Robot Button Pressing In Human Environments

Fan Wang[1], Gerry Chen[1], Kris Hauser[1]

Abstract— **Service robots will need to actuate buttons and switches designed for humans in order to conduct many desirable functions. This paper presents the design of a robot named SwitchIt that is small, relatively inexpensive, easily mounted on a mobile robot, and actuates buttons reliably. Its operating characteristics were developed after conducting a systematic study of buttons and switches in human environments. From this study, we develop a categorization of buttons based on a sets of physical properties relevant for robots to operate them. After a human calibrates and annotates buttons in the robot's environment using a hand-held tablet, the system automatically recognizes, pushes, and detects the state of a variety of buttons. Empirical tests demonstrate that the system succeeds in operating 95.7% of 234 total buttons/switches in an office building and a household environment.**

I. INTRODUCTION

Recent years have seen a growing interest in service robots that assist humans in their daily lives, such as in factories, households, offices, and hospitals. Interacting with physical switches and buttons is a pervasive part of human life, used to operate lighting, appliances, computers, elevators, and machinery, and will therefore be an important capability for these robots. As a result, these control devices have been designed for simple, intuitive, and reliable operation by humans, both in terms of their ergonomic mechanical properties and distinctive physical appearance. There are many types of control devices in human environments, including push buttons, toggles, slides, and knobs, and in this paper these devices will hereafter be considered synonymous to a *button* or a *switch*. Manipulating a button/switch to perform a desired effect may also be variously referred to hereafter as *button pressing*, *switch operation*, or *switching*. Operating switches with a robot with human-level ease and reliability remains a challenging task, due to the fundamentally different sensing and actuation modalities on robots vs. humans.

This paper presents the system development and design of a 3DOF autonomous button pressing robot system. Called SwitchIt, our framework is a compact robotic switching solution based on relatively inexpensive hardware (Fig. 1). A relatively quick calibration setup is performed once for a given environment using a handheld tablet and small markers to identify the identity and purpose of each button. After calibration, our system recognizes and operates a wide variety of buttons automatically. For many buttons, it can also detect whether it has been successfully switched using sensor feedback.

*This work is partially supported by NSF CAREER grant #1253553.

F. Wang, G. Chen and K. Hauser are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA {fan.wang2, gerry.chen, kris.hauser}@duke.edu

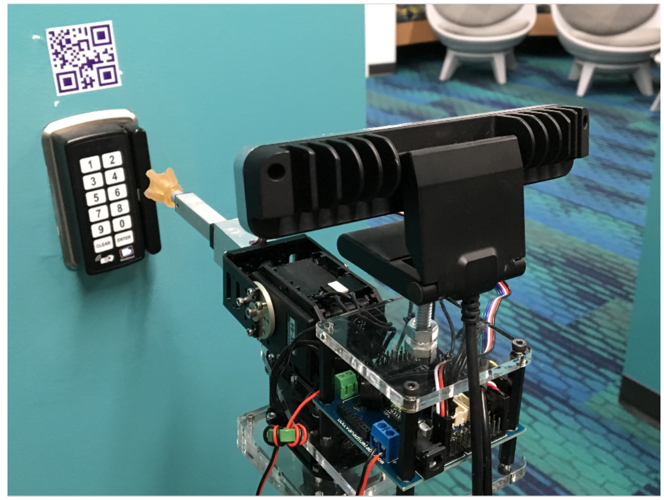


Fig. 1: SwitchIt is a spherical robot equipped with an RGB-D camera. Button panels are annotated using a QR code sticker affixed during a manual setup phase. Shown here mounted on a tripod, the robot is preparing to press buttons on an electronic passcode panel.

Our work presents three primary contributions:

1) A systematic categorization of buttons and switches into 6 classes based on their physical properties required for robotic actuation. We categorize and measure over 600 buttons found in offices and homes, and propose a taxonomy of buttons from a robot's operational point of view. We characterize several relevant physical properties of these buttons, including travel, size, shape, and operating force.
2) An annotation, calibration, and perception subsystem that achieves high-reliability button recognition, localization, and state detection. The calibration process also handles reflective and dark surfaces.
3) A compliant, scalloped end effector tip that can actuate pull buttons and turn knobs, and is robust to positioning error.

In controlled testing, our perception system localizes buttons with < 2 mm error and detects the state of toggle switches with 100% accuracy. For typical localization errors, the scalloped tip design achieves 99% repeatability compared with 89% for a cylindrical tip. We also test the system in an uncontrolled office and home environment, with 234 switches attempted in total. The platform succeeds at operating 224 (95.7%) total switches. In particular it was highly reliable at operating push buttons, sliders, rockers, and switches.

