the internal brightness settings on the webcams, the method in which the workspace was illuminated, and color of the laptop case. A localized light source and higher brightness setting as the laptop's case became darker produced the best results. In this brief, the SE screwdriver was able to successfully remove 96.5% of the screws.**

***Note to Practitioners*—The amount of discarded electronic waste (e-waste) is increasing rapidly, yet efficient, nondestructive, automated methods to handle the waste have not been developed. Many e-waste products such as laptops use fasteners that need to be removed. In this brief, we focus on removing screws from laptops in a nondestructive manner in order to not damage the laptop, so its parts can be recycled. Due to the vast amounts of laptop models, it is necessary to create a method that will automatically recognize the locations of these fasteners. This brief presents a prototype robotic system that integrates force and vision sensing to automatically locate and remove screws from various models of laptops. The methodology presented in this brief is applicable to other e-waste products with a casing attached by screws. A current limitation of this brief is the robotic system that has to investigate all potential hole locations found by the vision system, although some of these locations may not correspond to valid screw locations. This brief can be extended to include a memory feature that will remember the locations of the screws for cases with similar laptops that are handled by the system to improve the processing time.**

***Index Terms*— Automated disassembly, computer vision, disassembly tooling, electronic waste (e-waste), robotic disassembly.**

I. Introduction

ELECTRONIC waste (e-waste) is a growing concern all over the world. While consumer demand for electronic products continues to grow, the lifecycles of products are shortening, causing a stockpile of discarded e-waste that must be dealt with. E-waste is composed of many heavy metals [1] that are harmful to both human health and the environment. In developing countries, common methods of e-waste disposal are burning, burying, or dumping it in the sea [2]. In 2014, 7.7 million metric tons of e-waste was generated in the U.S., but only 15% of it was recycled. On a global scale, 41.8 million metric tons of e-waste was generated in 2014, and this number is expected to keep growing to 49.8 million metric tons by 2018 [3].

E-waste is recycled by destructive, semi-destructive, and nondestructive methods. Destructive methods are suitable if the aim of the disassembly process is to recycle materials from the part; however, the part will be destroyed. Semi-destructive disassembly will destroy parts of the object such as screws and snap fits. Nondestructive disassembly aims to reverse engineer the assembly cycle. Nondestructive disassembly is appealing if the goal is to reuse parts in a different product or if the disassembled part is hazardous. Manual (nondestructive) disassembly is costly and time consuming since most products are designed for assembly not disassembly.

Robotics is a technology that can revolutionize manufacturing. However, several issues need to be addressed in order to increase the use of robots in manufacturing operations such as the automated disassembly of e-waste. Issues include the high cost of robotic work cells, creation of cost-effective force sensing tooling, and the ability to work in an unstructured environment [4].

The high cost of the primary robotic hardware is only a fraction of the total cost of creating a robotic work cell. Since most industrial robots cannot operate in unstructured environments, large amounts of money are spent designing and fabricating additional equipment, so that parts and features are located at precise prespecified locations. If a robot was able to learn what tasks it needs to perform from sensors in a nonengineered environment, these expenses can be substantially reduced. A robot needs force information to guide interactions with an environment. It would be advantageous to develop tooling that utilizes reliable, low-cost force sensors such as force-sensing resistors (FSRs) [5]–[9] since current sensors such as load cells are expensive.

Several researchers have investigated the design of disassembly tooling and grippers. Rebafka *et al.* [10] proposed a flexible unscrewing tool that could create its own acting surfaces to improve loosening. Park and Kim [11] discussed the development of a six-axis force-moment sensor for an intelligent robot's gripper. Feldmann *et al.* [12] detailed the design of a drill driver, which could use or create a working point to transmit torque to remove a fastener or drill to destroy the fastener. Zuo *et al.* [13] presented a screwnail that could create an indentation in a product and attach to transmit forces and torques required for dismantling operations.

Manuscript received June 23, 2016; revised September 30, 2016 and January 4, 2017; accepted February 28, 2017. This paper was recommended for publication by Associate Editor Y. Li and Editor M. Wang upon evaluation of the reviewers' comments.

The authors are with the Mechanical Engineering Department, University of Rhode Island, Kingston, RI 02881 USA (e-mail: ndifilippo@my.uri.edu; jouaneh@uri.edu).

Seliger *et al.* [14] presented an unscrewing tool consisting of a modified drill bit with movable needles to hold objects down while disassembly steps were performed. This tool was designed for flexibility over accuracy since different variations of products can exist. Peeters *et al.* [15] designed a prying tool that could pull apart housing components to remove liquid crystal display screens.

This brief builds upon previous work performed by Schumacher and Jouaneh [5], [16] where a disassembly tool was created that used FSRs to remove snap fits and spring-loaded batteries from calculators. The tool used two FSRs to detect horizontal forces and one FSR to detect vertical forces on the tool tip. The force applied to the snap fit was monitored with the FSRs in order to fully depress but not damage it. OpenCV's template-matching algorithms were used to identify the type of device and its orientation in the workspace. After a match was determined, the x-y position of the snap fit was retrieved from a lookup table, and robot would begin the disassembly process. Using feedback from the FSRs, the system would first remove the battery cover and then the batteries from the device. The use of a Kinect sensor helped make the system more robust due to minor variations that could occur loading the device. This brief extends the previous work to products that it has no prior information on. This means that if a product placed in a workspace was missing a battery cover or had some other physical defect, the system would still be able to perform disassembly operations.

