

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Development of a budget 16x16 dynamic-shape display with depth perception and computer vision



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autora: Oihana Elizagoyen

Directores: Iñigo Ezcurdia y Asier Marzo

Pamplona, 6 de Septiembre de 2023

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Final Degree Project in Computer Engineering

DEVELOPMENT OF A BUDGET I6XI6
DYNAMIC-SHAPE DISPLAY WITH DEPTH
PERCEPTION AND COMPUTER VISION

Autora:
Oihana Elizagoyen

Director:
Iñigo Ezcurdia

Codirector:
Asier Marzo

September 1, 2023

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Contents

Resumen	v
Palabras clave	vii
Agradecimientos	ix
1 Introducción	I
1.1 Este proyecto: inFORM 16x16	1
1.2 Contexto de la autora	2
1.3 Human Computer Interaction	2
1.4 Dynamic Shape Displays	3
2 Justificación y objetivos	5
3 Estado del arte	7
3.1 inFORM Original	7
3.2 TFG previo inFORM 8x8	8
4 Resultados y demos	11
4.1 Resultados Hardware	11
4.2 Resultados Software	13
4.2.1 Aplicaciones anteriores	13
4.2.2 Interacción remota mediante cámaras de profundidad	14
4.2.3 Personalización y renderizado 3D de funciones matemáticas con fines educativos	16
4.2.4 Minijuego de lógica en vertical	17
4.2.5 Potenciador de un entorno Desktop mediante Computer Vision	18
5 Mecánica y Electrónica	21
5.1 Bill of materials	21

5.2	Mecánica	24
5.2.1	Estructura modular de dimensión 2x2	24
5.2.2	Piñón y cremallera	25
5.2.3	Pixel	26
5.2.4	Patas	27
5.2.5	Soporte	29
5.2.6	Resumen de Material y Tiempos de Impresión 3D	30
5.3	Electrónica	30
5.3.1	Servomotor	30
5.3.2	Arduino Nano	31
5.3.3	Controlador PWM	31
5.3.4	Fuente de alimentación	33
6	Software	35
6.1	Tecnologías empleadas	35
6.2	Interacción remota mediante cámaras de profundidad	36
6.3	Personalización y renderizado 3D de funciones matemáticas con fines educativos	39
6.4	Minijuego de lógica en vertical	40
6.5	Potenciador de un entorno Desktop mediante Computer Vision	41
7	Consejos para makers valientes	47
8	Trabajo Futuro	49
	Bibliography	51

Resumen

En este proyecto se detalla el proceso de diseño y construcción de un *dynamic shape display* capaz de realizar representaciones físicas de objetos en 3D y de moverlos en tiempo real. Inspirado en el dispositivo InForm creado por el Instituto de Tecnología de Massachusetts (MIT), se propone una alternativa económica, funcional y accesible, siguiendo una filosofía maker Do-It-Yourself que permita a aquellas personas interesadas replicar este display. Este proyecto detalla todos los aspectos relevantes para la concepción hardware y software de un prototipo final, recorriendo disciplinas como electrónica, mecánica, diseño e impresión 3D, desarrollo software y visión por computador. Se propone la mejora de un prototipo previamente existente de 8x8 (64) pistones, expandiéndose a 16x16 (256) pistones, y se acompaña de demos ejecutables cubriendo los siguiente ejemplo de uso: (1) Interacción remota mediante cámaras de profundidad; (2) Personalización y renderizado 3D de funciones matemáticas con fines educativos; (3) Minijuego de lógica en vertical; (4) Software de dibujo libre; (5) Potenciador de un entorno Desktop mediante Computer Vision.

Palabras clave

dynamic shape display, human computer interaction, inform, cámaras de profundidad, computer visión

Agradecimientos

Me gustaría agradecer a mi director Iñigo Ezcurdia por darme la oportunidad de realizar este proyecto y ayudarme enormemente durante su desarrollo. He aprendido mucho de él y sin su ayuda no hubiese sido capaz de llevar a cabo el proyecto. También me gustaría agradecer a mi codirector Asier Marzo que me ha ayudado siempre que lo he necesitado y al resto de miembros del UpnaLab que no han dudado en ayudarme en lo que han podido. Gracias a mis amig@s por creer en mí y valorar mi esfuerzo. Gracias a mis compañer@s de laboratorio por su compañía durante el desarrollo de este proyecto, sin ellos el proceso habría sido mucho más aburrido. Gracias a mi familia por cuidarme y apoyarme siempre. Pero sobre todo me gustaría agradecer a mi madre, mi padre y mi hermano, por estar siempre ahí y apoyarme en todas mis decisiones.

Introducción

1.1

ESTE PROYECTO: inFORM 16x16

En este proyecto se detalla el proceso de diseño y construcción de un *dynamic shape display* capaz de realizar representaciones físicas de objetos en 3D y de moverlos en tiempo real.

Este proyecto se inspira en inFORM [11], un dispositivo creado por el Instituto de Tecnología de Massachusetts (MIT). inFORM es un 2,5D shape display de última generación, el cual permite explorar las posibilidades de interacción con los dispositivos shape display y pretende allanar el camino de la investigación en este campo. Para más detalles sobre este dispositivo, acudase a la sección "3.1 - Estado del Arte - inFORM original".

Por otra parte, cabe destacar que este proyecto no parte de cero, ya que se fundamenta en el espléndido trabajo realizado por Yuxiang Xu en su Trabajo de Fin de Grado (TFG) titulado "Budget inFORM, a dynamic shape display" [14]. En su TFG, Yuxiang detalló la construcción de una réplica asequible de inFORM con una resolución de 8x8 pistones, así como dos aplicaciones que permiten interactuar con dicho dispositivo. Para más detalles sobre este dispositivo, acúdase a la sección "3.2 - Estado del Arte - TFG previo inFORM 8x8".