II. RELATED WORK

Computer vision techniques have long been employed for service robot to navigate and interact with objects in human environments [1], [2]. Specifically, several authors have addressed button identification issues that uses visual features in RGB images to identify and locates buttons. Identifying features could either come from a priori knowledge of the type of button [3], [4] or based on results of machine learning [5]. Most prior research using features based on prior knowledge is performed on elevators buttons [3]–[5].The advantage of this approach both seen and unseen buttons can be detected and to some degree understood automatically. However, for understanding these algorithms usually must rely on contextual cues such as grid layout and sequential arrangement of the buttons. Machine learning strategies are also commonly used [6], [7]. Sukhoy and Stoychev (2010) use an active machine learning strategy to identify and trigger a button autonomously. This method was used to train a robot to identify the active part of a door bell, and to trigger it effectively [8]. The auditory feedback of the feedback is used to determine whether the button was successfully pressed.

Our approach asks for a small amount of environment augmentation and manual labeling to identify each button definitively. Environment argumentation has been used in other robotics systems well to aid in object identifications. This achieves much higher accuracy than automated identification and require minimal setup time. Tools most commonly used are virtual reality tags such as RFID tags [9], QR code [10] and other artificial marks [11], [12]. Those tags usually provide information on the location of the objects, instructions on how to interact with this specific object and a task completion criteria. We use a similar approach with QR codes, which was also previously used to enable a mobile manipulator to plug itself in [10].

The work arguably closest to ours is Nguyen et al (2009). They uses a combination of an augmented environment and a variety of sensors to help a robot interact with its environment [9]. This work does allow the robot to operate certain switches. Force sensing and visual feedback, in the form of a change in lighting condition, is used to detect the change in button condition. In our work,we further demonstrate that high accuracy localization can be achieved using AR marker and RGB-D sensor alone, we also propose parameterized motion primitives for each class of buttons as opposed to the explicit defining instructions for each individual object to interact with.

III. CHARACTERIZATION OF SWITCHES IN HUMAN ENVIRONMENTS

A button or switch changes its internal electrical connection or signal based on the force applied to its external active mechanical component. For a human or robot operator, the underlying circuitry of a switch can be considered a black box and can be mostly ignored. The main focus is performing the appropriate physical action to correctly and safely trigger the switch.

Although some industrial settings employ switches that require significant force (or even tools) to be applied, here we focus on switches that are designed to be operated by one or two human fingers. These switches are designed with size, shape, material, and mechanical resistance that are comfortable for human finger to manipulate. Moreover, a switch usually has a distinctive appearance that indicates its mechanical functionality and semantic meaning, and usually provides feedback that can be promptly perceived and interpreted.

Although switches are ergonomically designed, the movement needed to trigger a switch safely and reliably is actually a delicate skill, acquired by humans through years of practice. Humans use memory, visual feedback, tactile feedback, and a variety of finger and hand contact strategies, and also progressively improve the efficiency and comfort of switching motions. For example, to operate stiff switches,a senior citizen with reduced hand strength will adopt finger postures that apply more leverage to stiff switches.

We collected data for over 600 switches in office and home environments,to help design our robots operational characteristics.This section describes their typical distribution and operational characteristics.

A. Button Taxonomy

Laypeople usually address buttons by a common name that references its function, such as light switch, toggle button, touch pad, dimmer, or keyboard. Electricians categorized them on the basis of their electrical connection such as single-pole single-throw (SPST) or double-pole single-throw (DPST), or by their triggering mechanism (sliders, push buttons) or the type of the application it is used on (light switches, dimmers). For a robot, perhaps the most useful categorization of switches is in terms of its physical triggering mechanism.

The operating mechanism of buttons and switches can be described with respect to a normal direction facing outward from a *button panel* (a plane behind which the electrical circuits are hidden). Our proposed taxonomy divides all household buttons into following 6 types based on their operating mechanism (Fig. 2).

- **Push-buttons** operate via application of a force that moves the operating part inward toward the panel. The movement of the button is linear.
- **Toggles**, such as in household light switches, generally have a rod-shaped protrusion (known as a *level*) that can be rotated about an axis to toggle between 2 distinct states. Some toggle switches have 3 or even more states. Internally, a spring and plunger mechanism is used to aid in operation, and equilibrium is only achieved in the extreme positions.
- **Sliders** require lateral movement, but the level can remain in equilibrium in any state in its travel range. Typically a slider is operated with a linear motion and stays in place using friction. However, some have discrete equilibrium states enforced by internal springs.