Disassembly of products is a difficult task because of the uncertainty regarding their physical structure and conditions. Tian *et al.* [17], [18] described a probability analysis method of disassembly cost with a new evaluation parameter, cost disassemblability degree, and chance constrained programming models in order to determine an efficient disassembly sequence. Tang and Zhou [19] presented an extended disassembly petri net to help develop the best disassembly sequence for a product. Kim *et al.* [20] developed a partly automated disassembly system capable of generating and adapting disassembly plans based on the product, as well as determining whether the disassembly step should be performed manually or by an automated process. Guo *et al.* [21] introduced a model using petri nets to determine the disassembly sequence for large scale products which was optimized for disassembly operators and tools.

The motivation for this brief was to create a robotic system that could recycle e-waste and minimize the need for a human operator. This brief presents a novel approach to finding and removing screws that combines vision and force sensing. This approach can later be augmented with a cognitive architecture and using information learned from these methods, that will facilitate the disassembly process.

The rest of this brief is organized as follows. The following section discusses the overall setup of the automated system. Section III goes into further detail about the two main components, the cameras and sensor-equipped (SE) screwdriver. The testing methodology and an overview and explanation of the logic of the system are given in Section IV, while the results of the testing are detailed in Section V. Concluding remarks and a short discussion are given in Section VI.

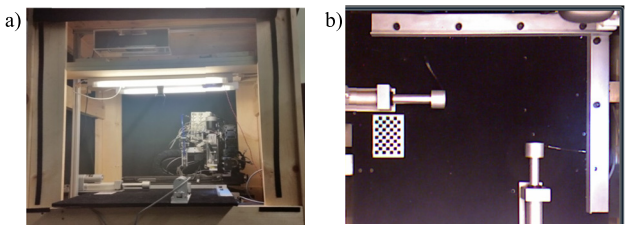

Fig. 1. (a) View of disassembly module with the front fabric panel removed. (b) Top down view of the workspace.

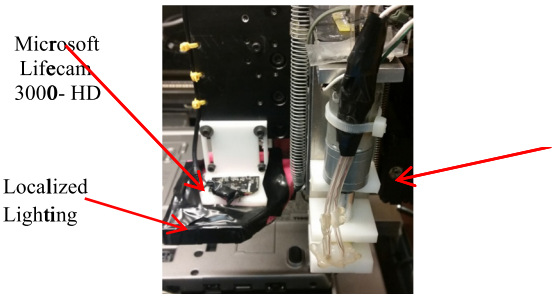

Fig. 2. Disassembly module consisting of the SE screwdriver tool, webcam, and localized lighting.

II. Overall System Setup

The robotic platform used was an ENCore 2s System (MRSI North Billerica, MA) modified to be controlled by a Galil DMC-2143 controller. This controller was interfaced with MATLAB, and a graphical user interface was designed to control the system. A wooden frame to hold panels of black fabric was constructed around the robot and is shown in Fig. 1. The fabric allows for the ambient fluorescent light to be eliminated and a uniform lighting to illuminate the workspace. The black fabric is attached to the frame by Velcro which allows the panels to be removed for maintenance. The workspace platform is constructed out of an aluminum sheet covered with black felt to absorb the IR beam emitted from the Microsoft Kinect sensor. The felt also aids parts in sliding when they are clamped to the edge of the workspace. There is a cutout that allows a checkerboard to be placed level to the surface to set the origin for the camera system. Two linear actuators were added to the workspace to provide automated clamping capability. The SE screwdriver and webcam were mounted on the robot and are shown in Fig. 2. A second webcam was mounted above the robot allowing it to see the entire workspace. In the following section, each of the components will be explained in greater detail.

III. System Components

A. Camera System

The robot uses two different cameras in order to identify laptop features. The first camera is an overhead camera that can view the entire workspace and is used to find circles on the laptop that may contain screws. The second camera is mounted on the robot and used to find and center screw holes. Both cameras are the Microsoft Lifecam 3000-HD

TABLE I
INTRINSIC PROPERTIES FOR MICROSOFT LIFECAM 3000-HD

Symbol	Properties	Values
c_x	Optical center x	654.4 px
c_y	Optical center y	367.7 px
f_x	Focal length x	1129.5 px
f_y	Focal length y	1129.5 px
s	Skew coefficient	2.1673

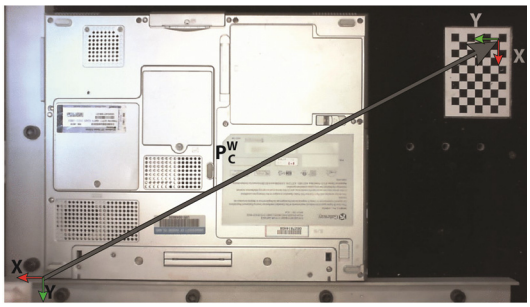

Fig. 3. Coordinate transformation from the camera to world coordinates.

which was chosen because it can capture images in high definition (1280×720 pixels) as opposed to the Kinect sensor which has a lower resolution (640×480 pixels). A webcam calibration process was used to remove lens distortion, which is greatest near the edges of an image, from the overhead camera. The calibration process consisted of using a 7×9 checkerboard pattern (8.5-mm square) and the Camera Calibrator App in MATLAB. The checkerboard is moved around the workspace, and images are taken with it at different angles in order to calculate the camera's intrinsic matrix parameters listed in Table I.

