



Leveraging stencils and capacitive touch devices to iterate in-cabin UIs

Scott Carter
scott.carter@tri.global
Toyota Research Institute
Los Altos, CA, USA

Jonathan Yao
jonathan.yao@tri.global
Toyota Research Institute
Los Altos, CA, USA

Monica Van
monica.van.ctr@tri.global
Toyota Research Institute
Los Altos, CA, USA

Alexandre Filipowicz
alex.filipowicz@tri.global
Toyota Research Institute
Los Altos, CA, USA

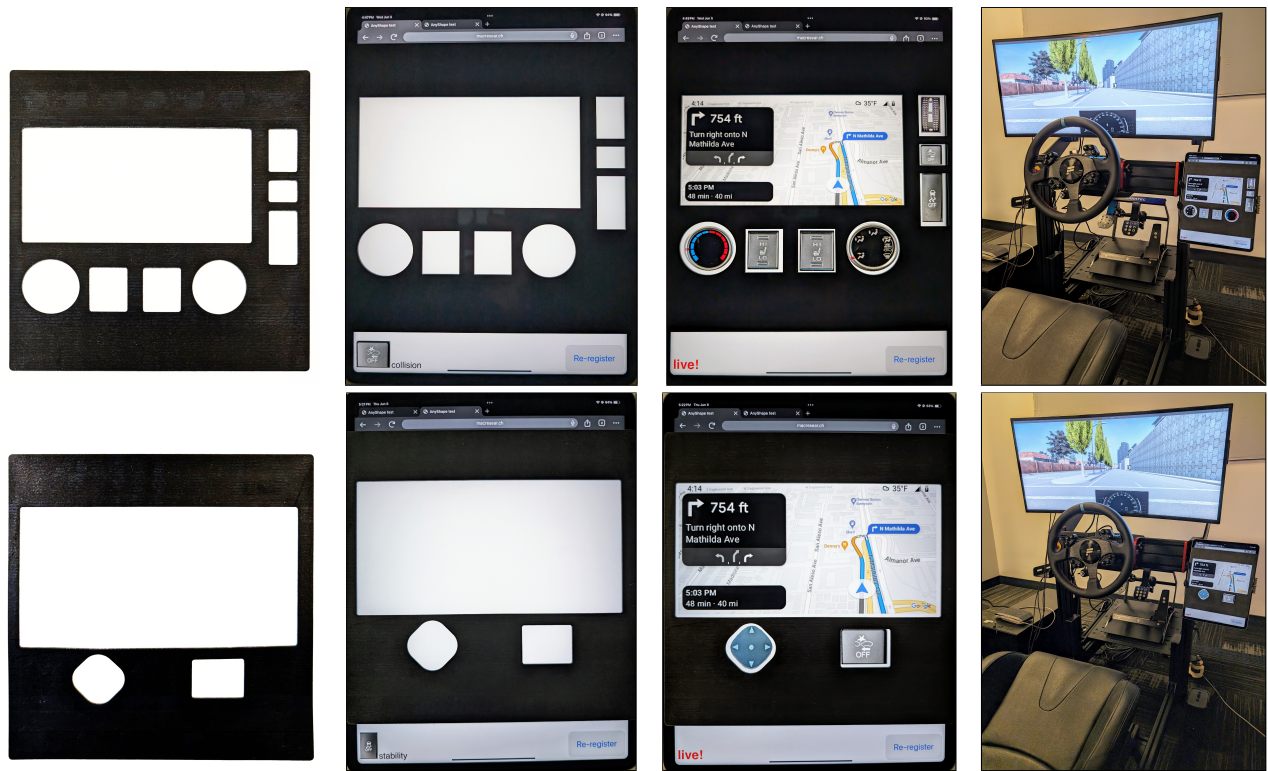


Figure 1: Using our system to deploy interactors on two different stencils with different interactor sizes and locations. The lower stencil reuses two interactors from the top one (the map and the collision button). Stencils (left) are affixed to a device (center-left) with low-tack glue (if the tackiness of the stencil material is not sufficient). The user registers each interactor to their stencil location and size (center-right). When registration is complete, the display is ready for deployment to an in-cabin testing environment (right).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

AutomotiveUI Adjunct '24, September 22–25, 2024, Stanford, CA, USA

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0520-5/24/09

<https://doi.org/10.1145/3641308.3685012>

ABSTRACT

We present a demonstration system that combines off-the-shelf capacitive screens with easy-to-produce stencils to facilitate the rapid iteration of tactile in-cabin user interfaces. Designers can use this tool to experiment with the position and layout of interactive components; they can 3D print or manually cut stencils out of common material and mount them to the display surface with low-tack glue to create a tactile experience that simulates tactile controls.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools.**