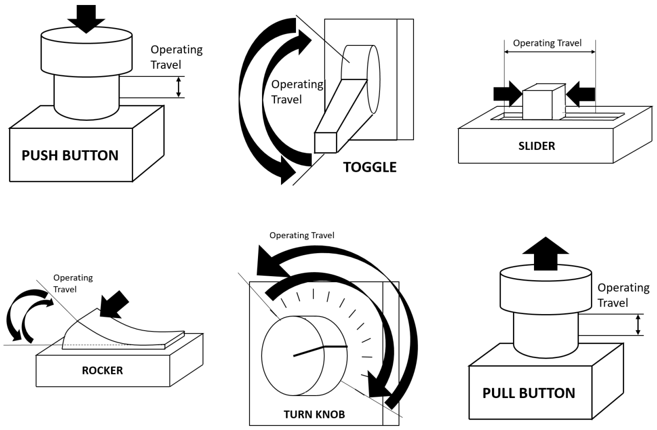


Fig. 2: Button characterized by type KH: again, is the travel distance TT or OT? Also the switch, slider, and rocker travel distance is incorrect on the figures.

FW: changed...

- **Rockers** are pressed on one of 2 ends like a seesaw and toggle between two discrete states. Unlike a toggle, a rocker is triggered by applying an inward force, primarily normal to the button panel.
- **Turn knobs** rotate along a center axis perpendicular to the panel face to adjust either a continuous or discrete value. Humans typically operate knobs using much of 2 or more fingers to achieve sufficient tangential friction about the axis.
- **Pull buttons** operate by a pulling action that moves the operating part away from the base to open or close the contacts. To operate the button, it must be solidly grasped either on the back or via friction on the sides.

B. Operation

Switches operate in three typical patterns: Momentary, Alternating, and Latching. In momentary operation, the switch is in an active state only while a force is being applied, and then once unloaded, a spring returns the switch to its original position. For alternating operation, the switch's state is held after it is released, and a different force must be applied to change state (this characterizes almost all toggles, pull buttons, and rockers). In latching operation, a second force in the same location/direction of the initial force returns the switch to its initial position, which is accomplished by some spring-loaded toggling mechanism. Latching characterizes several types of push-buttons.

C. Location and Geometry

The location of the button in the environment, direction of travel, and the travel distance affect whether the operational capabilities are within the workspace of the robot. We are primiarily concerned with height, but to ensure accessibility of a mobile base it may also be important for a robot designer to consider surrounding obstacles and clutter. The geometry of the button and its relationship to surrounding buttons is also an important aspect of for finger design, since it is

TABLE I: Breakdown of button characteristics by type

	Push	Toggle	Slider	Rocker	Turn	Pull
Prevalence	66%	25%	2%	2%	2%	4%
Force (N)	0.7-12.5	2.6-6.2	0.3-0.7	2.4-9	X	3-20
$\overline{\text{Force}}$ (N)	7.15	4.61	0.5	4.5	X	X
Trav. (cm)	0.1-0.35	1-2	0.5-11	0.2-0.45	X	X
$\overline{\text{Trav.}}$ (cm)	0.167	1.53	4.3	0.3	X	0.4
$\overline{\text{Sep.}}$ (cm)	1.6	8.5	X	X	X	X
$\overline{\text{Height}}$ (cm)	119	126	122	118	X	X

important to be able to press the button without accidentally activating nearby buttons.

In our survey, the vast majority of buttons on the walls or doors have height 1.06–1.44m above the ground. The exception is elevator buttons, which have height range 0.88–1.72m. (The highest button is designed to be difficult to reach; it is to be used only in case of emergency.)

D. Force

There are different types of forces associated with button pressing, but we primarily focused on Operating Force (OF) which is the peak force needed to change the state of the button. The typical force-stroke characteristic of a button displays no movement until a breakaway force is reached, after which the force increases with increasing displacement until the peak at the operating point. Afterwards, the force follows a sharp decrease, and then gradually increases again until reaching the total travel distance [13]. If tactile sensing is available, force profiles can be very helpful for detecting the success of button pressing. However, our system uses encoder derivatives to estimate the applied force.

Tab. I gives results of our survey, listing range and mean value of operating force (Force and $\overline{\text{Force}}$, respectively). Most buttons have OF in the 4.5–8.5N. 19.7% of buttons have OF $>$ 8.5N and 20.8% have OF $<$4.5N. Only two buttons exceeded OF $>$ 12N. For example, an old-fashioned elevator pull button required 22N to operate. However, we find these buttons are not designed to be used on a daily basis.

It should be noted that all buttons we surveyed have been in operation for at least one year. New buttons are usually much harder to activate and require some usage before their internal springs soften.

E. Surface material

The body of electronic switches are usually made with metal or plastic. While highly polished plastic and metal makes attractive appearances, reflective surfaces challenging for depth estimation with sensors. Our survey shows that 39% of buttons are made with highly polished plastic or metal and 10% are made with black material.

