IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 5, 2018, accepted June 10, 2018, date of publication June 13, 2018, date of current version July 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2846761

Image Rectification Software Test Automation Using a Robotic ARM

DEBDEEP BANERJEE, (Member, IEEE), KEVIN YU, AND GARIMA AGGARWAL
Qualcomm Technologies, Inc., San Diego, CA 92121, USA

Corresponding author: Debdeep Banerjee (debdeepb@qti.qualcomm.com)

ABSTRACT In this paper, we discuss the use of a robotic arm for testing phone software features, such as image rectification, on mobile devices. The problem statement is that we needed an accurate and a precise test automation system for testing and validating the computer vision algorithms used for image rectification in a mobile phone. Manual testing may be error-prone and tedious and thereby the need for a reliable test automation system is of utmost necessity to check the software quality of the image rectification algorithms. The solution to this problem was to design and develop a test automation system using a robotic arm to validate the image rectification algorithms. The robotic arm-based software test automation was deployed and has performed functional performance-based stability tests on multiple software products. The reason for using a robotic arm setup is because it provides us with the flexibility to run our test cases using different speeds, rotation angles, and tilting angles. In this paper, we describe how the robotic arm rotation works. We first measure the center coordinate of the test subject relative to the base of the robotic arm. Then, a 3-D model of the subject is created with those coordinates via simulation mode to represent the real distance ratio setup. Then, the tip of the robotic arm is moved to the proper distance facing the subject. The tests were executed with clear and blurry images containing text with and without image rectification enabled. The result shows the increase in accuracy of text recognition with image rectification algorithm enabled. This paper talks about the design and development of the test automation for the image rectification feature and how we have used a robotic arm for automating this use case.

INDEX TERMS Software engineering, software testing, robots, robotics and automation.

I. INTRODUCTION

This technical paper focuses on software test automation of the software features such as image rectification, which is often used in mobile phones.

Image rectification is the process of running algorithms to help an image become sharper, remove noise (via the moiré effect), enhance contrast and enhance colors. Image rectification features are used for rectifying image quality as well.

With the explosion of image processing algorithms being commercialized on phones it is extremely important to design and develop reliable methods of testing these algorithms to ensure quality products. We discuss the approach of using a robotic arm for developing test automation for the image rectification algorithm. Image rectification algorithms needs extensive testing based on the different processing and computing of the processors.

The robotic arm has significantly improved the accuracy of the test execution and enabled us to exercise these tests on multiple software products. We have presented results and accuracy measurements of the robotic arm while performing the image rectification tests.

From the tester's point-of-view, we have test plans to test functionality features such as performance, power and stability. We have developed tests for different stages of the software integration cycle, which includes component model tests (at the developer check-in level) and sanity for every software builds. We also run our cumulative tests during the lifecycle of a product line. The concept of adding a test ahead of the software integration cycle can enable us to catch regressions earlier in the software cycle. This test has led to faster fix rates because the bug can be relatively and easily isolated. Since the component model often changes, many other technologies, such as Wi-Fi, have not been incorporated into the software builds because it is easier to isolate the issue in a specific tech area.

The challenge was to develop software test automation for validating application programming interfaces exposed to the image rectification software feature. The image rectification test automation includes functional, stability, performance and quality tests. This validation strategy includes testing end-to-end applications, power use cases, fuzzing, code coverage, competitive analysis and developing post-processing tools (using MATLAB) for validating the output frames. The introduction of the robotic arm for executing the feature testing, such as image rectification has significantly cut down the manual test times and has improved precision. We use a framework for test execution and reporting. Automated software testing is needed more than ever in this emerging mobile world. This paper focuses on the problems, proposes solutions and provides validation results. This study also discusses the future scope of the work.

II. MOTIVATION

The test automation has provided many key benefits like improved test efficiency, scalability, and autonomy, as well as allowing us to test more comprehensively to find issues more quickly.

A. INCREASE TEST COVERAGE

The test automation has provided us the capability to increase test coverage and execute the automation simultaneously over multiple product lines. Therefore, we have more than doubled the number of test cases and added more test images and test vectors. Software test automation has allowed us to increase the test requirements of this feature, while the resources with which to execute the tests have remained constant.

B. REDUCE MANUAL TESTING

The goal of developing the test automation is to have test engineers focus on test development of the new features and have the regression tests executed through automated software. For example, as automation systems evolve and become more robust, human intervention becomes less essential. The post-processing of test results for the image rectification features has also been automated. A huge challenge to executing the tests for this feature is that we need someone to run the tests manually with different camera angles, panning speeds, lighting conditions and color backgrounds and then to verify visually and qualitatively how well image has been rectified of color saturation, moiré effect, etc.

C. IMPROVE TEST EFFICIENCY AND SCALABILITY

Iteration stability tests for executing tests such as rotating the device at a specific angle in front of the test subject and keeping it parallel to the test subject multiple times is extremely difficult to perform by human test engineers. The use of the robotic arms for testing these advanced multimedia features is critical to product commercialization and smooth test execution. A critical goal of automated testing is to find faults with the shortest possible test sequences. The more tests that can be run at any given time, then the quicker we can catch issues and the more thorough testing we can achieve. Moreover, we can commercialize a product more quickly and therefore support more product lines.

Not only has automation process allowed us to meet the growth of the computer vision project, but it has become a necessity as we continue to support more software products and more features.

III. BACKGROUND AND RELATED WORK

Robot usage has been commonly associated with growth and the need for more mobile software testing [1]. The need for automation and high-quality test execution has increased for the exploding new feature sets of mobile software applications [2]. The emphasis of using the mobile phone as a desktop is increasing fueling the need for a rapid increase in the processing capabilities.

Robots have been used by the industry to perform tasks to facilitate the automation of repetitive tasks. Software testing has also greatly benefited the end-user GUI (graphical user interface) based testing using robots.

The usage of a robotic arm for software testing is prevalent as we can precise use a robot to repeat a specific test sequence accurately [3]. The robots have been used for performing graphical user interface testing etc. Black box testing involves testing of a software program without knowing the internal implementation of the software functions and its building blocks. The software program can be tested in a black box technique if the inputs and the expected output is known. A robot setup can be efficiently used for software testing as it can precisely repeat the test sequences and the test results can then be analyzed [4]–[9].

Robots can be used as flexible, reconfigurable mode. The robot which achieves rectilinear locomotion by coupling structural deformation and directional friction promoting a locomotion strategy ideal for traversing narrow channels [11], [12]. Testing the software using robot-based systems is challenging because physical systems are necessary to test many of their characteristics [13], [14].