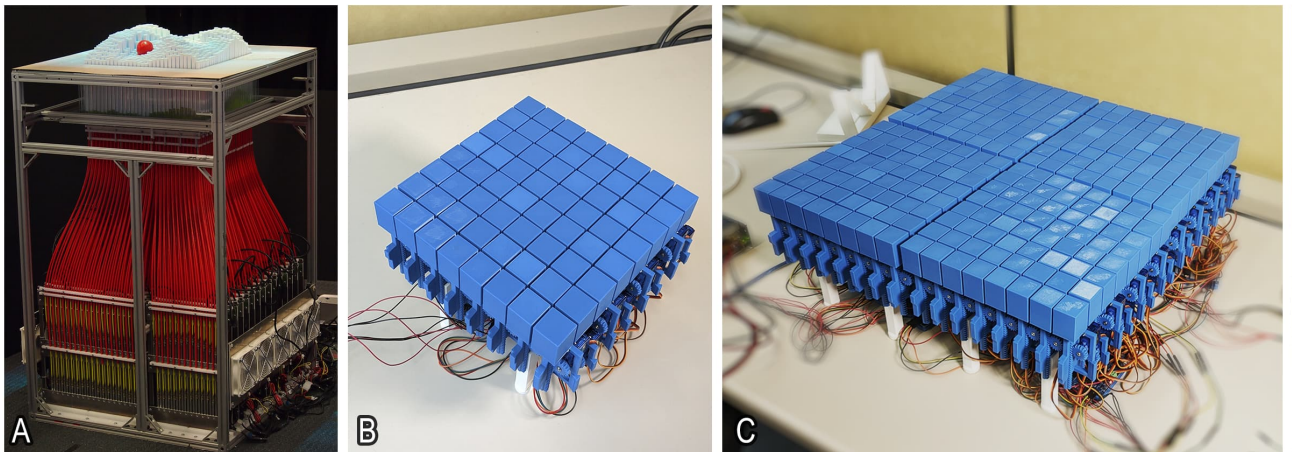


Figure 1.1: A) inFORM original desarrollado por el MIT en 2013. B) Réplica económica 8X8 desarrollada por Yuxiang Xu en UPNA. C) Este proyecto: Modelo ampliado a 16x16 pistones con nuevas funcionalidades.

Por consiguiente este TFG pretende, por un lado, ampliar la resolución de dicho dispositivo a 16x16 pistones, adaptando su hardware y software previamente existentes a dicha resolución y, por otro lado, desarrollar 5 nuevas aplicaciones de interacción con este dispositivo a modo de ejemplos de casos de uso en distintos ámbitos. Los resultados obtenidos, tanto hardware como software, se exponen en la sección "4 - Resultados y Demos". Posteriormente, con la intención de guiar e informar a cualquier persona interesada en la fabricación de su propio dispositivo, se diagraman y detallan todos los aspectos electromecánicos en la sección "5

- Mecánica y Electrónica". De igual modo, en la sección "6 - Software" se detallan también aquellos aspectos relevantes sobre el desarrollo de software requerido para operar y personalizar el comportamiento del dispositivo en los distintos casos de uso escogidos como ejemplo.

Por último en la sección "8 -Trabajo Futuro", se enumeran posibles ideas y puntos de progreso que sería interesante atajar en una posible futura expansión y/o mejora del dispositivo.

Diseños 3D, circuitería y código fuente quedan disponibles en el siguiente repositorio:
<https://github.com/InigoEzcurdia/InForm2023>

I.2

CONTEXTO DE LA AUTORA

Entre los diversos proyectos que me fueron propuestos para desarrollar, este fue el que más interés me generó, ya que se trata de un proyecto que combina múltiples campos: electrónica, mecánica, fabricación, desarrollo de software, visión por computador... Este proyecto no está totalmente relacionado con la Ingeniería del Software, que es la mención en la que me he especializado durante el grado, pero ahí es donde reside parte de mi interés por él. El hecho de poder expandir mi conocimiento y experiencia en campos como la electrónica, mecánica o visión por computador me pareció una muy buena oportunidad para aprender y motivarme. Si bien es cierto que los conocimientos obtenidos en la mención que he cursado me han permitido desarrollar aplicaciones con un código más limpio y de mayor calidad. También existen otras asignaturas de la carrera que he podido poner en práctica a la hora de desarrollar este proyecto. La asignatura de "Fundamentos de la electrónica" me proporcionó una pequeña base de conocimientos de electrónica, campo con el que he trabajado en este proyecto. Además, me gustaría destacar que, de todas las asignaturas de la carrera, la que me ha sido de mayor utilidad para este proyecto ha sido "Sistemas multimedia y diseño centrado en el usuario", puesto que en dicha asignatura aprendimos, entre otras cosas, nociones sobre Arduino, a soldar y a imprimir en 3D, conocimientos que me han sido necesarios para este proyecto.

Este proyecto ha sido realizado en colaboración con el laboratorio UpnaLab [4], un grupo de investigación de la Universidad Pública de Navarra dedicado a la investigación sobre "Human Computer Interaction" mediante la combinación de computación, física, electrónica, mecánica y química para la creación de dispositivos interactivos.

I.3

HUMAN COMPUTER INTERACTION

"Human Computer Interaction, en castellano Interacción Humano Computador, es una disciplina que se ocupa del diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano y del estudio de los principales fenómenos que los rodean" [13].

El concepto de Interacción Humano Computador surgió con la invención del primer computador. Desde entonces, la Interacción Humano Computador ha evolucionado enormemente gracias a los grandes esfuerzos

que se han realizado para encontrar nuevas y mejores formas de que los usuarios interactúen con los sistemas informáticos. El principal objetivo del Human Computer Interaction es mejorar las interacciones entre usuarios y computadoras, logrando una mayor usabilidad y respuesta a las necesidades de usuario.

El Human Computer Interaction es muy importante debido al gran aumento de los computadores en la vida cotidiana. Resulta necesario encontrar una forma de interactuar que permita una comunicación segura y efectiva entre los humanos y los computadores. En este contexto, las nuevas propuestas en el campo del HCI son cruciales para el diseño de interfaces intuitivas que faciliten o extiendan su uso y accesibilidad de caras a personas con distintas habilidades y conocimientos o experiencia.

Muchos de los grandes avances en HCI provienen de la propuesta de nuevos displays o de la propuesta de nuevos dispositivos de interacción con ellos. Este proyecto pretende cubrir ambos caminos con un solo dispositivo. Un nuevo display capaz de renderizar contenido tangible en 3 dimensiones, capaz tanto de actuar como de sensar su entorno. Este proyecto ofrece una ventana al campo del "Human Computer Interaction", desde su primitivas más tecnológicas y técnicas para la construcción del dispositivo, hasta una perspectiva mas cercana a la persona humana, mediante novedosas propuestas de interacción.