F. Travel distance

There are two parameters regarding travel distance: the total travel (TT) and operating travel (OT). TT indicates distance to a hard stop, while OT indicates distance until the switch is triggered. We are primarily interested in determining OT, although TT may be useful for tactile feedback. Most switch specifications suggested TT-OT to be between 0.5mm-1.8mm [14], [15].

In our survey, OT varies significantly, particularly among sliders (Tab. I). However, we are most interested with the travel of push buttons, since they provide the least reliable visual cue of button state. From this data, we determined that an open-loop position controlled robot could press down a maximum of 5mm before linear actuator motor stop is detected to maximize its chance of activating a button successfully, while also being unlikely to cause damage to the button.

IV. THE SWITCHIT PLATFORM

With the above data in mind, the SwitchIt robot accessory is designed to operate a large number of switches and to be easily mounted on a mobile robot platform. Setting up the system for use in a new environment requires a human to first perform a calibration procedure, which involves affixing QR codes to button panels and annotating reference models of the panels using a hand-held tablet. Afterward, the system will autonomously recognize any visible panel, suggest a reference position for the robot's base, and once in position, press a requested button or sequence of buttons.

A. *Hardware*

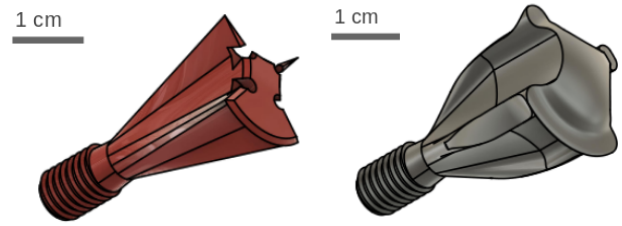
The robot arm used in our system is a custom 3DOF spherical robot that can pan, tilt and extend. The pan-tilt DOFs are ScorpionX MX-64 Robot Turret kit item number KIT-SXT-MX64 and extension is provided with a 50mm Firgelli Linear actuator. We have tested the physical capability of the arm in its workspace and measured a output force exerted by the tip in a range from 6N to 18N, although the robot is generally weaker and more susceptible to flexing the further it moves away from the center of the work space, it should nonetheless be strong enough to trigger most buttons, which have operating force $<$ 12N. The positioning accuracy of the robot after calibration is measured to be sub-millimeter on average, and less than 3mm maximum.

For sensing we use a single Intel RealSense F200 RGB-D camera to do colour and depth capture. These cameras are inexpensive (current list price $129), have a depth range of 20–120 cm, and work optimally in well-lit, indoor environments and diffuse objects.

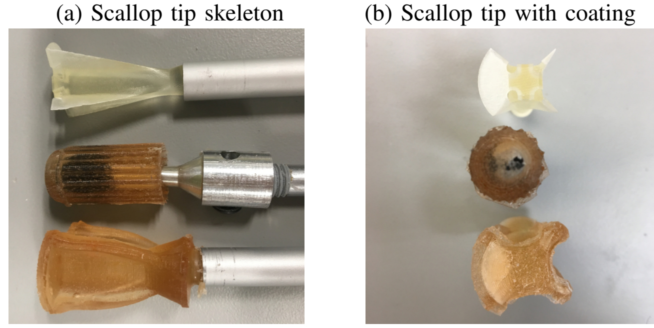
Both the robot and camera are mounted on a fixed base using Plexiglass and a camera mount. We assume that the mount is attached to a mobile robot or an arm that has sufficient rigidity to keep the unit roughly in place while it presses buttons.

B. *Scallop fingertip*

We considered using a cylindrical rubber-coated tip, with similar shape and size to a human-finger. However, a novel scallop design proved to manipulate buttons much more reliably. The principle of the design is to increase tolerance against positioning error, and to enable motion primitives



(a) Scallop tip skeleton   (b) Scallop tip with coating



(c) Side view of tips. From top to bottom: scallop skeleton, cylindrical, scallop coated with rubber

(d) Top view of tips. From top to bottom: scallop skeleton, cylindrical, scallop coated with rubber

Fig. 3: Cylindrical and scallop tip designs

of pulling and turning which are usually difficult to achieve with a 3-DOF robot.

The design contains four rigid scallops protruding from its lateral edges (Fig. 3). For switches with activating rods, a scalloped channel guides motion towards the center-line, which corrects for positioning error and increases effective lateral friction. An underlying skeleton is 3D printed from rigid plastic, and this is coated with a 1-3mm thick PMC-121/30 rubber compound. The rubber coating provides compliance and large friction that helps correct for positioning errors and reduce slippage. The coating at the forward tip of the finger is curved with a dimension and shape similar to a human's index finger. The scallop protrusions are also coated with longer rubber "skirts" to establish larger contact area when turning knobs. Finally, the rigid skeleton is also designed with a narrow "hook" located on one side of the finger, which is designed for holding onto pull buttons.