After calibration, the location of an object (in mm) in the workspace is given by solving (1) for X and Y. In this equation, f_x and f_y are the focal length of the camera in pixels, s is the skew coefficient, and c_x and c_y are the optical center in pixels of the length and width. R and T are the rotation and translation values of how the world is transformed relative to the camera (extrinsic matrix) while x and y are the pixel's locations. The extrinsic matrix is calculated by the location of the checkerboard and the world coordinates origin

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & T_1 \\ R_{21} & R_{22} & T_2 \\ R_{31} & R_{32} & T_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \quad (1)$$

Finally, using the transformation matrix K_c^w, the coordinate system is transformed from the edge of the checkerboard to the corner of the workspace, as shown in Fig. 3. Using (2)–(4), it is possible to take the location of an object and transform its coordinates from the camera coordinates to world coordinates. Solving (4) yields a 4×1 matrix containing the X and Y locations of the object with respect to the world origin.

$$K_c^w = \left[\begin{array}{ccc|c} & R_C^W & & P_C^W \\ & 3 \times 3 & & 3 \times 1 \\ 0 & 0 & 0 & 1 \end{array}\right] \quad (2)$$

$$R_C^W = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad P_C^W = \begin{bmatrix} -477 \\ -251 \\ 0 \end{bmatrix}, \text{ and } p^c = \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

$$p^w = K_c^w * p^c = \begin{bmatrix} X_{\text{workspace}} \\ Y_{\text{workspace}} \\ 0 \\ 1 \end{bmatrix}. \quad (4)$$

The overhead camera's resolution (~0.5 mm/px) is not high enough to place the SE screwdriver on the head of a screw. When determining the circle's center, an error of a few pixels could result in an error in the placement of the SE screwdriver of a few millimeters. A camera mounted on the robot's head needs to be used to center the SE screwdriver with respect to a hole. Since the distance from the center of the camera to the tip of the SE screwdriver is known, once the screw is centered, the robot can be moved this offset and place the SE screwdriver directly on the screw. This camera does not need to go through the same calibration process because the center of a lens does not have distortion. The robot camera was removed from the plastic housing, and a plano-convex optical lens (10-mm diameter, 50-mm focal length Edmunds Optics) was used to make the camera focus on the laptop at short range. A lighting fixture consisting of four white LEDs with a large viewing angle ($>100°$) was attached to the outside of the webcam.

B. Sensor-Equipped Screwdriver

The motivation behind this tool was to create a screwdriver capable of probing and removing Philips head screws from a laptop in a nondestructive manner. The screwdriver needed to be able to determine if a screw was located at a position identified by the vision system and when the screw had been completely removed from the hole. The SE screwdriver, shown in Fig. 4, consists of an inner shell and an outer shell, connected by a low-friction linear slide that allows the shells to move relative to one another. A small extrusion on the top of the inner shell triggers an FSR when the tip of the SE screwdriver makes contact with a surface. A pneumatic cylinder connected to the outer shell allows the tool to keep a constant pressure on a screw when it is trying to remove it and to retract above the product when the robot needs to move.

An accelerometer on top of the outer shell is used to determine when a screw has been completely loosened from a screw hole. Due to the geometry of a screw thread, when a screw thread ends and continues to rotate, it will fall downward every complete rotation, creating an acceleration along the axis of the screw which can be picked up by the accelerometer. The signal from the accelerometer is amplified and sent through

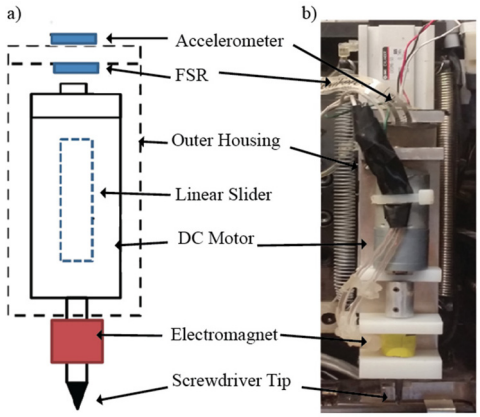

Fig. 4. (a) Sketch of screwdriver. (b) Prototype of screwdriver. (Linear slider is hidden behind shell.)

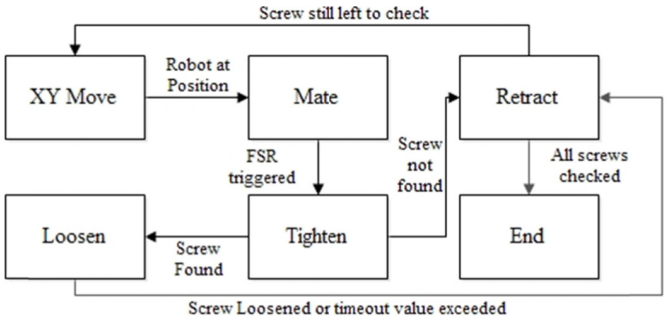

Fig. 5. SE screwdriver state transition diagram.

a set-reset latch circuit on a custom Arduino shield, which makes sure two pulses from the accelerometer are detected before it signals the screw has been loosened. The driving force of the SE screwdriver is a dc motor controlled by an Arduino Motorshield R3 which allows for the motor's current to be easily monitored. If the SE screwdriver is engaged with a screw, the current will increase as the motor stalls out. The SE screwdriver also has an electromagnet that surrounds the tip and when energized, magnetizes it allowing the SE screwdriver to remove the screw from the hole.