Robot Framework is an open source software and based on the Python language keyword driven automated test framework [14]. The challenges of robotic software testing extend beyond conventional software testing. Valid, realistic and interesting tests need to be generated for multiple programs and hardware running concurrently, deployed into dynamic environments [15].

Manual testing is a time-consuming process. In addition, regression testing, because of its repetitive nature, is error-prone, so automation is highly desirable. Robot Framework is simple, yet powerful and easily extensible tool which utilizes the keyword driven testing approach [16].

IV. SOLUTION

The solution was to design and develop a robust image rectification test automation using a robotic arm.

A. OVERVIEW

Image rectification is a feature that rectifies the text region of an image. The algorithm also enhances that region of the image for better text recognition. The image rectification algorithm helps image sharpening, noise (via moiré effect) removal or contrast enhancement of the rectified image.

From the testing perspective, we have test plans that test the feature's functionality, performance, power, stability, etc. We have organized the test plan in test suites. We have added computer vision tests in different stages of the software integration for performing functional and performance tests.

B. OBJECTIVE

Traditional manual testing on image rectification took several weeks to complete and involved several testers to take the device to different parts of the test setup for image background requirements. Additionally, testers had to constantly bend down and adjust the angle to hold the device for some test cases. These manual testing steps were inefficient and may lack precision. We envisioned to have these tests automated, but it was not an easy task. The test cases involved many physical position changes of the device. Therefore, we were searching for a way to mitigate this constraint and discovered a solution to our problem by utilizing the robotic arm. Thanks to the robotic arm, we have the automated many of the manual test cases. Now we can complete the testing in 1/10th time compared to manual testing.

C. TEST SETUP

We have implemented specific lighting conditions and safety features for operating the robotic arm. The robotic arm system contains three parts: backlight illuminated bench with a camera, DENSO Robotic Arm, and a test subject bench.

We use a DENSO VS-Series six-axis articulated robot. There is some clear advantage of using this robotic arm in our test automation. We offer high precision, which is important when we are required to rotate the device to a certain angle. The robotic arm keeps the device at a precise distance with several objects in some test cases. The compact design of the robotic arm helps us to save space. The six-axis motion design makes it highly flexible for the relevant motion. We can perform continuous motions and difficult angular motions that the human arm may never do.

D. DIFFERENT BACKGROUNDS

For every test case of image rectification, there is a configuration on the background setup. One of the functions in image rectification is subject detection, which detects document region based on a rectangular shape. We test this feature for drawing the document boundary correctly from every type of background, as shown in Table 1.

Testers used to take the device under test to the available backgrounds in the lab and take snapshots, then switch to another background. In our test automation setup, we have collected most of the background and subject combination needed. We can add more combinations if required. We also have a PC monitor which can display multiple pictures in a slideshow (We can analyze the image after the removal of moiré effect). We use this setup to expand into a testing environment for many more test cases. Please refer to figure 1 for details.

TABLE 1. Examples of the requirements from different test cases.

Test_Case_Name	Subject	Background
ImageCorrection_Portrait_Document _White Wall_0 Subject Rotation	Document	White Wall
ImageCorrection_Portrait_Document _Gray Paper_0 Subject Rotation	Document	Gray Paper
ImageCorrection_Portrait_Coated Document _Gray Paper_20 Subject Rotation	Coated Document	Gray Paper
ImageCorrection_Portrait_Yellow Post it _Cardboard Surface_0 Subject Rotation	Yellow Post-it	Cardboard
ImageCorrection_Portrait_Oxford notebook _Wood table_0 Subject Rotation	Oxford Notebook	Wood table

FIGURE 1. Test setup 1 – Document & Whitewall, Test setup 2 – Yellow Post-it & Cardboard, Test setup 3 – Document & Gray Paper, Test setup 4 – Coated Document & Gray Paper, Test setup with the wooden table 5 – Monitor screen with images.

For test cases with a non-zero subject rotation such as the ''ImageCorrection_Portrait_Coated_Document_GrayPaper_20_Subject_Rotation'' test case, it requires the subject to rotate a 20-degree angle. We have a special setup for this with a mechanism called Programmable Rotor. It can rotate the test subject to a desired programmed angle. Please refer to figure 2.

The robotic arm is efficient in precisely positioning the device under test in front of a test subject for executing the tests. Please refer to Figure 3 for details.

The test automation workflow is mentioned in figure 4.

E. DEVICE ANGLE ROTATION

Image rectification test involves not only the rotation of the subject but also the rotation of the test device itself. The reason for this is that users may not always hold their cameras perfectly aligned to the documents they are trying to

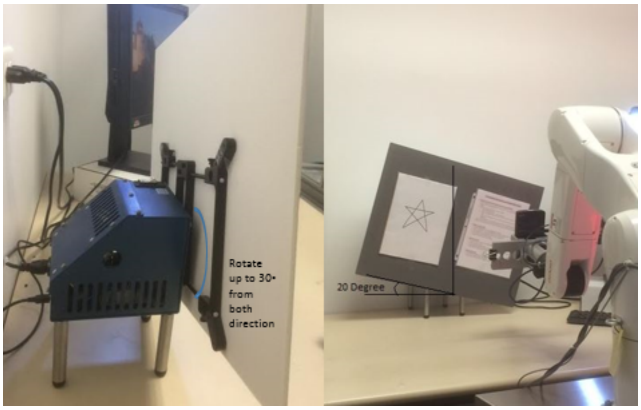

FIGURE 2. **Programmable rotor controlled by the lab station that rotates the test subject to a precise degree angle on demand.**

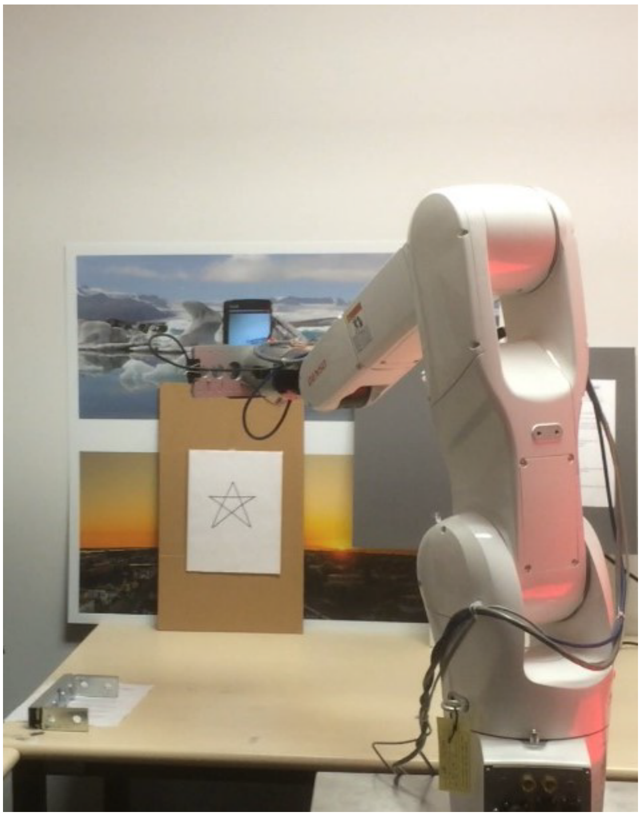

FIGURE 3. **Robotic arm is positioning the device under test in front of the test subject.**