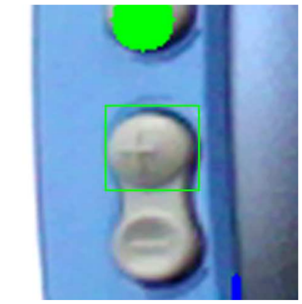
C. *Environmental annotation and calibration*

To apply the method to a new environment, a manual setup procedure must be performed to populate a database of known button panels. The process is relatively fast and the environment is minimally altered. For example, to complete annotation, calibration, and information entry for a medical device panel with 10 buttons takes less than 4 minutes (Fig. 4). The procedure consists of the following steps:

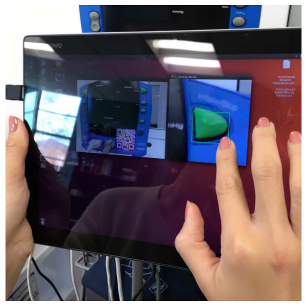
- The user affixes a QR code on or near the button panel.
- Using a tablet with attached RGB-D camera, the user takes a picture of the panel and provides an identifier for the panel.
- Guided by the annotation GUI, the user adds each button by name, type, and designates areas of interest

(a) Taking a picture of the panel panel with RGB-D camera

(b) Tap on a button to zoom in. Already calibrated buttons will be marked



(c) Draw rectangles in the area of interest as guided for each type of buttons



(d) Zoomed in button details make it easier for operators to calibrate with higher accuracy

Fig. 4: Steps of the panel calibration process

on the picture.
- The panel identifier, QR code, RGB-D information, button names, type, location, size, and areas of interest are saved to a database.

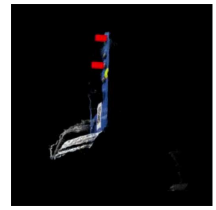
It should be noted that dark or highly reflective panels or buttons cause problems with depth estimation, which could lead to erroneous 3D button location estimates. To accommodate these types of materials, the user should temporarily apply matte tape (masking or painter's tape) to the button panel when capturing the reference RGB-D image. After calibration the tape may be removed.

One omission of the current procedure is that we do not store a 3D map of button panel locations. As a result, to use our system, a mobile base must be able to first position the camera to observe the panel's QR code. Future iterations of our system might record panel location, and incorporate simultaneous localization and mapping (SLAM) software to guide a mobile base to a desired panel.

D. Button Panel Recognition and Localization

Recognition and localization consists of an imprecise QR code localization followed by a more accurate point cloud registration via Iterative Closest Points (ICP) algorithm [16], [17] . When a QR code is detected, the panel reference RGB-D image and all button annotations become available. A first guess is obtained from the QR code, which gives an estimated affine transformation between the reference image coordinates and the current camera coordinates. Since QR codes are relatively small, this estimate is often inaccurate.

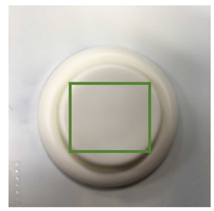
(a) Front view



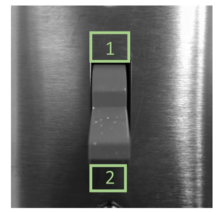
(b) Side view

(c) RGB front view

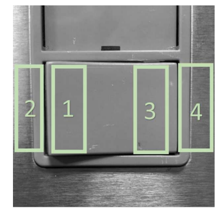
Fig. 5: Point cloud and RGB view showing localization of button surface in real-time. Computed locations of the buttons are drawn as red rectangles.



(a) Push button

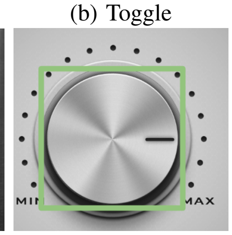


(b) Toggle

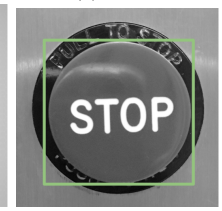


(c) Rocker

(d) Slider



(e) Turn knob



(f) Pull button

Fig. 6: Calibration areas by button type

To improve accuracy, we then apply the ICP to match the point cloud corresponding to the reference RGB-D image to the currently observed point cloud. The QR code localization gives a reasonable initial guess for this optimization. It should be noted when depth data is missing or corrupted by dark or highly reflective surfaces, ICP is not as effective, and the system relies more on the QR code and surrounding non-reflective surfaces.