IV. Testing Methodology

A. Sensor-Equipped Screwdriver Tests

The SE screwdriver follows the state transition diagram shown in Fig. 5, starting in the state "XY Move" where it moves to a desired position. After the SE screwdriver has reached that position, the state changes to "Mate," the pneumatic cylinder extends, and the screwdriver lowers toward the part. When the tip of the screwdriver makes contact with the part, the pressure supplied by the pneumatic cylinder will keep the outer shell in place which allows the inner shell to move upward and trigger the FSR. The state will change to "Tighten," and the SE screwdriver will check to see if a screw is present by either having the accelerometer trigger (screwdriver motion akin to stripping a screw) or the motor current increase (motor stalled out). If neither of these conditions are is met, and a timeout value is exceeded, then no screw is present and the state will move to "Retract." If a screw is found, the state will change to "Loosen" where the screwdriver will loosen the screw until the accelerometer triggers (screw removed) or a timeout value is exceeded. This process will repeat for all of the screws that are found.

An aluminum block ($50\times70\times25.4$ mm) containing six rows for different screw types (#10-32 [3/4", 1/2", 1/4"], #10-24 [3/4", 1/2", 1/4"], #8-32 [1/2", 1/4"], #6-32 [1/2", 1/4"], #4-40 [1/4"], #2-56 [1/4"]) with four holes each was used to test how well the SE screwdriver could remove screws. The holes in the block not containing screws were used to test how well the SE screwdriver could distinguish between holes with and without screws. Five trials were performed (120 holes) in which the screw removal order was defined, and three trials (72 holes) were performed in which the order was randomized. The locations of the center of the holes were preprogrammed into the system for these tests.

B. Computer Vision Algorithm

The computer vision algorithm used to find the holes is shown in Fig. 6. An image is taken and a 5×5 Gaussian blur (standard deviation [std] $= 5$) and Prewitt edge detection (sensitivity threshold value [stv] $= 0.75$) are applied. The edges are dilated, any enclosed regions filled in, and all clusters touching the borders are removed. A structured diamond element is used to erode the image and find any connected components. The original image and results of these steps are shown in Fig. 7. If no connected components are found, then the stv is decreased by 0.03 and the process repeats. If the stv reaches 0, it is reset to 0.75 and std value is increased by 2. If the std reaches 15 and a hole is still not found, the std is reset to 2, the stv is reset to 0.75, and the algorithm will repeat the previous steps again but use a disk structured element to close any contours in the image. When a connected component is found, its area and length to width ratios are checked to ensure they are within certain bounds. If so, the centroid is used to center the hole with the robot webcam.

C. Laptop Screw Removal Tests

A set of 36 trials was performed that varied the type of lighting, the cameras brightness levels, and the color of laptop cases. The overhead camera brightness level was varied at 30, 90, and 150, and the robot camera brightness level was varied at 30 and 90. The two types of lighting tested were overhead lights mounted in the frame and the localized lighting fixture around the robot camera. Three different laptops models were used and consisted of light, medium, and dark cases.

A flowchart detailing the disassembly process is shown in Fig. 8. An initialization process homes the robot and the laptop is clamped. Then a snapshot of the workspace is taken, the image is undistorted, and circles are located on the laptop using the Hough circle transform (Tao Peng, MATLAB Central). The program moves the robot camera over every circle and attempts to detect a hole. The distance from the center of the camera to the center of the centroid is calculated, and the robot is moved to offset this error until the error is

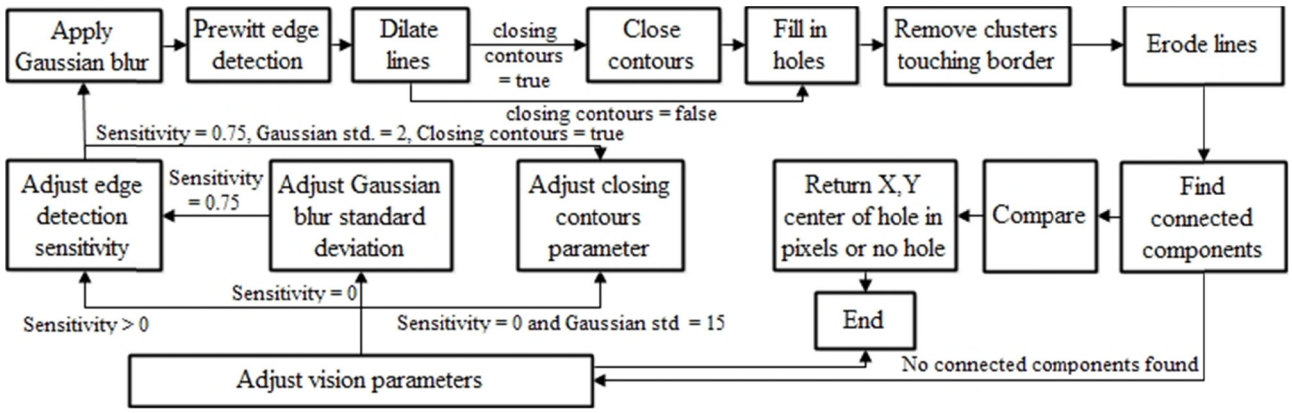

Fig. 6. Computer vision logic.