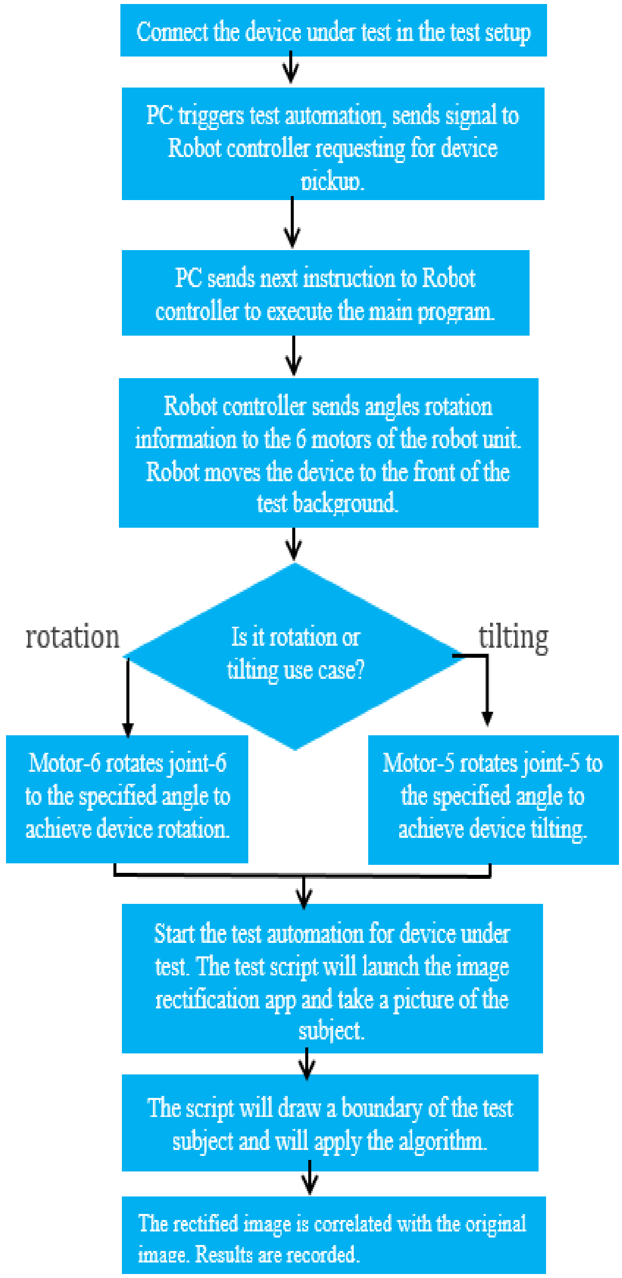

FIGURE 4. **Image Rectification Test Automation with Robotic Arm Workflow.**

rectify. There will always be some small rotation between them. We need to simulate these realistic scenarios and test thoroughly with varying angles to test the limits. The range of rotation is between 0 to 90-degree angles. A 90-degree rotation indicates the use of the camera in landscape mode. Landscape mode is easy to rotate but not so easy for small angle rotations on manual testing. The tester must bend down and align his or her eyesight, the sensor of the device and the test subject must be at the same level and then adjust the test device to the specific angle. The problem with this manual testing is not only that it requires the tester's physical effort but also that the angle of rotation can only be estimated, which impacts the testing precision. This is where the advantage of using robotic arm automation comes in. A robotic arm rotates the device precisely using a pre-programmed angle and saves the time-consuming adjusting process. Please refer to Figure 5.

Here, we describe how the robotic arm rotation works. We first measure the center coordinate of the test subject (X_1, Y_1, Z_1) relative to the base of the robotic arm (X_0, Y_0, Z_0). Then, a 3D model of the subject is created with those coordinates via simulation mode to represent the real

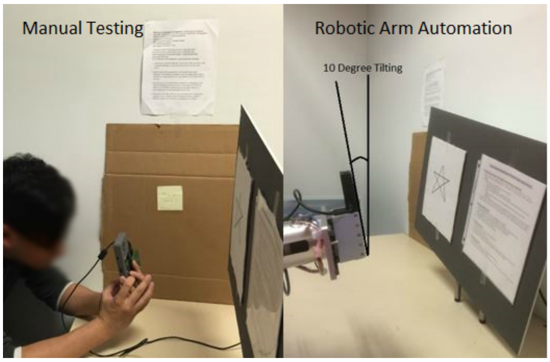

FIGURE 5. Manual Testing setup VS. Robotic Arm Automation setup on test case requires device angle rotation.

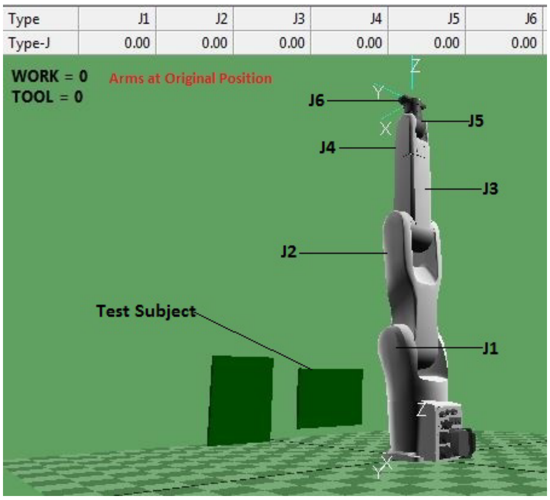

FIGURE 6. Before rotation, arms are at the original position (0, 0, 0, 0, 0, 0).

distance ratio setup. Then, the tip of the robotic arm is moved to the proper distance facing the subject via a user-based eye-level approximation in the simulation. (Please refer to figure 6: Before Rotation and figure 7: Estimated Rotation)

Now the 6 joints rotation coordinates can be found on the simulator as J1 = −21.25, J2 = 64.50, J3 = 91.50, J4 = −27.00, J5 = 63.00, J6 = 18.00. J1, J2 and J3 are the base joints, their values can be set as fixed. J4, J5 and J6 should be adjusted accurately so that the tip of the robotic arm that holds the device will be parallel to the test subject while perpendicular to the ground.