E. Button State Recognition

Many buttons provide tactile and/or auditory feedback primarily intended for humans, but which can be used by the robot to determine whether it has successfully been switched. For robot not equipped with microphone or force sensor, it can be quite difficult to determine the state of the button and to confirm the completion of the task. Some of the previous research uses heuristics such dimming in the lighting condition to determine if light switch has been successfully switched off [9]. However, those heuristics are not always available or can be relied on.

SwitchIt uses RGB-D information to detect the state of toggle, slider, and rocker switches, as well as push-buttons with back-lit LEDs. The shape of different switch positions can be quite distinctive. The lever rod of a toggle switch rests on 2 opposite sides at different states, and rocker surfaces tilt up at different sides. Take a light switch as an example. If the switch is in the up position, the average distance from the button surface to the underlying plane will be greater on the upper side of the button.

Using a region of interest from the calibration data, our method calculates the average displacement from the panel plane in both halves of the switch, and detects the switch position by the maximum displacement. We filter out noise at 5 cm distance from the panel since this is most likely caused by obstacles in front of the camera, e.g., the robot itself. We use the same method for detecting states of rocker buttons and sliders.

State detection is challenging for push-buttons, since many do not change in appearance and shape after activation. Some push buttons do provide a visual cues, for example, an LED back-light. We therefore focused on detecting these differences in the RGB image. However, we find that very few push buttons provide visual cues, and the interpretation is not always consistent (e.g. a backlight turns on vs the button itself lights up).

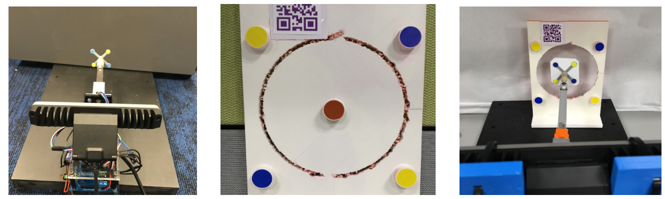
F. Control

The controller of the robot is initiated when the robot is in reach of the button panel, and a button pressing sequence has been specified. The robot performs an end effector motion governed by the button type and marked areas of interest collected during calibration. The pushing strategy for each type of button is as follows:

- Push: Approach the center of the marked zone and push down. Pushing stops if one of the following conditions have been met: 1) the linear actuator has fully extended by 5mm, or 2) encoder readings indicate that the linear actuator has been stopped for 0.2s.
- Toggle: Linearly interpolate between center of 2 zones, in the direction needed to switch off / on.
- Rocker: Linearly interpolate from zones 1 to 2 or from 3 to 4, depending on the operational state. (We found this diagonal movement to slip less frequently than pushing straight downward.)
- Slider: Same as the toggle, except that the user / supervisor can specify a fractional travel amount.
- Turn Knob: Button center locations, button radius, and knob depth are determined from the marked region. During actuation, the tip touches the side of the knob and moves in a circular motion in the direction specified.
- Pull Button: First, approach the side of the button with the "hook" pointing inward and then move inward by 5mm. Interpolate toward a point 5mm in front of the pull button surface center, or until the encoder reading indicates that the linear actuator motor has been stopped for 0.2s.

V. EXPERIMENTS

We have done an full accuracy measurement of our system and separate localization errors with and without ICP. We also test our system with test panels that contains buttons of various types, shapes and stiffness, and finally an exhaustive real-world test in office and home environments.



(a) Robot and camera are calibrated using a cross marker

(b) Center button is located at the intersection of opposing diagonal circles

(c) Center plate is removed during the test to not interfere with tip motion

Fig. 7: Accuracy test apparatus

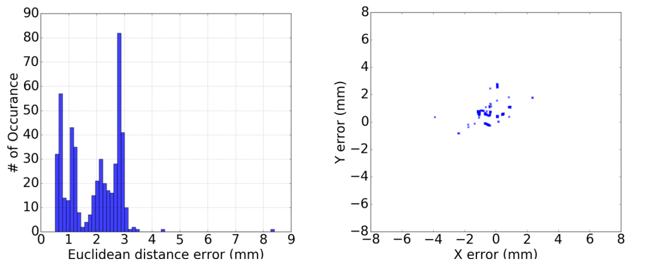
A. Calibration

The robot and camera are automatically calibrated using a colored cross shape fiducial fixed to the end of the robot with a pin. Colored blob detection and averaging produces a relatively accurate measurement of the 3D tip position in the camera frame. The robot is driven to 30 random locations within its workspace and its joint coordinates are recorded. When the four circles are visible in both color and depth images and agree on the tip position within 5 mm, we consider the tip to be accurately measured. A least-squares transformation matrix between joint coordinates and sensed positions is then fit to the data. After calibration we find the mean average error in the range of 0.8mm-1mm with the maximum error is less than 3mm.