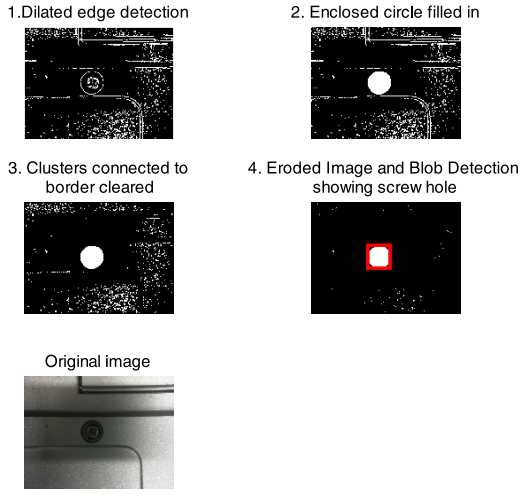

Fig. 7. Computer vision results at different stages in the screw hole detection process.

less than 0.5 mm. After the hole has been centered, the SE screwdriver is moved over the hole to check for a screw. If a screw was found, then other circles within a certain distance (10 mm) are discarded. This entire process is repeated for all remaining holes. A visual of the robot performing these steps is shown in Fig. 9.

V. Results

A. SE Screwdriver Testing Results

A summary of how well and with which method the SE screwdriver was able to determine if a screw was present is shown in Table II. When the screw removal order was preset, the SE screwdriver was able to correctly determine the presence or lack of presence of a screw in all 120 holes. In 60 of the holes, there was no screw present. In the other 60 holes in which a screw was present, the accelerometer was used to determine the screws presence 56 times (93.3%), while current monitoring was used four times (6.7%). When the removal order was random, it was correct in all 72 cases and 19 (52.8%) times it used the accelerometer and 17 times (47.2%) it used current monitoring to determine the screw presence.

TABLE II
Summary of Method Used to Find Screws

			Method used to determine screw presence	
Test	Percentage	No Screw	Accelerometer	Motor Current
Set Removal Order (120 holes)	100	60	56	4
Random Removal Order (72 holes)	100	26	19	17

The time the SE screwdriver took to unscrew the different lengths and types of screws is shown in Fig. 10. There is a positive linear relationship between the length of the screw and the time that it takes to unscrew it as well as a positive relationship in the amount of threads per inch that a screw has and the amount of time it takes to unscrew it.

B. Automated Laptop Screw Removal Test

The results of the automated screw removal test are shown in Table III, where the first four columns list the parameters varied and the next four present the results. The number of circles, located by the overhead camera, is listed in the column "Total Number of Circles Found". These circles include screw holes on the laptop, in addition to shapes or geometries perceived as circles. Every circle represents a location that the SE screwdriver has to move to and check during a test. The number of screw holes that were correctly identified by the overhead camera is listed in the column "Total Number of Screw Holes Found." The column "Total Number of Screw Holes in Workspace" lists the number of screw holes that is present in the reachable workspace on the laptop. The column "Percentage of Screw Holes Found" is the "Total Number of Screw Holes Found" divided by "Total Number of Screw Holes in Workspace." Finally, the row labeled "Total" shows cumulative statistics for all of the trials for each model of laptop.

The results show the localized lighting helps when the laptop case is darker as the average amount of screws found goes from 16.7% to 30.8%. For the dark laptop, the best

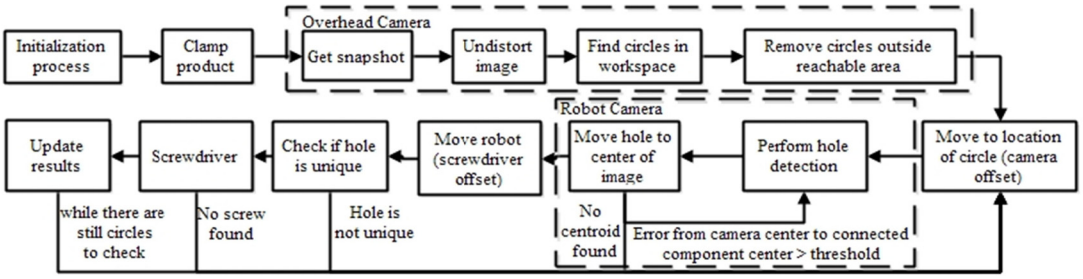

Fig. 8. Flowchart of automated disassembly process.

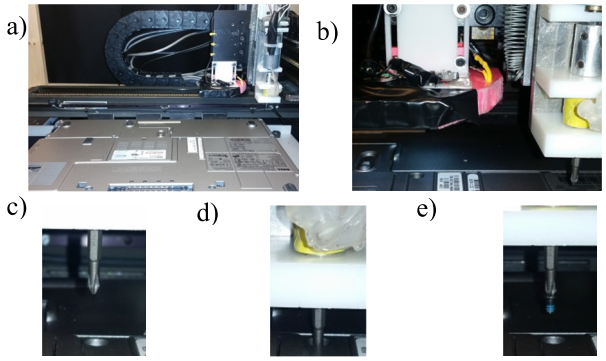

Fig. 9. (a) Robot homes itself and finds circles in the workspace. (b) Robot moves the camera with localized lighting over the circle to center a hole. (c) Robot moves the SE screwdriver over the hole. (d) SE screwdriver is lowered into the hole and removes the screw. (e) SE screwdriver removes the screw from the hole.