I.4

DYNAMIC SHAPE DISPLAYS

Las pantallas 2D, el tipo de display más común en nuestras interacciones diarias, permite visualizar contenidos a todo color y resolución, pero siempre contemplando únicamente 2 dimensiones y negando a las personas usuarias la posibilidad de interactuar física y tangiblemente con los elementos renderizados.

Las nuevas tecnologías de Realidad Virtual 3D añaden una tercera dimensión a los renderizados. Aprovechando la visión estereoscópica del ser humano, proveen de una imagen ligeramente distinta a cada ojo, traduciendo esas pequeñas diferencias en percepción de profundidad. Sin embargo, todavía no permiten a las personas usuarias tocar, palpar y sentir los elementos con los que interactúan. Algunas novedosas tecnologías intentan dar solución a esta problemática mediante complejos mecanismos wearables.

Sin embargo los Dynamic Shape Displays atajan dichos problemas e incluso ofrecen novedosas posibilidades de interacción adicionales. Leithinger et al. en [12] definen Dynamic Shape Displays como *“una nueva clase de dispositivos de entrada/salida (E/S) que representan dinámicamente formas físicas y geométricas. Permiten que múltiples usuarios experimenten información a través del tacto y la deformación de su topología de superficie. Las formas renderizadas pueden reaccionar a la entrada de usuario o actualizar continuamente sus propiedades en función de una simulación subyacente”*.

A menudo vinculados a los *shape-changing displays*, los dynamic shape displays tienen la capacidad de cambiar su configuración y forma, ofreciendo una superficie tangible y mutable con la que interactuar. En algunos casos esta misma superficie puede utilizarse tanto como interfaz de entrada/input como de salida/actuación. Además liberan a las personas usuarias de la carga de depender de dispositivos wearables adicionales para "sentir" el tacto de los elementos con los que interactúan.



Figure 1.2: Ejemplos de Shape Changing Displays. Fila superior) En ciencia ficción; Fila inferior) Dispositivos reales implementados por la comunidad científica.

II

Justificación y objetivos

Los “*displays*” o pantallas de visualización han sido parte integral de la historia de la humanidad, evolucionando su tecnología, forma y función conforme evolucionaba el conocimiento humano. Desde los primeros displays electrónicos, como el tubo de rayos catódicos (CRT), hasta los modernos displays de cristal líquido (LCD) o de diodos emisores de luz orgánicos (OLED), este tipo de tecnologías han sido explotadas en prácticamente todos los ámbitos fundamentales de nuestras vidas y sociedades. Comunicación, entretenimiento y ocio, trabajo y productividad, educación y consumo o divulgación de información han evolucionado al ritmo de las tecnologías de display. Al tratarse de herramientas versátiles y omnipresentes en nuestro mundo, cualquier progreso en el ámbito de los displays supone un impacto singularmente importante en nuestra sociedad en todos sus estratos. En los últimos años la evolución y desarrollo de displays e interfaces persona-máquina ha experimentado avances significativos, pasando de interfaces gráficas estáticas a sistemas interactivos de inmersión total. En este contexto, los “*dynamic shape displays*” abren las puertas a nuevas experiencias táctiles y visuales que van más allá de las capacidades de las pantallas tradicionales.

El hecho de que inFORM sea un dispositivo que ofrece al usuario múltiples formas para interactuar con él y en múltiples ámbitos de la vida, es una de las principales razones por las que merece la pena desarrollar este proyecto. Por ejemplo, inFORM puede ser útil en el ámbito académico, permitiendo mostrar contenido 3D de forma física para una mayor comprensión de los conceptos, o también puede ser útil en cualquier ámbito que requiera una comunicación a distancia, permitiendo a los participantes remotos mostrarse físicamente para dar una mayor sensación de presencia. Es por ello que el desarrollo de inFORM es un proyecto con multitud de posibilidades de crecimiento, puesto que se pueden desarrollar tanto nuevas formas de interactuar con este dispositivo como nuevas aplicaciones que permitan emplear inFORM en diferentes ámbitos de la vida.

La idea detrás de hacer un dynamic shape display más económico y accesible es sencilla: queremos que más gente pueda experimentar y beneficiarse de esta innovación. Si logramos reducir su costo, más personas y empresas podrán usarlo, lo que significa más oportunidades para descubrir nuevas aplicaciones, desde la educación hasta el entretenimiento. Nuestro objetivo es claro: democratizar el acceso a esta tecnología y fomentar un ambiente de creatividad y exploración en torno a ella.

Por ello, este proyecto propone los siguientes objetivos:

1. **Revisar y entender** diseño, electrónica, mecánica y software del prototipo de inFORM 8x8 desarrollado en un trabajo de fin de grado previo.
2. **Construir** un primer prototipo 8x8, familiarizándose con la tecnología y su funcionamiento.
3. Realizar diversas **mejoras hardware**:
 - Aumento a resolución 16x16 manteniendo la modularidad.
 - Adición de una peana que permita colocar el display verticalmente.
 - Sustitución de patas de soporte para evitar espacios innecesarios al anexas módulos 8x8.

4. Realizar diversas **mejoras software**:
 - Adaptar las dos aplicaciones de demostración ya existentes para operar en el nuevo display 16x16.
 - Software de dibujado libre.
 - Software de renderizado de funciones matemáticas.
 - Diseñar y desarrollar 4 nuevas aplicaciones que expandan la funcionalidad y aplicabilidad del display en distintos ámbitos:
 - Interacción remota mediante cámaras de profundidad.
 - Personalización y renderizado 3D de funciones matemáticas con fines educativos.
 - Minijuego de lógica en vertical.
 - Potenciador de un entorno Desktop mediante Computer Vision.
5. **Disponibilizar** código, diseños, circuitería y comentarios que informen y detallen todo este proceso para que aquellas personas interesadas puedan construir y/o mejorar su propio dynamic shape display.