KEYWORDS

rapid prototyping, vehicles, touch interaction, haptic interaction

ACM Reference Format:

Scott Carter, Monica Van, Jonathan Yao, and Alexandre Filipowicz. 2024. Leveraging stencils and capacitive touch devices to iterate in-cabin UIs. In *16th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI Adjunct '24), September 22–25, 2024, Stanford, CA, USA*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3641308.3685012>

1 INTRODUCTION

Capacitive touch surfaces are now ubiquitous. Most people carry at least one touch surface at all times (e.g., smartphones), and these are now found in cars, laptops and watches, and on all manner of personal and public tablets and public displays. While this is largely an improvement over what came before, the elimination of so many haptic controls is unfortunate. Humans have a fine-tuned mechanoreceptive sense. Eliminating tactile controllers means that most devices now command visual attention to be used, making them less useful for a wide swath of activities where vision could be used to support another task. However, tactile controllers take considerably longer to prototype and fabricate, an issue that has likely contributed to the proliferation of capacitive controllers.

Taking inspiration from past work showing that stencils can help reduce attentional demands in driving tasks [2], we present a software framework that takes advantage of the ubiquity of capacitive screens and combines them with easy-to-produce stencils to facilitate the rapid iteration of haptic in-cabin user interfaces. Our approach allows software developers to use a set of interactive components that can be redeployed to different sizes and locations on a display surface. Developers can 3D print or manually cut stencils out of common material and mount them to the display surface with low-tack glue to create a tactile experience that simulates tactile controls.

2 RELATED WORK

Although there are many systems designed to bring haptic control to bear on capacitive touch surfaces, they require significant manufacturing and preparation [5, 7–10]. Other related work focuses on a somewhat different problem, such as adding haptic controls to current UIs [4]. Static stencils have been used as a promising method to add haptic control to in-cabin touch surfaces [2]. However, this work leveraged stencils only in the context of a user study; there was no attempt to integrate them into a prototyping system or method.

Touchplates [6] also leveraged stencils to add tactile sense to touch displays, but the system was not designed for capacitive-touch technologies and thus cannot be used on the ubiquitous touch devices available today.

3 PROCESS

We propose a new approach that allows anyone with access to fabrication processes (e.g., waterjet cutter, 3D printer, die cutter, or even a fine cutting knife) to transform any capacitive touch surface into a tactile controller.

The process is straightforward and has two main configurations: one in which the main app and controller are collocated on the same display, and one in which the app and controller are on different devices (this may be the case when the interactor on the app takes up all of the displayable space on a device). The system is implemented as a web page using standard HTML/CSS/JS. A stencil can be created using the following steps:

- A user creates any contiguous 2D stencil shape (e.g., as a PNG file). They then either send that shape to a 3D printer or cutter or print it on standard paper to use as a guide to cut the shape into a material. The material can be tacky and non-conductive, capable of attaching temporarily to a capacitive touch surface. It is also possible to apply a substance (e.g., Krylon Easy-Tack Repositionable Adhesive) to any material that will render it tacky and repositionable.
- With the shape cut in relief, the material can be placed on the control capacitive touch surface.
- For each interactor, the user specifies in the app the interactor's shape file as well as a software callback function to handle events generated by the interactor. Users also supply a text label for each interactor.
- Next, the user connects the controller and app devices. If the app and controller are on the same device, this step is automatic. If the app and controller are on different devices, this step involves linking the devices by sharing an identifying code between them (i.e., scanning a barcode that encodes the identifying code or manually typing in the code).
- The app then instructs the user to trace the outlines of the inner part of the first interactor with their finger on the capacitive touch device.
- As the user draws, the app determines the bounding box of the current trace, automatically injects the interactor image, and sends an event to the registered callback function with the interactor's current dimensions. When the user is finished placing one interactor they can click a button to advance to the next one. This step is repeated for all interactors.
- At this point, the initialization process ends, and the user can begin interacting with the region. Our system sends a "live" event to all interactors so that they can change their representation appropriately. When the user touches an interactor, the system automatically adjusts the relative size and position of events relative to the bounding box of the shape and then dispatches them to the user-specified callback function. The callback function can update the interactor and send updated events to other applications, such as driving simulation systems.
- Furthermore, designers can add vibrations to events. This can help communicate differences in events. Using the previous example, a designer could configure the "on()" event to pulse 2 rapid vibrations in succession; an "off()" event can