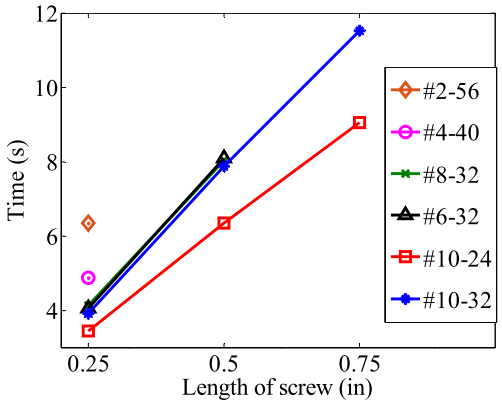

Fig. 10. Amount of time taken to remove screws of different lengths and threads.

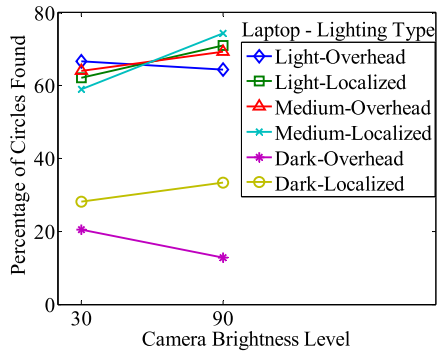

Fig. 11. Percentage of screws found versus robot camera's brightness and laptop color.

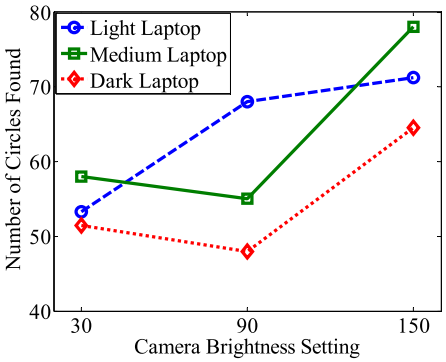

Fig. 12. Number of circles found versus overhead cameras brightness level.

combination is when the overhead camera brightness is 150, robot camera brightness is 90, and localized lighting is used. The localized lighting eliminates shadows cast by the robot as it illuminates the laptop from under the robot. The results for varying the lighting and brightness parameters of the robot camera are presented in Fig. 11 and show a combination of using the localized lighting and higher robot camera brightness increases the system's ability to find a hole. Table III shows the results were best for each laptop model for all the trials when localized lighting was used and the robot camera brightness was 90.

The number of circles located is dependent on the overhead camera brightness. Fig. 12 shows the overhead brightness has a positive linear relationship on the number of circles found, which, up to a certain point, increases the chance that more screw holes will be found. The lighter the case, the faster this point will occur as the entire image will eventually turn white. A tradeoff for allowing more circles to be found is that the time of the test increases. As the laptop case color gets darker, the system's ability to locate holes increases as the overhead camera brightness increases.

Fig. 13 shows an image summary of the results for the laptops with different color cases. The reachable region of the workspace is shown by the enclosed area of the rectangle. A "◯" represents a screw was found and removed at that location. An "X" indicates the computer vision system found a circle, but a screw was not found when probed by the SE screwdriver. A "+" indicates the circle found was discarded because it was within 10 mm of a screw hole, and a "△" shows

TABLE III
SUMMARY OF IDENTIFYING SCREW HOLES IN LAPTOPS WITH DIFFERENT PARAMETERS

Laptop Model	Localized Lighting	Robot Camera Brightness	Overhead Camera Brightness	Total Number of Circles Found	Total Number of Screw Holes Found	Total Number of Screw Holes in Workspace	Percentage of Screw Holes Found
Light	No	30	30	50	12	15	80.0
			90	70	11	15	73.3
			150	82	7	15	46.7
		90	30	53	11	15	73.3
			90	65	11	15	73.3
			150	69	7	15	46.7
	Yes	30	30	53	10	15	66.7
			90	68	11	15	73.3
			150	71	7	15	46.7
		90	30	57	13	15	86.7
			90	69	11	15	73.3
			150	63	8	15	53.3
Total					119	180	66.1
Medium	No	30	30	60	8	13	61.5
			90	59	7	13	53.8
			150	77	10	13	76.9
		90	30	52	8	13	61.5
			90	50	10	13	76.9
			150	84	9	13	69.2
	Yes	30	30	60	7	13	53.8
			90	52	9	13	69.2
			150	72	7	13	53.8
		90	30	60	9	13	69.2
			90	59	9	13	69.2
			150	79	11	13	84.6
Total					104	156	66.7
Dark	No	30	30	58	4	13	30.8
			90	54	2	13	15.4
			150	62	2	13	15.4
		90	30	45	2	13	15.4
			90	48	2	13	15.4
			150	67	1	13	7.7
	Yes	30	30	53	4	13	30.8
			90	43	2	13	15.4
			150	60	5	13	38.5
		90	30	50	3	13	23.1
			90	47	3	13	23.1
			150	69	7	13	53.8
Total					37	156	23.7

that the computer vision system was not able to find a circle. The "□" also means the computer vision system did not find a screw at that particular circle location; however, at this location, the robot made at least one movement.