III

Estado del arte

3.1

inFORM ORIGINAL

El "*MIT Media Lab*" [1] es un laboratorio dentro de la Escuela de Arquitectura y Planificación en el Instituto de Tecnología de Massachusetts que se dedica a proyectos de investigación que combinan diferentes áreas: diseño, multimedia y tecnología. Cabe destacar que el "*MIT Media Lab*" no es una única institución, sino que está formado por diferentes grupos. "*Tangible Media Group*" (TMG) [5] es un grupo de investigación del "*MIT Media Lab*" centrado en ampliar los límites de "*Human Computer Interaction*" (HCI - Interacción persona-ordenador). Este último grupo fue el responsable de desarrollar inFORM [6, 11], el cual es un "*Dynamic Shape Display*" que puede representar contenido 3D de forma física, para que las personas usuarias interactúen con la información digital de forma tangible. Además, inFORM también puede interactuar con el mundo físico que le rodea, por ejemplo moviendo objetos en su superficie.

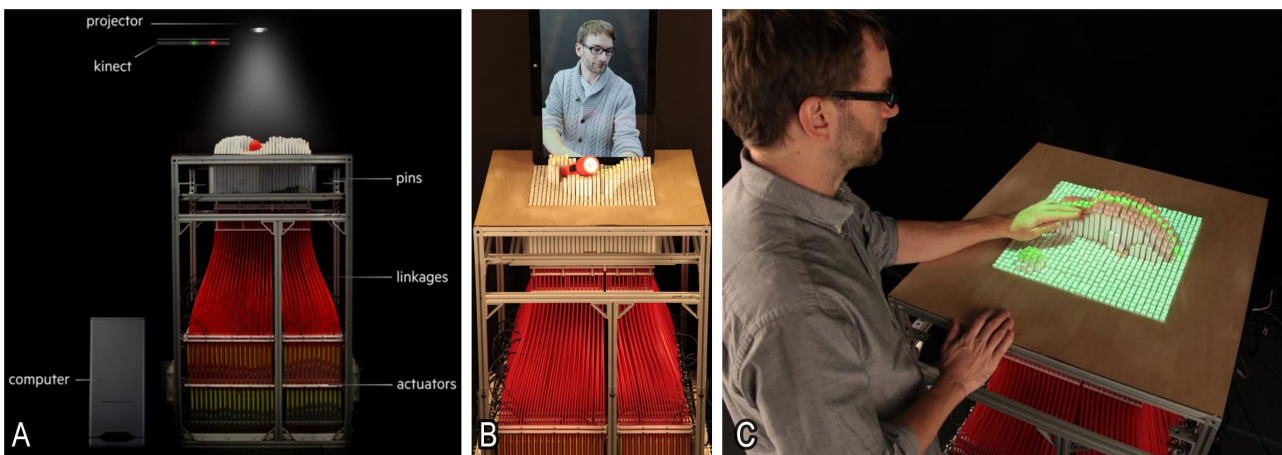


Figure 3.1: A) Concepto general del inFORM original. B) inFORM original empleado para replicar interacción física de forma remota. C) inFORM original renderizando un coche digital de forma tangible.

El inFORM desarrollado por el MIT cuenta con una resolución de 30x30 en un área de 381mmx381mm y está formado por clavijas motorizadas de poliestireno blanco. Cada una de las clavijas tiene un área de 9.525 mm² y puede desplazarse 100 mm en vertical. La separación entre clavijas es de 3.175 mm. Para conectar cada una de las clavijas con su actuador correspondiente se emplea un cable Push-Pull. De esta forma, el tamaño de las clavijas es independiente del tamaño del actuador, lo que permite una disposición más densa de las clavijas del inFORM. Además, debido a esto, el inFORM alcanza una altura de 1100 mm.

Desde el MIT, se propusieron tres aplicaciones para el inFORM. La primera de las aplicaciones es "*3D Model Manipulation*". Dicha aplicación demuestra como las capacidades dinámicas del sistema inFORM pueden ser utilizadas para renderizar contenido en 3D y emplear fichas y herramientas para transformar, editar y pin-

tar. La segunda aplicación propuesta por el MIT es “*Marble Answering Machine*”. Esta aplicación permite recibir, guardar y reproducir mensajes de voz, los cuales se representan por medio de canicas. La tercera aplicación es “*Actuating Interactive Devices on the Surface*”. Esta aplicación permite activar dispositivos interactivos, como teléfonos o tabletas, que también se pueden mover, inclinar, girar y vibrar.

Sin embargo, el inFORM desarrollado por el MIT es un aparato grande y costoso, así como complejo a nivel mecánico. Se trata de un dispositivo no accesible para los usuarios finales, no está disponible comercialmente y para hacer uso de él, es necesario concertar una cita y acudir al MIT.

3.2

TFG PREVIO inFORM 8x8

En el TFG titulado “*Budget inFORM, a Dynamic Shape Display*” [14] se pretendió replicar el inFORM desarrollado por el MIT, aunque de una forma económica y fácil de montar. Cabe decir que estaba fuera del alcance de dicho proyecto el replicar todas las funcionalidades con las que cuenta el inFORM desarrollado por el MIT. Como resultado de dicho proyecto se obtuvo un inFORM de dimensión 8x8. Dicho inFORM está compuesto por 16 estructuras modulares, cada una de las cuales es capaz de albergar 4 servo motores, dando como resultado un total de 64 servo motores, uno por cada pistón del inFORM. Para poder mover cada uno de los pistones, no solo es necesaria la presencia de un servomotor, si no que también se emplea un actuador lineal de piñón y cremallera que permite transformar el movimiento rotacional del servomotor en un movimiento lineal. Para poder controlar cada uno de los servomotores, se emplea un controlador PWM, concretamente el PCA9685, y todos esos controladores están conectados en serie a un Arduino Nano, que es el encargado de enviar la información a los controladores PWM sobre la posición en la que debe estar cada servo motor. Además, como resultado de ese TFG también se obtuvieron dos aplicaciones desarrolladas con el lenguaje de programación Python. Una de esas aplicaciones permite dibujar sobre la superficie del inFORM con una interfaz de imagen, mientras que la otra aplicación muestra funciones matemáticas predefinidas en bucle. En ambos casos, la información sobre la altura de cada pistón se envía desde Python a Arduino, para que este a su vez la envíe a los controladores PWM.