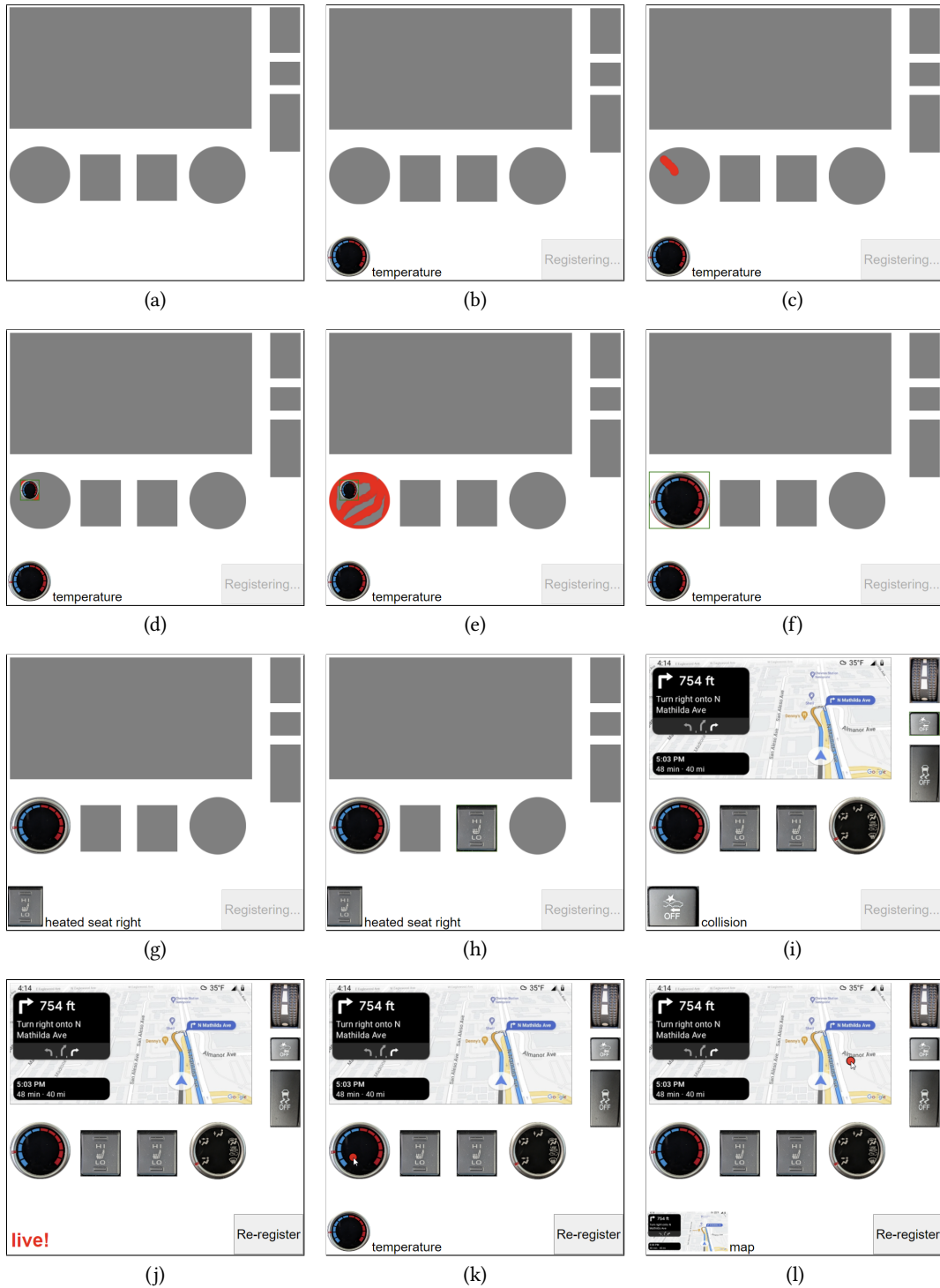


Figure 2: The ultimate target for the system is a real stencil and capacitive touch device (see Figure 1), but layout designs can also be tested on the web using a virtual stencil (a). Here, gray regions represent the "holes" in the stencil in which pre-registered interactor images are placed. During the registration process, each interactor image and its label is presented in sequence (b, lower left). The user scribbles inside the region where they want to place the image (c). The system places the image within the detected bounding box of the scribbled area (d). The user can update the marked area (e,f) and advance to the next image by clicking on the current image and label (g,h). When all images are placed (i,j), the system enters "live" mode in which touch events are detected and sent to their respective interactor. The currently active interactor can also appear in the lower left for verification (k,l).

trigger a single vibration. (This is accomplished with the `navigator.vibrate()` API available in most modern browsers.)

- As mentioned above, the control surface can be on a different device entirely. In this case, the two displays are linked using websockets. The app presents a QR code that resolves to a URL that incorporates a unique ID. When a user scans the code with the built-in camera on their device, the controller will navigate to the target URL and use the ID to establish a connection to the app. Once complete, the app and controller can send configuration information back-and-forth to complete the registration process.