B. Measurement of system accuracy

We built an apparatus to measure the cumulative positioning error of the system including human set-up error, calibration error, localization error, and hardware inaccuracies. We built a button panel with a "virtual button" in the center, whose coordinates are at the center of two colored diagonal visual features (Fig. 7). We conducted 500 test pushes using panel localization to determine the button location. Between each push we changed the position and orientation of the panel, with the entire panel oriented on each of its four sides and with yaw altered to up to $45°$. The tip position was measured using the calibration cross marker, and "true" button center position was measured using the larger cross features. Results show the average euclidean distance from the tip center to the sensed button center is 1.9mm, with an outlier of 8.39mm (Fig. 8). This outlier was likely caused by a fault on the linear actuator.

To measure localization error we performed 500 localization readings on a different test panel that has a slider,rocker,push button and a switch on KH: is this the same panel as before, or something else? FW: it's a different one... KH: How can you get ground truth readings then??? FW: I put a yellow circle sticker on the push button and setup that button... with and without using ICP and compared the depth readings with the "ground truth" from color tracking. Repeatability for a static button panel (i.e., the effect of camera noise) suggests the maximum euclidean distance error using only QR for localization was 2.56mm, which is reduced to 0.5mm with ICP. We performed the same



(a) Histogram of system errors (b) Distribution on the XY plane

Fig. 8: System accuracy measurement using the apparatus in Fig. 7

experiments while shifting and rotating the panel for each reading, and obtained maximum error 2.76mm without ICP and 1.3mm with ICP.

C. Test panel experiments

As a controlled test of our platform's reliability in operating switches, we built 2 test panels, one with toggle switches and another with push buttons. The push button panel consists of 5mm radius buttons made with polished plastic. Each pair of buttons is separated by at least 2mm. The toggle test panel holds 5 toggles of different size, shape, and operating force ranging from 2.5N to 8.2N. All buttons are located within a 12cm by 10cm rectangle centered on a 30cm by 23cm flat panel, placed approximately 21cm away from the robot's base.

In the switch test, the robot localizes and switches on the 5 switches in sequence and then reports the perceived state of the switches. Then, it switches them off and again reports their perceived state. In case of a failure, we manually flip the switch to the desired state for the next sequence. For the pushbutton test, robot localizes and pushes 10 buttons in sequence. In each test, the robot runs 10 sequences and attempts 100 actuations in total. The panel is illuminated with indoor office lighting. To judge the impact of different components on performance, we performed these tests with the two tips (cylindrical and scallop) and with and without ICP activated.

Tab. II shows that although the cylindrical tip performed well at button pushing, it failed in roughly one fifth of the switch attempts. We observed that errors occurred due to slippage or flexing of the structure. The scallop tip eliminated slippage, although it still failed 4 times when activating the lower left button with high 8.2N OF. The failure case was further reduced to only 1 when ICP was used in localization. The scallop tip design still has a high success rate on push buttons, even though it has a larger cross section than the small buttons we tested on. This is due to the curved tip that can direct force within a small area.

D. Experiment pressing in an office building and home

We tested our system on all accessible buttons on doors/walls and large electrical appliances in the a 4-story office building and a 2 story town house. We mounted our robot on a height adjustable stand with lockable wheels and pushed the robot along the corridor while testing on all buttons that are accessible and safe to test on.

(a) Door passcode (b) Disabled door exit (c) Thermostat buttons (d) Switch with dimmer

(e) Security door entrance (f) Electronic keypad (g) Light switch (h) Office stereo control

Fig. 9: A selection of button panels our platform succeeded in activating

(a) An old-style timer (b) An elevator pull button (c) A washer control button (d) A turn knob on oven

Fig. 10: Four button types that SwitchIt failed to activate.

We have tested on 98 different button panels that contain a total of 379 individual buttons. The test set covers 39 distinct classes of panels and all 6 button types. We tested all distinct button on every panel, but when panels contain many identical buttons, we did not test every button. Specifically, if a panel contains exact duplicates of one button (such as a numeric keypad), we only tested 2 or 3 of them at extreme positions. In total, we asked the robot to operate 234 buttons.

Our robot succeeded in activating 224 of 234 buttons. We note that several of these buttons were quite challenging (Fig. 9). Successes include office passcode entry with small, stiff, and slippery metallic buttons; small and stiff rockers; turn knobs on a classroom stereo control; and non-conventional light switches that are activated from the side.

Out of the 10 failures, 3 switches are within the capability of the device, but failed due to various positioning errors. The other 7 were of 4 button designs our system currently cannot actuate (Fig. 10). They include "push to stop, pull to run" emergency stop button with a shallow smooth indentation for human finger to pinch. The hook in our finger cannot establish a solid hold onto this indentation. The other 3 are variations of turn knobs such as an old-style timer that requires more than 20N to turn, 3 oven temperature knobs that must pushed in while turning, and 2 washing machine controls that are "pull to start, push to stop, and turn to select." More work is needed to develop actuation strategies for these exotic button types.