Sin embargo, la replica del inFORM desarrollada por Yuxiang Xu presenta algunas limitaciones:

- El dispositivo de dimensión 8x8 cuenta con poca resolución, lo que provoca que el contenido renderizado en 3D tenga poco detalle.
- La información para interactuar con el dispositivo únicamente se puede transmitir a través del ratón o del teclado. Incorporar nuevas formas de interactuar con el dispositivo podría mejorar su usabilidad.
- Las patas con las que cuenta el dispositivo son un poco endebles, y la rotura de una de ellas podría provocar que el dispositivo se cayera. Para aumentar la seguridad del dispositivo, se podrían desarrollar unas patas más fuertes y robustas. Además, habría que reubicarlas a una zona interior para facilitar el anexo de más módulos en sus laterales. Por otra parte, conviene rediseñar unas patas que permitan colocar el dispositivo en vertical de cara a ciertas aplicaciones.

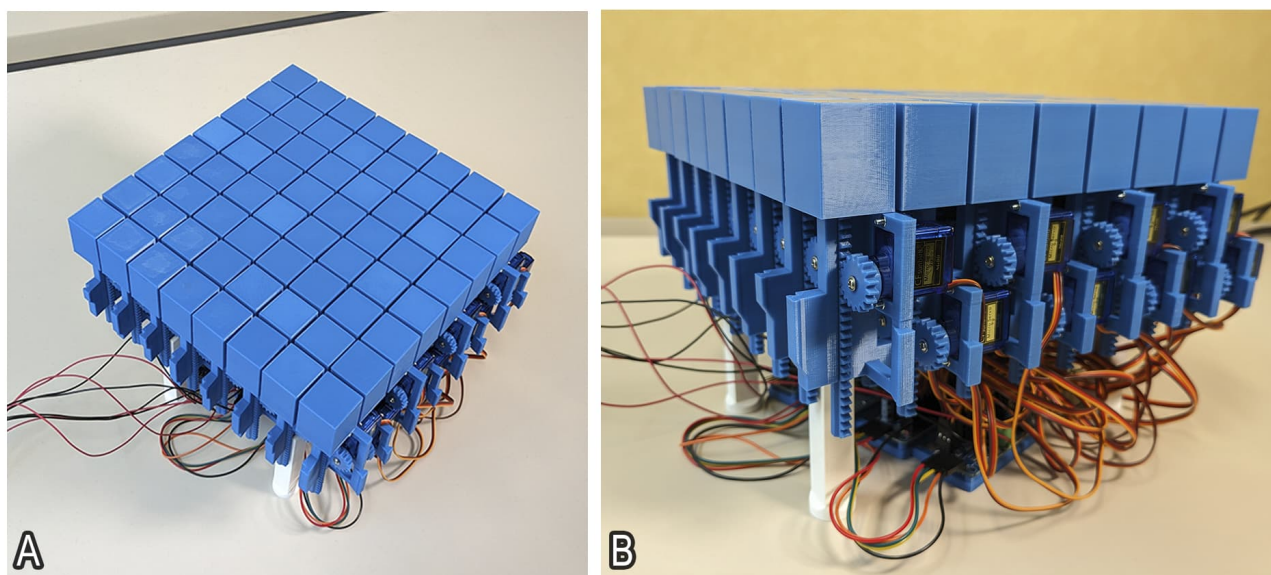


Figure 3.2: A) Replica 8X8 de inFORM desarrollada por Yuxiang Xu B) Lateral de replica 8x8. Se observa el sistema de servos y piñones empleados para actuar los pistones.

IV

Resultados y demos

En esta sección se recorren los objetivos propuestos en la sección "2 - Justificación y Objetivos". Consideramos que todos ellos han sido alcanzados satisfactoriamente. A continuación se exponen los resultados obtenidos, tanto su espectro hardware como software.

4.1

RESULTADOS HARDWARE

En la parte de hardware, como resultado de este proyecto se ha obtenido un inFORM de dimensión 16x16. Se trata de un dispositivo compuesto por 4 módulos de dimensión 8x8. Cada módulo 8x8 mide 11,5 cm de alto, 25 cm de ancho y 23 cm de largo y cuenta con 64 pistones. Cada uno de los pistones que componen el inFORM es capaz de desplazarse verticalmente 26,69 mm. El desplazamiento de los pistones hace que el dispositivo sea capaz de mostrar contenido 3D de forma física, así como interactuar con el mundo que le rodea.

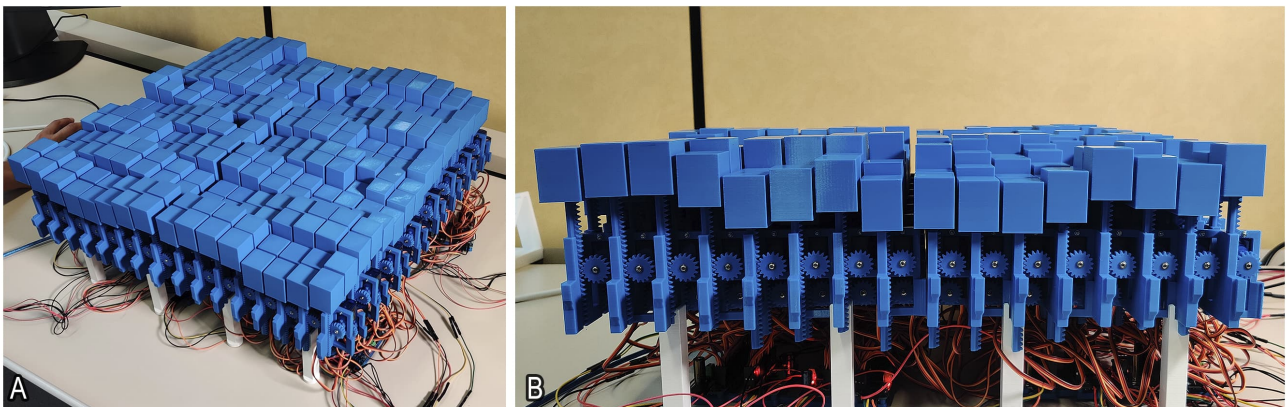


Figure 4.1: A) inFORM 16x16 desarrollado en este proyecto B) Lateral de inFORM 16x16. Sistema de piñones y engranado disponiendo pistones a distintas alturas.