The actual stencil itself has only one requirement: that the border be larger than the detectable width of touchpoints on the controlling device. This is necessary to indicate a tactile border.

4 EXTENSIONS

We have prepared a demo of the system described above and we are also investigating other extensions.

4.1 3D controllers using layering

We can take advantage of conductive fabrics to create 3D physical controllers. In this embodiment, stencils can be layered on top of a touchscreen to create depth and layer controls. For example, we can use two stencils to create a circular interactor that has both dial and button properties (e.g., an automotive in-cabin stereo system controller in which the dial controls volume and the button is an on/off switch). The lower stencil (the one affixed to the screen) would have a small circular cutout while the larger one (affixed to the lower stencil) would have a wider circular cutout. Conductive fabric [1] can be placed on the annulus formed on the lower stencil. User interactions with the fabric can be mapped to the dial (volume). Meanwhile, a button can be displayed on the touchscreen and mapped to the button interactor (on/off).

4.2 Reusing the registration process beyond capacitive touch devices

We can use the registration process described in the previous section to augment any controllable object connected to the controller or app interface. For example, conductive fabric can be draped over arbitrary 3D objects made from clay (or any malleable material). A board connected to the conductive fabric can itself be connected to the app or controller, allowing the user to assign any known 3D interactor to the area described by the fabric.

Finally, we can take advantage of reformable materials (such as FibeRobo [3] and thermoplastics) to improve reusability.

5 CONCLUSION

Past work has shown that using stencils to add "tactile feedback to touchscreen interaction...reduce[s] attentional demands" in driving scenarios [2]. Stencils have also been used to add tactile controls to touch screens for blind users [6]. Our work extends this past work, demonstrating a method to iterate touchscreen UIs that take advantage of this kind of tactile interaction. Our method allows designers to use off-the-shelf capacitive devices and easily manufactured stencils. Furthermore, software interactors can be repositioned and

resized using a simple registration process. With this approach, designers can rapidly experiment with different in-cabin UI layouts while simulating some aspects of tactile controllers.

ACKNOWLEDGMENTS

Thanks to Sheryl Chau for her help.

REFERENCES

- [1] Adafruit. 2024. Woven Conductive Fabric. <https://www.adafruit.com/product/1168>. Accessed: 2024-06-04.
- [2] Andy Cockburn, Dion Woolley, Kien Tran Pham Thai, Don Clucas, Simon Hoermann, and Carl Gutwin. 2018. Reducing the attentional demands of in-vehicle touchscreens with stencil overlays. In *Proceedings of the 10th international conference on automotive user interfaces and interactive vehicular applications*. 33–42.
- [3] Jack Forman, Ozgun Kilic Afsar, Sarah Nicita, Rosalie Hsin-Ju Lin, Liu Yang, Megan Hofmann, Akshay Kothakonda, Zachary Gordon, Cedric Honnet, Kristen Dorsey, et al. 2023. FibeRobo: Fabricating 4D Fiber Interfaces by Continuous Drawing of Temperature Tunable Liquid Crystal Elastomers. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.
- [4] Anhong Guo, Junhan Kong, Michael Rivera, Frank F Xu, and Jeffrey P Bigham. 2019. Statelens: A reverse engineering solution for making existing dynamic touchscreens accessible. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 371–385.
- [5] Yvonne Jansen, Thorsten Karrer, and Jan Borchers. 2010. MudPad: tactile feedback and haptic texture overlay for touch surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*. 11–14.
- [6] Shaun K Kane, Meredith Ringel Morris, and Jacob O Wobbrock. 2013. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–8.
- [7] Van-Cuong Nguyen, Victor Oliva-Torres, Sophie Bernadet, Guilhem Rival, Claude Richard, Jean-Fabien Capsal, Pierre-Jean Cottinet, and Minh-Quyen Le. 2023. Haptic Feedback Device Using 3D-Printed Flexible, Multilayered Piezoelectric Coating for In-Car Touchscreen Interface. *Micromachines* 14, 8 (2023), 1553.
- [8] Martin Nisser, Yashaswini Makaram, Lucian Covarrubias, Amadou Yaye Bah, Faraz Faruqi, Ryo Suzuki, and Stefanie Mueller. 2022. Mixels: Fabricating Interfaces using Programmable Magnetic Pixels. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–12.
- [9] Kentaro Yasu. 2019. Magnetact: magnetic-sheet-based haptic interfaces for touch devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–8.
- [10] Neng-Hao Yu, Sung-Sheng Tsai, I-Chun Hsiao, Dian-Je Tsai, Meng-Han Lee, Mike Y Chen, and Yi-Ping Hung. 2011. Clip-on gadgets: expanding multi-touch interaction area with unpowered tactile controls. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 367–372.