The mathematical logic behind is explained here. Since J1 to J6 only rotate with aspect to Y and Z axis, these rotations can be represented in a plane. (Please refer to figure 8: Angles of Rotation)

First, we need to find the value of J4. Since both J1 and J4 are rotations with aspect to the Z axis, J4 must rotate the same amount in the opposite direction of the J1 rotation to keep the device in the original position. However, since the J4 arm is no longer in the same vertical line with the J1 arm, the J4 arm now has a γ-degree angle from its original position

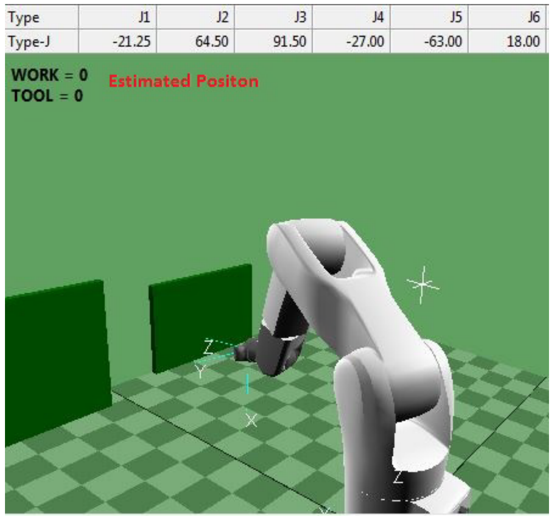

FIGURE 7. The estimated position that a user manually moves is simulated by the robotic arm with tip facing the center of the test subject.

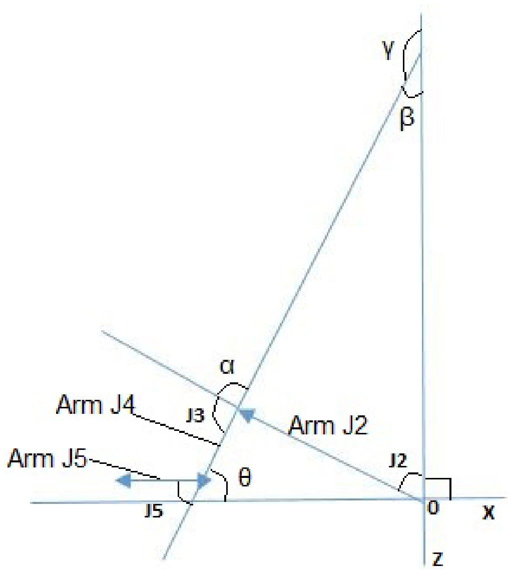

FIGURE 8. Angles of Rotation.

with respect to the Y-axis. J4 rotation must be multiplied by cosine of Y to accommodate the rotation of J1.

Calculation of γ:

$$\text{Angle } \alpha = 180 - \text{J3} = 180 - 91.5 = 88.5$$

$$\text{Angle } \alpha = \beta + \text{J2 so } \beta = \alpha - \text{J2} = 88.5 - 64.5 = 24$$

$$\text{Angle } \gamma = 180 - \beta = 180 - 24 = 156$$

With γ calculated, J4 can be found from this formula:

$$\text{J1} = \text{J4} \times \cos(\text{J2} + \text{J3}) = \text{J4} \times \cos(\gamma)$$

$$\text{J4} = \frac{J1}{\cos(|2 + |3)} = \frac{J1}{\cos(\gamma)} = \frac{21.25}{-0.9135} \approx 23.35$$

For J5, it rotates with aspect to Y axis, J5 must rotate until arm J5 is parallel to X axis.

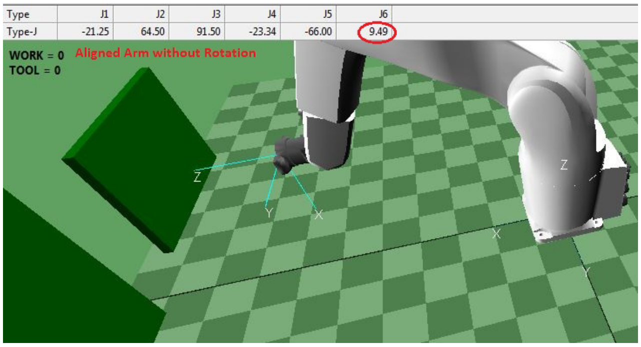

FIGURE 9. Aligned Arm without rotation with J4 J5 J6 accurately adjusted.

Since Arm J5 // X-axis, then J5 $= \theta$.

And $\theta + \beta = 90$, $\theta = 90 - \beta = 90 - 24 = 66$

Since J5 rotates to the opposite direction, it's represented as -66 degrees.

Simplified Formula:

$$\mathrm{J5} = 90 - \mathrm{J2} - \mathrm{J3}$$
$$\mathrm{J5} = -66$$

With J4 and J5 values solved, J6 angle can be calculated. J6 also rotates with aspect to Z axis along with J4. Since J4 rotated 23.25 degree from original position, to keep the phone parallel with Y axis, J6 must rotate with the opposite direction. In addition, since Arm J5 is at -66 degrees with Arm J4, J6 must rotate cosine -66 degree of J4 rotation.

$$\mathrm{J6} = \mathrm{J4} \times \cos(\mathrm{J5}) = 23.35 \times \cos(-66) \approx 9.49$$

With J4 J5 and J6 all calculated, the robotic arm will place the device facing accurately parallel with the test subject and 0 device rotation. (Please refer to figure 9: Aligned Arm without rotation)

The final goal is to be able to rotate J6 motor with any angle with aspect to the test subject, in this case +15 degree. Since we already have the J4 and J5 aligned, we just need to add 15 degrees to J6. The new angle of rotation will be

$$J6_{final} = \mathrm{J6} + 15 = 24.49$$

The robotic arm now has rotated the device to a perfect +15-degree angle. This set of angle rotation data for the 6-axis case can be saved in the program as a point (ex: P1) in the 3D simulation model. To get the device to our designed position, we just need to run the robotic arm to that point: Move P and P1. Please refer to figure 10: After Rotation.

F. TILTING

Image rectification has test cases that involve rectification scenarios with test device tilted in different angles. Most of the time, the user captures the text image holding the device a little bit tilted, resulting in a skewed image. To test such scenarios, the tester must tilt the device to some specific angle. Similar to device rotation, the problems for this manual testing are not only that it requires the tester's physical effort, but the angle of tilt can be only estimated. This impacts the testing precision. Robotic arms help us to overcome from such situations.