El desplazamiento de cada uno de los pistones es posible gracias a los servo motores con los que cuenta el dispositivo. Cada uno de los módulos 8x8 cuenta con el mismo número de servo motores que de pistones. La energía necesaria para que todos los servo motores de un módulo 8x8 puedan desplazarse es proporcionada por una fuente de alimentación. Cada módulo 8x8 necesita una fuente de alimentación que proporcione una salida de corriente continua(DC) de 5V y 30A. Por tanto, para el funcionamiento del dispositivo completo, se requieren 4 fuentes de alimentación que proporcionen dicha salida.

Cabe destacar que cada uno de los 4 módulos que componen el inFORM puede ser separado del resto, es decir, el inFORM de dimensión 16x16 no es un dispositivo único, si no que se puede dividir de forma muy sencilla en los 4 módulos que lo componen. Esta característica es bastante beneficiosa. Por un lado, el transporte del dispositivo de dimensión 16x16 puede ser un tanto complicado y su división permite la obtención

de dispositivos bastante más pequeños, cuyo transporte resulta mucho más sencillo. Por otro lado, habrá ocasiones en las que emplear un dispositivo de dimensión 16×16 no sea necesario y resulte demasiado grande. La división del inFORM en módulos permite que cada uno de ellos pueda ser empleado de forma independiente como si de un único dispositivo se tratase.

Además, el dispositivo cuenta con un soporte y varias patas. Por un lado, las patas permiten colocar el dispositivo en horizontal. En horizontal puede ser colocado tanto al dispositivo 16×16 en su conjunto como cada uno de los módulos 8×8 si estos se emplean por separado. Por otro lado, el soporte únicamente permite colocar un módulo 8×8 en vertical diagonalmente.

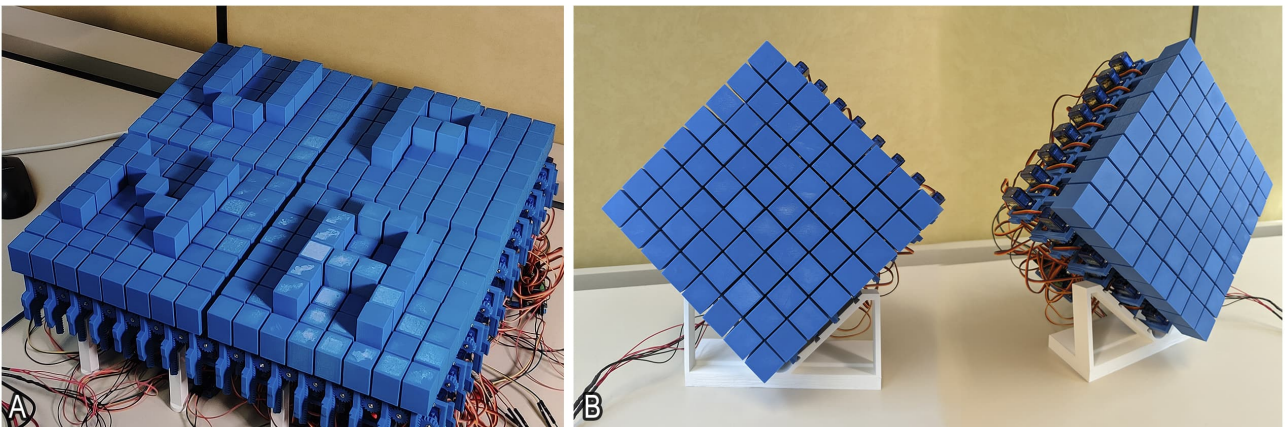


Figure 4.2: A) inFORM 16×16 , compuesto de 4 módulos 8×8 , colocados horizontalmente mediante patillas impresas en 3D. B) inFORM 8×8 colocado verticalmente en diagonal mediante un soporte impreso en 3D.

4.2

RESULTADOS SOFTWARE

En la parte de software, como resultado de este proyecto se han obtenido diversas aplicaciones que permiten al usuario interactuar con el inFORM de diversas formas. Por un lado, se han adaptado las aplicaciones existentes para que también puedan ser empleadas por el dispositivo 16×16 y, por otro lado, se han creado nuevas aplicaciones. En la sección "6 - Software", se detalla en profundidad cómo han sido diseñadas e implementadas cada una de estas nuevas aplicaciones. A continuación, explicaré en qué consiste cada una de las aplicaciones y por qué las hemos escogido.

4.2.1

APLICACIONES ANTERIORES

El proyecto anterior dio como resultado dos aplicaciones. La primera de las aplicaciones permite dibujar en el dispositivo gracias a una interfaz de dibujo. Vease figura 4.3.