The time statistics for the tests are summarized in Table IV. These results show that the clamping time is not significant as it is an order of magnitude smaller than the robot movement time and screw removal time (srt), and two orders of magnitude smaller than the computer vision time (cvt). The cvt is the greatest due to the number of circles that need to be explored and the various parameters associated with the computer vision algorithm. It should be noted that the srt time for the dark laptop is significantly lower because there were fewer holes detected.

The performance of the SE screwdriver does not depend on how well the computer vision system worked or even the model of the laptop used. In order to not penalize the SE screwdriver for "missing" holes that the computer vision module was not able to find, the performance of the SE screwdriver is defined as the "screws found" divided by

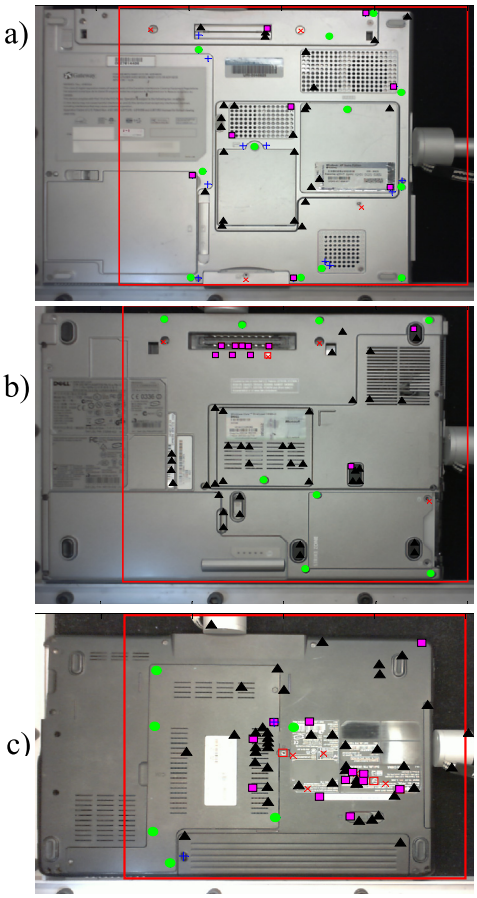

Fig. 13. (a) Results for the laptop with a light case. (b) Results for the laptop with the medium case. (c) Results for laptop with a dark case. Circles indicate the screwdriver was able to determine a screw while Xs indicate that the screwdriver was not able to determine a screw. Plus signs indicate the screw hole was discarded due to proximity to a screw that was found, and triangles and squares mean the computer vision algorithm was not able to determine a hole.

TABLE IV
TIMING RESULTS FOR AUTOMATED SCREW REMOVAL TEST

	Clamping Time (s)		Robot Movement Time (s)		Computer Vision Time (s)		Screwdriver Time (s)	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Light	11.2	0.8	337.8	60.0	2671.7	967.9	166.2	36.4
Medium	8.8	0.4	357.4	55.2	2654.8	509.3	139.8	31.0
Dark	10.1	0.4	294.7	43.1	3013.3	332.2	55.3	26.0

total "screw holes that the computer vision found." The SE screwdriver performed well and correctly removed 251/260 (96.5%) of screws in screw holes. In 27 of the 36 trials of the laptop screw removal test, the SE screwdriver determined if a screw was in the hole correctly 100% of the time. Even when the SE screwdriver missed a hole on a laptop, it never missed more than one.

VI. CONCLUSION

This brief presented a new method that combines vision and force sensing in order to automatically find and remove screws from a laptop without preprogramming their locations. This method would be the first step in disassembling laptops or other products with a plastic casing covering internal electronics. The results presented in this brief show that this is a viable strategy for removing screws from laptops and could be incorporated into a system that uses a cognitive architecture, such as one presented by Kara *et al.* [22] and Vonbunyong *et al.* [23], [24], to facilitate a disassembly process to recycle and reuse materials from e-waste.

A significant challenge is trying to automatically locate screw holes on the laptop. The results show that this is composed of two tasks. The first involves finding holes with a computer vision module, and the second involves probing the hole with an SE screwdriver. The lighting and the brightness of the cameras are the two major parameters that need to be adjusted and are dependent on how dark the laptop case is. The darker the laptop case is, the higher the brightness the cameras should use. The higher the brightness in the overhead camera, the more circles will be found and the greater chance that all of the screws will be found. A greater brightness with the robot camera creates a higher contrast with a hole making it easier to find. The best results occurred when using the localized lighting as it was able to light the workspace from underneath the robot, eliminating shadows that would occur if the robot was lit from above. A screwdriver tool was also introduced that was able to tell if a screw was present and when that screw had been completely loosened by using an accelerometer.

A current drawback of this system is the length of time it takes to complete a test. Next, the cognitive architecture Soar will be added to the robot and will allow it to remember the locations of the holes to reduce the time. Soar described in [25] will be able to use the methods presented in this brief to find screws and be used in order to select the hole to remove a screw from. The time will be reduced because if a circle has already been explored and a screw was not found that location will not need to be explored again. Optimization of different image parameters such as the brightness levels for the different laptops or parameters governing the Hough circle transform, the amount of screws that are found on each laptop could be improved.