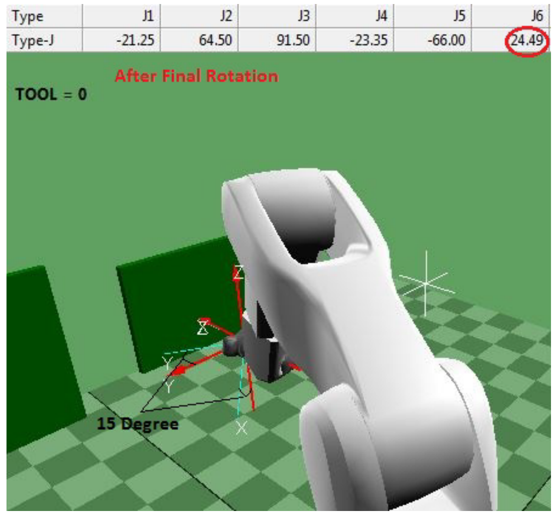

FIGURE 10. 3D simulation of positioning the robotic arm before rotation, and after J6 rotate 15-degree angle. Thick red coordinate axis represents the base axis, the thin one represents the axis of the arm's tip.

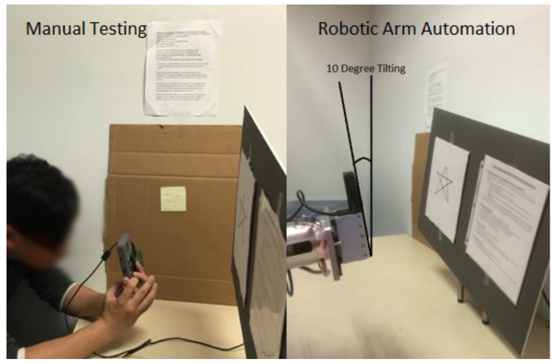

FIGURE 11. Manual Testing setup VS. Robotic Arm Automation setup on test case requires device tilting.

The robotic arm can tilt the device precisely in any pre-programmed angle. This precision saves time and provides accurate test results. Please refer to Figure 11.

Here, the robotic arm works to tilt the phone in the same way it works for the rotation of a phone. We start the robotic arm from the original position and program it to move to a position where the device is parallel to test image and perpendicular to the table, the same position we found in the previous section before rotation. To tilt the device, the robotic arm needs to rotate along the Y direction by rotating the robotic arm's J5 motor. The robotic arm will move the test device by the precise desired tilted angle with the test document.

G. MOIRÉ EFFECT

1) OVERVIEW

In physics, the moiré effect's manifestation is the beat phenomenon that occurs in many wave interference problems. The moiré effect occurs when the device has the camera preview on and is capturing video or snapshots from a monitor.

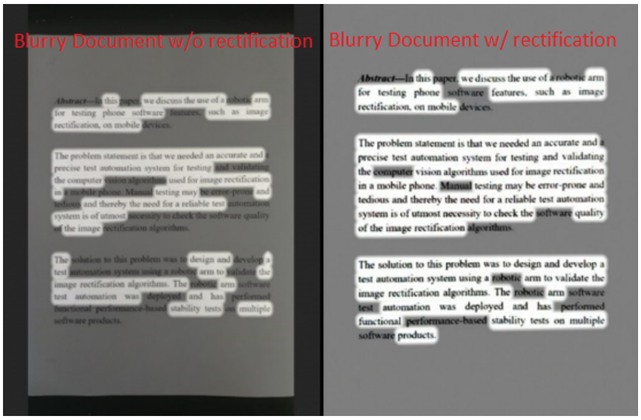

FIGURE 12. Comparison between w/o and w/ image rectification on text recognition results.

Taking snapshots of a monitor causes the moiré effect, which is an unnecessary pattern. The goal of the image rectification app is to remove the moiré effect.

2) ROBOTIC ARM USAGE FOR TESTING THE MOIRÉ EFFECT

For testing moiré effect patterns, we need to test images captured from a monitor. We use the robotic arm setup to take the device under test to a monitor screen and capture snapshots. We have automation, which starts a slideshow on the Windows PC, and we synchronize the snapshot timing in the device with the photo transition timing in the slideshow so that we can capture a new image every time.

V. RESULTS

Image rectification software is designed primarily for enhancing text recognition performance. It sharpens the text on the image and helps making the text darker to create better contrast with the background. The strategy of determining image rectification test result is by feeding a snapshot of a document and the rectified version of that snapshot, both to a 3rd party text recognition software. Let the software performs a text recognition on both images, then compare the amount of text recognized in percentage from the outputs to determine the image rectification effect.

A. RESULTS OF ROTATION USE CASE OF IMAGE RECTIFICATION

We have prepared two printout documents with the same content. One has clear text and good contrast, the other one with blurry text and bad contrast. We use robotic arm to move the test device to the front of both documents to take snapshots one time with image rectification and one time without image rectification. Both image sets were sent to the text recognition software for processing. The result images showed text recognition performed well on the clear document upfront for both with and without rectification. However, on the blurry document we saw clear difference in term of number of text recognized and text quality with image rectification enabled as demonstrated in Fig. 12.

TABLE 2. Rotation scenarios text recognition results for clear and blurry documents.

Scenarios	Clear Document		
	w/o Rectification	w/ Rectification	Delta
0° Rotation	96.1%	98.4%	2.3%
15° Rotation	67.4%	98.4%	31.0 %
30° Rotation	29.5%	97.7%	68.2 %
Scenarios	**Blurry Document with Low Contrast**		
	w/o Rectification	w/ Rectification	Delta
0° Rotation	62.8%	86.8%	24.0 %
15° Rotation	34.1%	83.7%	49.6 %
30° Rotation	5.4%	79.8%	74.4 %

From Fig. 12, we can clearly see the results from image rectification. First, the boundary of the document was correctly detected as the document was extracted from the background. The text on the rectified image was darkened and sharpened. The contrast from the rectified image was much better. This makes the text on the rectified image more readable to human eye and to the text recognition algorithm.

Using the robotic arm program, we executed the automation testing on 15-degree angle rotation and 30-degree angle rotation use cases. The simulated the scenarios text recognition performed on documents were not in the upfront position in some images. Thanks to the advantage of robotic arm's consistency and precision, we got the snapshot of the documents from the precise angles and repeated the process 10 times to generate reliable comparison results. We have averaged the percentage of text recognized for both clear and blurry documents. The results are listed in table 2.

The result data indicated on clear document, without rotation image rectification slightly improved the already good text recognition performance from 96.1% to 98.4% with a 2.3% delta. From 15° rotation, the text recognition performance started to drop 28.7%, and at 30° rotation it dropped 66.6%, while text recognition rate with rectification didn't seem to get affected by the rotation and stayed around 98%. Figure 13 shows the comparison of text recognition rate with and without image rectification on clear document.