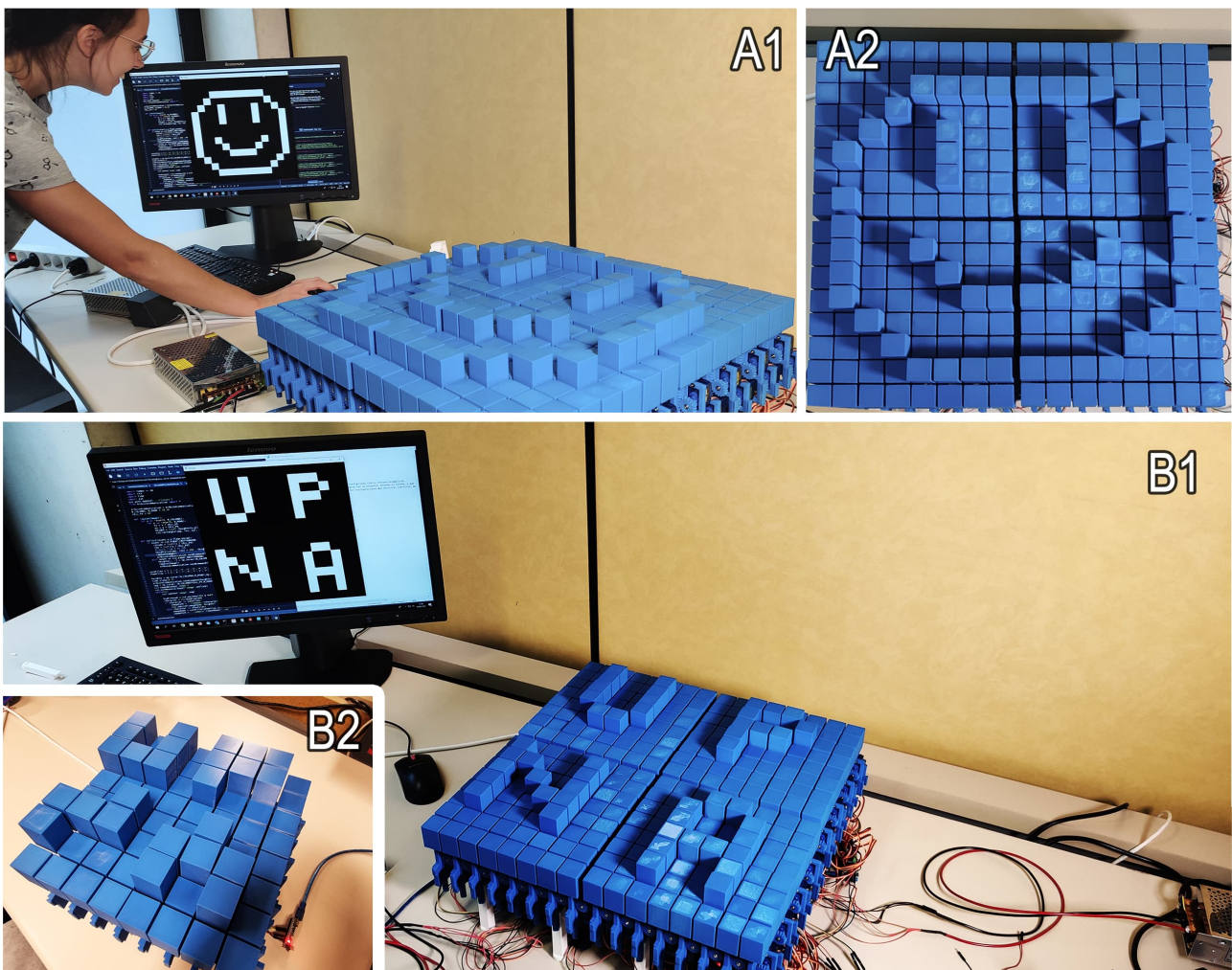


Figure 4.3: A1) Usuaría utilizando la app de dibujo libre en inFORM 16×16 . A2) Vista cenital de un dibujo smiley en 16×16 . B1) Dibujo de un letrero "UPNA" en 16×16 . B2) Comparación con un letrero similar "UPNA" en 8×8 .

Esta aplicación también permite que las imágenes dibujadas se puedan guardar en su archivo correspondiente pulsando las teclas ['1','2','3','4','q','w','e','r'], así como visualizar las imágenes guardadas pulsando las teclas ['5','6','7','8','t','y','u','i'], correspondiéndose las imágenes visualizadas con las guardadas según el orden de las teclas en las listas.

La segunda de las aplicaciones reproduce funciones matemáticas predefinidas en bucle. Además, también permite cambiar la función a mostrar pulsando la tecla A o la tecla D, de forma que la tecla A te permite volver a la función anterior, mientras que la tecla D te permite pasar a la siguiente función. Ambas aplicaciones han sido adaptadas para que, además de poder funcionar con un módulo 8x8, también puedan ser empleadas con un inFORM de dimensión 16x16. Vease figura 4.4.

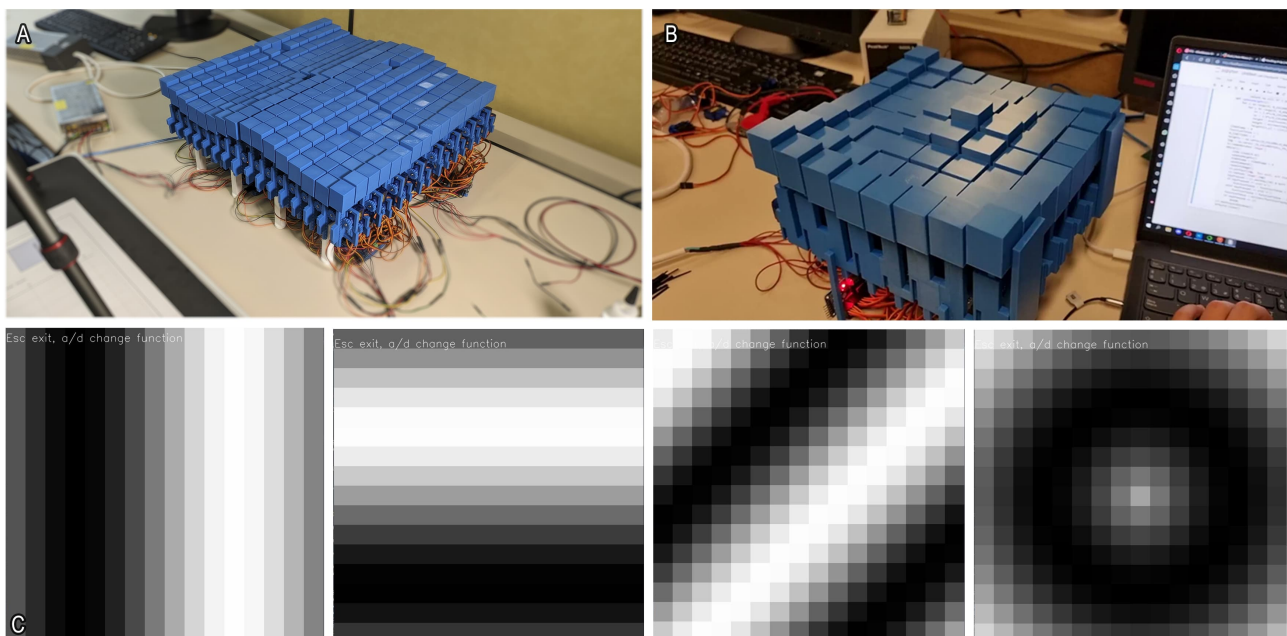


Figure 4.4: A) inFORM 16x16 renderizando la función mostrada en el mapa de alturas C1. B) inFORM 8x8 renderizando la función mostrada en el mapa de alturas C4. C) Mapas de alturas de las distintas funciones renderizables en nuestros inFORM. Los renderizados son animados y cambian su forma en tiempo real.