REFERENCES

[1] A. Chen, K. N. Dietrich, X. Huo, and S. Ho, "Developmental neurotoxicants in E-waste: An emerging health concern," *Environ. Health Perspect.*, vol. 119, no. 4, pp. 431–438, Apr. 2010.
[2] I. C. Nnorom and O. Osibanjo, "Overview of electronic waste (E-waste) management practices and legislations, and their poor applications in the developing countries," *Resour. Conserv. Recycl.*, vol. 52, no. 6, pp. 843–858, Apr. 2008.
[3] C. P. Baldé, F. Wang, R. Kuehr, and J. Huisman, "The global E-waste monitor—2014," United Nations Univ., IAS-SCYCLE, Bonn, Germany. Tech. Rep., 2015.
[4] (2013). *A Roadmap for U.S. Robotics From Internet to Robotics*, accessed on Sep. 18, 2016. [Online]. Available: https://robotics-vo.us/sites/default/files/2013RoboticsRoadmap-rs.pdf
[5] P. Schumacher and M. Jouaneh, "A force sensing tool for disassembly operations," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 2, pp. 206–217, Apr. 2014.
[6] A. A. Gopalai, S. M. N. A. Senanayake, and D. Gouwanda, "Determining level of postural control in young adults using force-sensing resistors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 4, pp. 608–614, Jul. 2011.

[7] C. Castellini and V. Ravindra, "A wearable low-cost device based upon Force-Sensing Resistors to detect single-finger forces," in *Proc. 5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatronics*, Aug. 2014, pp. 199–203.

[8] C. Zerpa, D. Jackson, P. Sanzo, and D. Kivi, "Examining the reliability of a force sensing resistor as a possible tool to assess carpal tunnel syndrome," *Sci. Technol.*, vol. 5, no. 1, pp. 1–4, 2015.

[9] R. A. Romeo *et al.*, "Development and preliminary testing of an instrumented object for force analysis during grasping," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2015, pp. 6720–6723.

[10] U. Rebafka, G. Seliger, A. Stenzel, and B. R. Zuo, "Process model based development of disassembly tools," *Proc. Inst. Mech. Eng, B, J. Eng. Manuf.*, vol. 215, no. 5, pp. 711–722, May 2001.

[11] J.-J. Park and G.-S. Kim, "Development of the 6-axis force/moment sensor for an intelligent robot's gripper," *Sens. Actuators A, Phys.*, vol. 118, no. 1, pp. 127–134, Jan. 2005.

[12] K. Feldmann, S. Trautner, and O. Meedt, "Innovative disassembly strategies based on flexible partial destructive tools," *Annu. Rev. Control*, vol. 23, pp. 159–164, 1999.

[13] B.-R. Zuo, A. Stenzel, and G. Seliger, "A novel disassembly tool with screwnail endeffectors," *J. Intell. Manuf.*, vol. 13, no. 3, pp. 157–163, Jun. 2002.

[14] G. Seliger, T. Keil, U. Rebafka, and A. Stenzel, "Flexible disassembly tools," in *Proc. IEEE Int. Symp. Electron. Environ.*, May 2001, pp. 30–35.

[15] J. R. Peeters, P. Vanegas, C. Mouton, W. Dewulf, and J. R. Duflou, "Tool design for electronic product dismantling," *Procedia CIRP*, vol. 48, pp. 466–471, 2016.

[16] P. Schumacher and M. Jouaneh, "A system for automated disassembly of snap-fit covers," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 9, pp. 2055–2069, Dec. 2013.

[17] G. Tian, M. Zhou, J. Chu, and Y. Liu, "Probability evaluation models of product disassembly cost subject to random removal time and different removal labor cost," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 288–295, Apr. 2012.

[18] G. Tian, M. Zhou, and J. Chu, "A chance constrained programming approach to determine the optimal disassembly sequence," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 4, pp. 1004–1013, Oct. 2013.

[19] Y. Tang and M. C. Zhou, "A systematic approach to design and operation of disassembly lines," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 3, pp. 324–329, Jul. 2006.

[20] H. J. Kim, R. Harms, and G. Seliger, "Automatic control sequence generation for a hybrid disassembly system," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 2, pp. 194–205, Apr. 2007.

[21] X. Guo, S. Liu, M. Zhou, and G. Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2435–2446, Nov. 2016, doi: 10.1109/TCYB.2015.2478486.

[22] S. Kara, S. Vongbunyong, and M. Pagnucco, "Vision-based execution monitoring of state transition in disassembly automation," *Int. J. Autom. Technol.*, vol. 10., no. 5, pp. 708–716, 2016.

[23] S. Vongbunyong, S. Kara, and M. Pagnucco, "Vision-based execution monitoring of state transition in disassembly automation," *Robot. Comput. Integr. Manuf.*, vol. 34, pp. 79–94, 2015.

[24] S. Vongbunyong, S. Kara, and M. Pagnucco, "Application of cognitive robotics in disassembly of products," *CIRP Ann-Manuf. Technol.*, vol. 62, no. 1, pp. 31–34, 2013.

[25] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: An architecture for general intelligence," *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, Sep. 1987.