The reason for this significant delta on rotation scenario is text recognition algorithm had a hard time processing text that are not horizontally aligned. The more document is rotated the harder it makes to detect words. On the other hand, image rectification software has the feature to extract document from background and auto-rectify it to the upfront position.

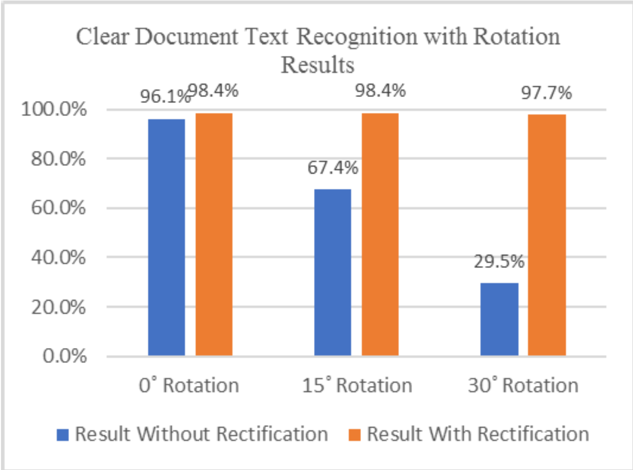

FIGURE 13. Comparison of text recognition rate with and without image rectification on clear document.

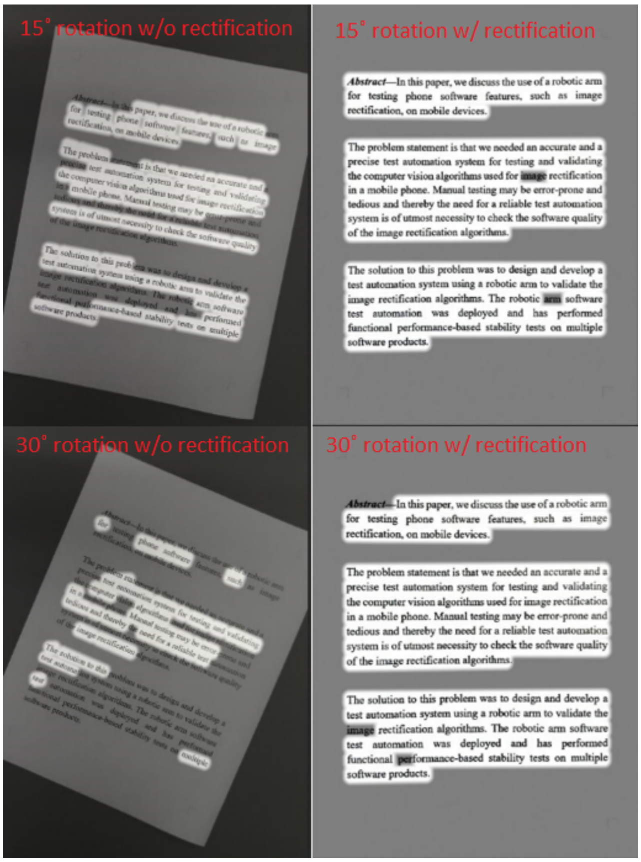

FIGURE 14. Image snapshots for clear document rotation scenarios.

This makes the text recognition much easier compare to rotated document. The image snapshots are demonstrated in Fig. 14.

On the blurry document rotation scenarios, the text recognition performance is impacted even more without image rectification on average of 30% drop from clear document scenarios. Text recognition performance with image rectification has a small steady drop with an average of 14.7% on blurry document. In these scenarios, text recognition rate gets an even larger delta comparing with and without image rectification shown in Fig. 15.

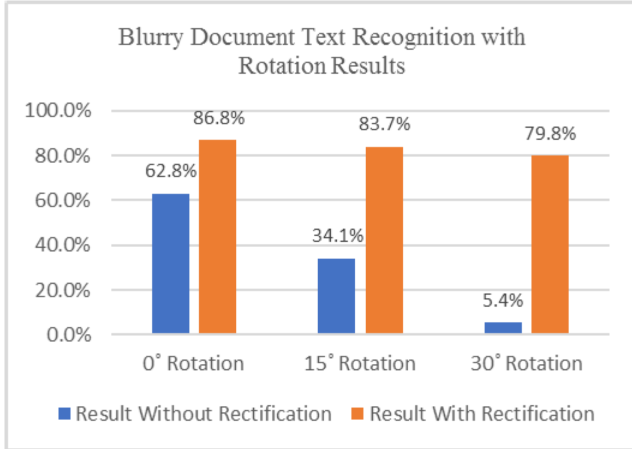

FIGURE 15. Comparison of text recognition rate with and without image rectification on blurry document.

TABLE 3. Tilting scenarios text recognition results for clear and blurry documents.

Scenarios	Clear Document		
	w/o Rectification	w/ Rectification	Delta
10˚ Tilting	94.6%	98.4%	3.8%
20˚ Tilting	93.8%	97.7%	3.9%
30˚ Tilting	83.7%	97.7%	14.0%
Scenarios	**Blurry Document with Low Contrast**		
	w/o Rectification	w/ Rectification	Delta
10˚ Tilting	47.3%	76.0%	28.7%
20˚ Tilting	35.7%	69.8%	34.1%
30˚ Tilting	27.9%	65.9%	38.0%

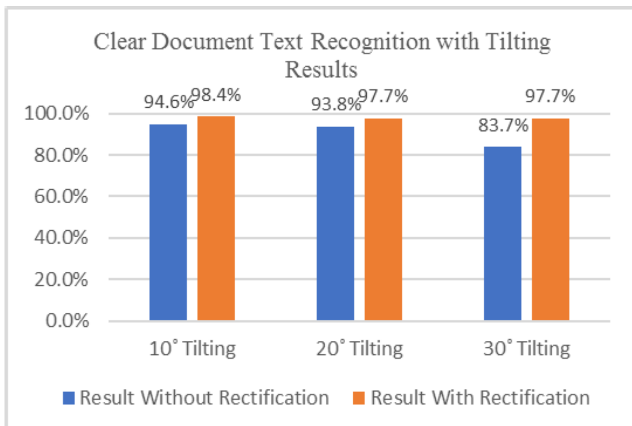

FIGURE 16. Comparison of text recognition rate with and without image rectification on clear document.