TABLE II: Test Panel Experiments

Button Type	Toggle	Toggle	Toggle	Toggle	Push	Push	Push
Tip	Cylinder	Scallop	Cylinder	Scallop	Cylinder	Scallop	Scallop
ICP	Off	Off	On	On	Off	Off	On
Success rate	79%	96%	84%	99%	98%	99%	100%
State detection	100%	100%	100%	100%	n/a	n/a	n/a
Duration (min)	20	20	22	22	13	13	16
Time / push (s)	12	12	13.2	13.2	7.8	7.8	9.6

VI. CONCLUSION

This paper presented a reliable robot system for button pressing in human environments. The SwitchIt platform is designed for accurate button identification and localization, and uses a tip designed to be robust to position errors. In our tests, 95.7% of buttons in an office and a home were successfully actuated. All push buttons, switches, sliders, rockers, and some turn knobs were successfully operated. Three of the failed switches were within the capabilities of our system, but failed due to positioning error. The other seven failure cases occurred in one pull button and turn knobs with unusual actuation characteristics. Nevertheless, these results are encouraging for the prospect of service robots that can navigate elevators, operate appliances, and control legacy lighting and temperature control systems.

Future work should study robustness to different lighting conditions, such as dark rooms and outdoor lighting. We would also like to improve the capabilities of our system on buttons with exotic actuation strategies. We also would be interested in reducing the manual annotation and calibration effort needed to set up a new environment. Object recognition and new sensing and control modalities, such as tactile exploration, may be needed to operate novel buttons.

REFERENCES

[1] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger, "Autonomous door opening and plugging in with a personal robot," in *IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 729–736.

[2] A. Pratkanis, A. E. Leeper, and K. Salisbury, "Replacing the office intern: An autonomous coffee run with a mobile manipulator," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 1248–1253.

[3] W.-J. Wang, C.-H. Huang, I.-H. Lai, and H.-C. Chen, "A robot arm for pushing elevator buttons," in *SICE Annual Conference 2010, Proceedings of*. IEEE, 2010, pp. 1844–1848.

[4] J.-G. Kang, S.-Y. An, and S.-Y. Oh, "Navigation strategy for the service robot in the elevator environment," in *Int. Conf. on Control, Automation and Systems*. IEEE, 2007, pp. 1092–1097.

[5] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Y. Ng, "Autonomous operation of novel elevators for robot navigation," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2010, pp. 751–758.

[6] M. Salganicoff, L. H. Ungar, and R. Bajcsy, "Active learning for vision-based robot grasping," *Machine Learning*, vol. 23, no. 2-3, pp. 251–278, 1996.

[7] A. Boularias, O. Kroemer, and J. Peters, "Learning robot grasping from 3-d images with markov random fields," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1548–1553.

[8] V. Sukhoy and A. Stoytchev, "Learning to detect the functional components of doorbell buttons using active exploration and multimodal correlation," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*. IEEE, 2010, pp. 572–579.

[9] H. Nguyen, T. Deyle, M. S. Reynolds, and C. C. Kemp, "Ppstags: Physical, perceptual and semantic tags for autonomous mobile manipulation." Georgia Institute of Technology, 2009.

[10] S. J. Lee, J. Lim, G. Tewolde, and J. Kwon, "Autonomous tour guide robot by using ultrasonic range sensors and qr code recognition in indoor environment," in *IEEE Int. Conf. on Electro/Information Technology (EIT)*. IEEE, 2014, pp. 410–415.

[11] R. Katsuki, J. Ota, T. Mizuta, T. Kito, T. Arai, T. Ueyama, and T. Nishiyama, "Design of an artificial mark to determine 3d pose by monocular vision," in *IEEE Int. Conf. on Robotics and Automation*, vol. 1. IEEE, 2003, pp. 995–1000.

[12] R. Katsuki, J. Ota, Y. Tamura, T. Mizuta, T. Kito, T. Arai, T. Ueyama, and T. Nishiyama, "Handling of objects with marks by a robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1. IEEE, 2003, pp. 130–135.

[13] OMRON. (2017) Switches: Basic operating characteristics. https://www.omron.com/ecb/products/sw/special/switch/basic01-08.html. (accessed March 2, 2017).

[14] C&K Switches. (2017) Product catalog. http://www.ckswitches.com/media/1416/ps.pdf. (accessed March 2, 2017).

[15] OMRON. (2012) Product catalog. http://www.ckswitches.com/media/1416/ps.pdf. (accessed September 9, 2012).

[16] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[17] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145 – 155, 1992.