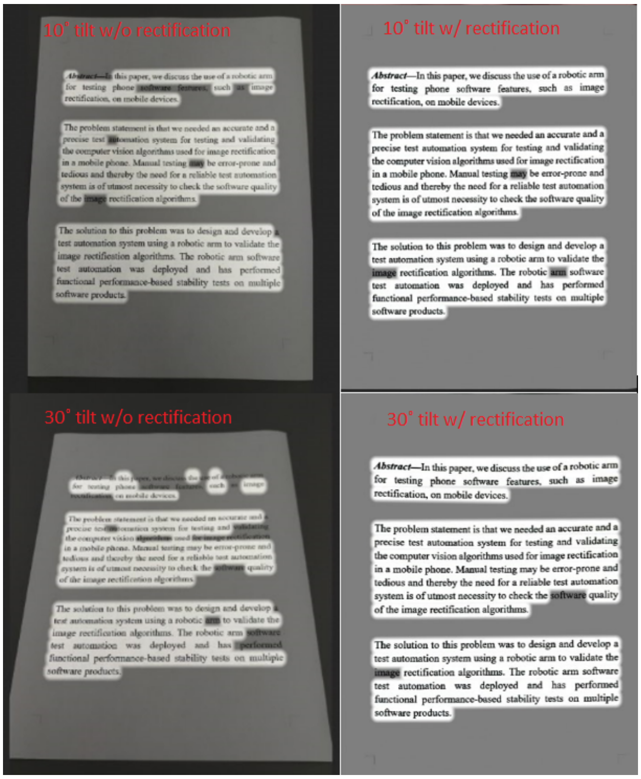

FIGURE 17. Image snapshots for clear document tilting scenarios.

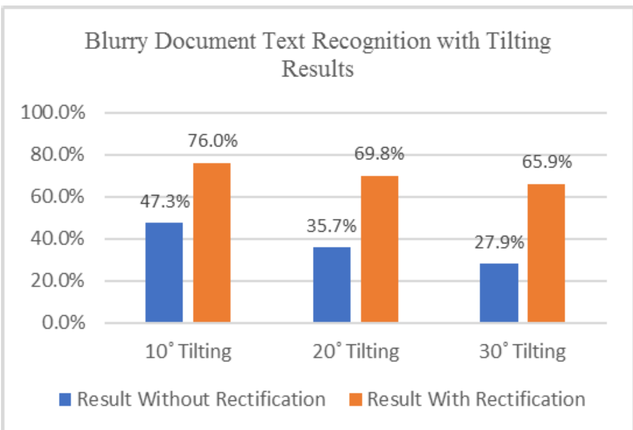

FIGURE 18. Comparison of text recognition rate with and without image rectification on blurry document.

B. RESULTS FOR TILTING USE CASE OF IMAGE RECTIFICATION

The image rectification test was conducted 10 times on tilting use case using the robotic arm tool. Images were captured on clear and blurry documents for 10°, 20°, 30° tilting angles. The result images were processed through text recognition software. The average text recognition rates are listed in table 3.

The results for clear document indicated the tilting scenario did not affect text recognition performance as much as rotations did. Within 20° tilting the text recognition rate without image rectification were around 94%. The performance with

TABLE 4. Text recognition results with moiré effect for clear and blurry documents.

Moiré Effect			
Scenarios	w/o Rectification	w/ Rectification	Delta
Clear Document	96.1%	98.4%	2.3%
Blurry Document	51.9%	66.7%	14.8 %

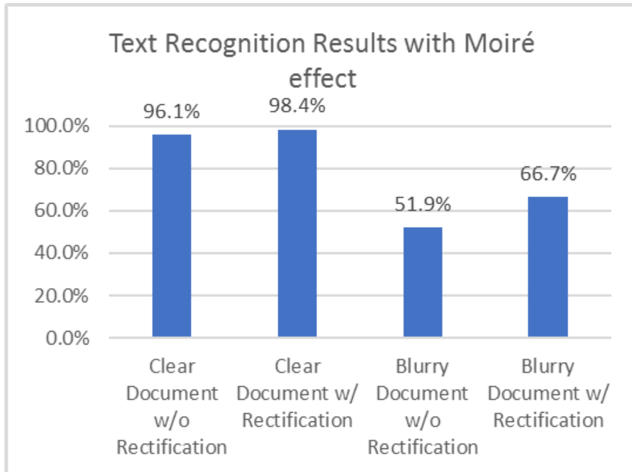

FIGURE 19. Comparison of text recognition rate with and without image rectification with moiré effect.

image rectification was slightly better with average recognition rate around 98% and about 4% delta comparing to the results without rectification. At 30° tilting angle, the recognition rate without rectification started to drop to 83.7%, a 10% decrease comparing to 20° tilting angle. On the other hand, with image rectification the recognition result remained unaffected, which made the text recognition result 14% higher than without image rectification. The results graph is shown in Fig. 16.

The reason why at lower tilting angles text recognition performance was not affect much was due to the texts were still horizontally aligned, they could be properly detected and processed by the text recognition software. At higher tilting angles, the texts on top of the images became further away and smaller, which made it harder to read and recognize, demonstrated in Fig. 17.

With image rectification, the document image was extracted and reformed to a more readable version. The texts on the rectified image were in uniform size, even though there were slight curvature on the text lines in the first paragraph from the rectified document image. This proved the advantage of using image rectification in tilting image text recognition scenarios.

In the blurry document case, the text recognition performance was impacted with and without image rectification as expected. However, the text recognition rate on unrectified images dropped on average of 53.7% while rectified images

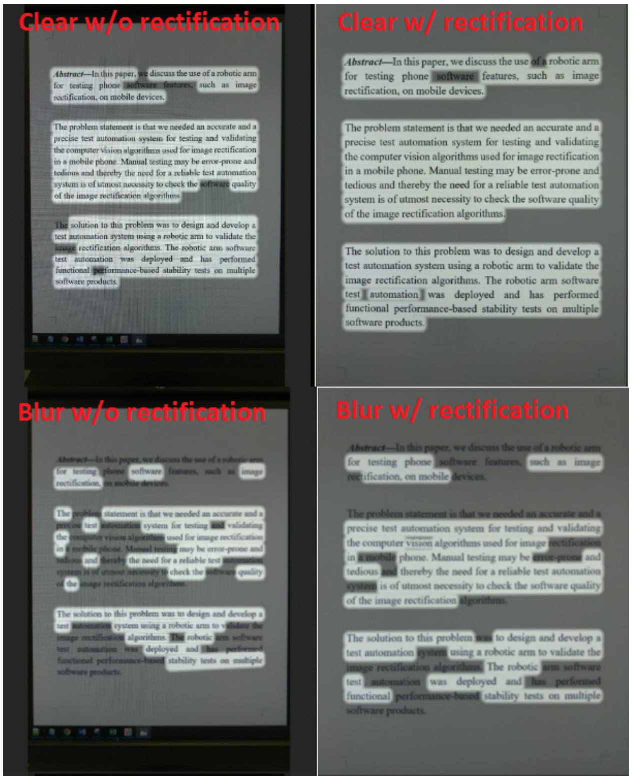

FIGURE 20. Image snapshots of tilting scenarios.

dropped on average of 27.4%. These data collected by image rectification test automation clearly indicated the positive effect of image rectification in the challenging scenarios. Comparison graph is shown in Fig. 18.

C. RESULTS FOR MOIRÉ EFFECT REMOVAL USE CASE OF IMAGE RECTIFICATION

In moiré effect use case image rectification test automation, instead of using printout documents we displayed the same documents on a monitor screen, and have the robotic arm focus the test device in front of the screen for inducing moiré effect. The automation was also ran 10 times on both clear and blurry documents, and the average results were listed on table 4.

The results indicated text recognition performed well on clear document with moiré effect inducing in our testing. However, on blurry document the text recognition performance was impacted. Both results from with and without image rectification were lower than the previous results in the same upfront position to the document without moiré effect. The comparison graph is shown in Fig. 19.

The delta of the results partially came from the low brightness of the monitor screen comparing to the brightness on the actual printout documents. Also, the moiré effect made the already blurry document texts even harder to recognize. However, from the result images shown in Fig. 20. The image rectification software did remove the moiré effect visible on the snapshot of the documents on the left. And the rectified image had 14.8% more text recognized in this scenario.

VI. CONCLUSION

The computer vision test automation has provided a framework for adding and managing tests using the robotic arm lab. The test automation efforts in power, fuzzing, performance, and stability have led us to discover several software issues and can lead to the release of a more robust software to our customers.

The tests were executed with clear and blurry images containing text with and without image rectification enabled. The result shows the increase in accuracy of text recognition with image rectification algorithm enabled.

We are working on expanding the test automation to both Smartphone and IOT (Internet of Things) platforms. We are adding new test content to our test suite to cover new features and solve customer-reported issues.

REFERENCES

[1] K. Mao, M. Harman, and Y. Jia, "Robotic testing of mobile apps for truly black-box automation," *IEEE Software*, vol. 34, no. 2, pp. 11–16, Mar./Apr. 2017.

[2] K. Mao, M. Harman, and Y. Jia, "Sapienz: Multi-objective automated testing for Android applications," in *Proc. 25th Int. Symp. Softw. Test. Anal. (ISSTA)*, 2016, pp. 94–105.

[3] D. Banerjee, K. Yu, and G. Aggarwal, "Robotic arm based 3D reconstruction test automation," *IEEE Access*, vol. 6, pp. 7206–7213, Jan. 2018, doi: 10.1109/ACCESS.2018.2794301.

[4] V. Garousi, and M. Felderer, "Worlds apart: Industrial and academic focus areas in software testing," *IEEE Software*, vol. 34, no. 5, pp. 38–45, May 2017.

[5] M. Harman, "Search based software testing for Android," in *Proc. IEEE/ACM 10th Int. Workshop Search-Based Softw. Test. (SBST)*, May 2017, p. 2.

[6] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robot. Auton. Syst.*, vol. 94, pp. 43–52, Aug. 2017.

[7] C. Yu and J. Xi, "Simultaneous and on-line calibration of a robot-based inspecting system," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 349–360, Feb. 2018.

[8] M. Jasiński, J. Mączak, P. Szulim, and S. Radkowski, "Autonomous agricultural robot—Testing of the vision system for plants/weed classification," in *Proc. Conf. Automat.*, vol. 743. Warsaw, Poland, 2018, pp. 473–482.

[9] J. Brookes, *et al.*, "Robots testing robots: ALAN-Arm, a humanoid arm for the testing of robotic rehabilitation systems," in *Proc. Int. Conf. Rehabil. Robot. (ICORR)*, Jul. 2017, pp. 676–681.

[10] N. Cramer *et al.*, "Design and testing of FERVOR: FlexiblE and reconfigurable voxel-based robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2730–2735.

[11] J.-H. Lim, S.-H. Song, J.-R. Son, T.-Y. Kuc, H.-S. Park, and H.-S. Kim, "An automated test method for robot platform and its components," *Int. J. Softw. Eng. Appl.*, vol. 4, no. 3, pp. 9–18, Jul. 2010.

[12] M. Mossige, A. Gotlieb, and H. Meling, "Testing robotized paint system using constraint programming: An industrial case study," in *Proc. 26th IFIP Int. Conf. Test. Softw. Syst.*, 2014, pp. 145–160.

[13] M. Mossige, A. Gotlieb, and H. Meling, "Test generation for robotized paint systems using constraint programming in a continuous integration environment," in *Proc. IEEE 6th Int. Conf. Softw. Test., Verification Validation (ICST)*, Mar. 2013, pp. 489–490.

[14] L. Jian-Ping, L. Juan-Juan, and W. Dong-Long, "Application analysis of automated testing framework based on robot," in *Proc. 3rd Int. Conf. Netw. Distrib. Comput.*, Oct. 2012, pp. 194–197.

[15] D. Araiza-Illan, A. G. Pipe, and K. Eder, "Intelligent agent-based stimulation for testing robotic software in human-robot interactions," in *Proc. 3rd Workshop Model-Driven Robot Softw. Eng.*, Leipzig, Germany, 2016, pp. 9–16.

[16] S. Stresnjak and Z. Hocenski, "Usage of robot framework in automation of functional test regression," in *Proc. 6th Int. Conf. Softw. Eng. Adv. (ICSEA)*, Barcelona, Spain, Oct. 2011, pp. 30–34.

DEBDEEP BANERJEE received the master's degree in electrical engineering from the Illinois Institute of Technology. He has over 10 years of industry experience in the field of software/systems engineering. He is currently a Senior Staff Engineer and an Engineering Manager with Qualcomm Technologies, Inc., USA. He is also the Software/Systems Development Engineer in test lead for the Computer Vision project. He is responsible for the test automation design, planning, development, deployment, code reviews and managing the project. He works closely with the software/system teams. He has been working on the software test automation team since the inception of the computer vision project with Qualcomm Technologies, Inc. He is involved in managing and developing software for the computer vision lab using the robotic arm.

KEVIN YU is currently a Support Test Engineer with Qualcomm Technologies, Inc., USA, and has contributed for test automation validation for continuous integration for validating the computer vision algorithms. He has also validated computer vision engine features, such as image rectification for the Android software products.

GARIMA AGGARWAL is currently a Test Engineer with Qualcomm Technologies, Inc., USA, and has actively involved in MATLAB post-processing modules for CV features and various other automation projects.