4.2.2

INTERACCIÓN REMOTA MEDIANTE CÁMARAS DE PROFUNDIDAD

Esta aplicación, gracias al uso de una Kinect, permite detectar la profundidad a la que se encuentran los objetos que hay dentro de la zona de interés. Con dicha información, la aplicación lo que hace es representar en el inFORM los objetos. Vease figura 4.5.

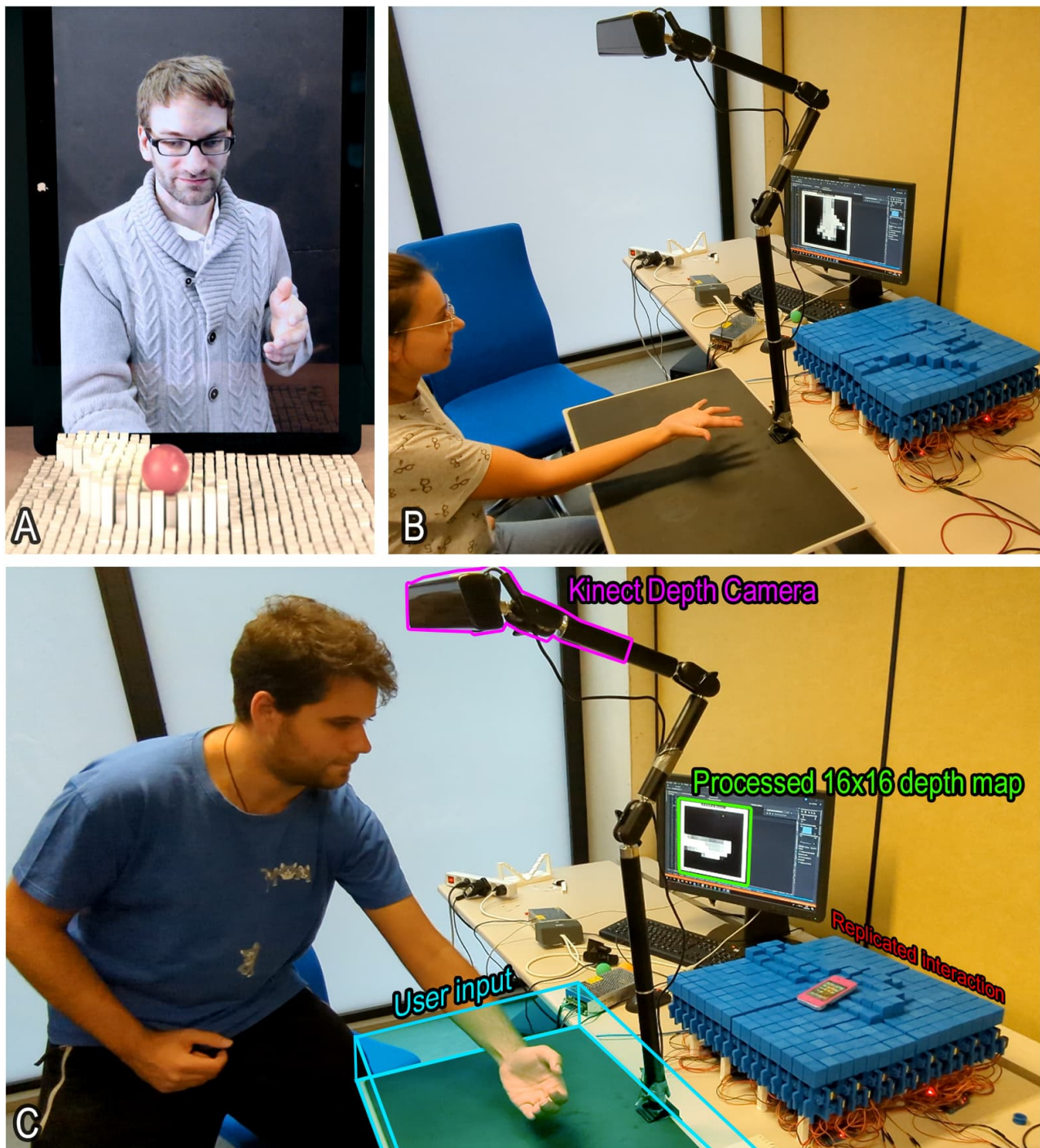


Figure 4.5: A) Interacción remota mediante cámaras de profundidad en inFORM original. B) Interacción remota mediante cámara de profundidad kinect en nuestro inFORM 16x16. C) La persona usuaria coloca su(s) mano(s) u cualquier otro elemento en el volumen de input. Una cámara kinect obtiene información de profundidad del volumen de input. Un PC realiza los reescalados y filtrados necesarios para traducir dicha información en una mapa de alturas 16x16 que es enviado en tiempo real al dispositivo inFORM.

Decidimos desarrollarla, entre otras cosas, porque es una de las aplicaciones desarrolladas por "Tangible Media Group" del MIT y, en cuanto la vimos, nos llamó bastante la atención y nos pareció una aplicación muy

interesante a la hora de desarrollar. También la escogimos porque es una aplicación que permite a un usuario remoto interactuar con este dispositivo a distancia, dando una mayor sensación de presencia, lo cual podría ser de utilidad en diversos escenarios.

4.2.3

PERSONALIZACIÓN Y RENDERIZADO 3D DE FUNCIONES MATEMÁTICAS CON FINES EDUCATIVOS

Esta aplicación aparentemente es similar a una de las aplicaciones desarrolladas en el proyecto anterior, más concretamente, la aplicación que reproduce en el inFORM funciones matemáticas en bucle. Sin embargo, hay una gran diferencia entre ambas aplicaciones. Esta nueva aplicación, además de reproducir en el inFORM una función sinusoidal en bucle, también permite al usuario personalizar en tiempo real ciertos parámetros de la onda renderizada. Para poder variar los parámetros de la función sinusoidal, existe una "etiqueta" que contiene un gráfico donde el eje y representa la amplitud y el eje x representa la frecuencia y, también, contiene dos "botones", uno que permite configurar el eje x de la función sinusoidal en vertical y el otro permite configurar el eje x en horizontal. Mediante el uso de una cámara, se puede detectar donde está el objeto que le permitirá al usuario configurar los parámetros. En este caso, dicho objeto será una pelota verde. En función de la zona de la "etiqueta" en la que se encuentre la pelota verde, se modificarán unos parámetros u otros. Vease figura 4.6.

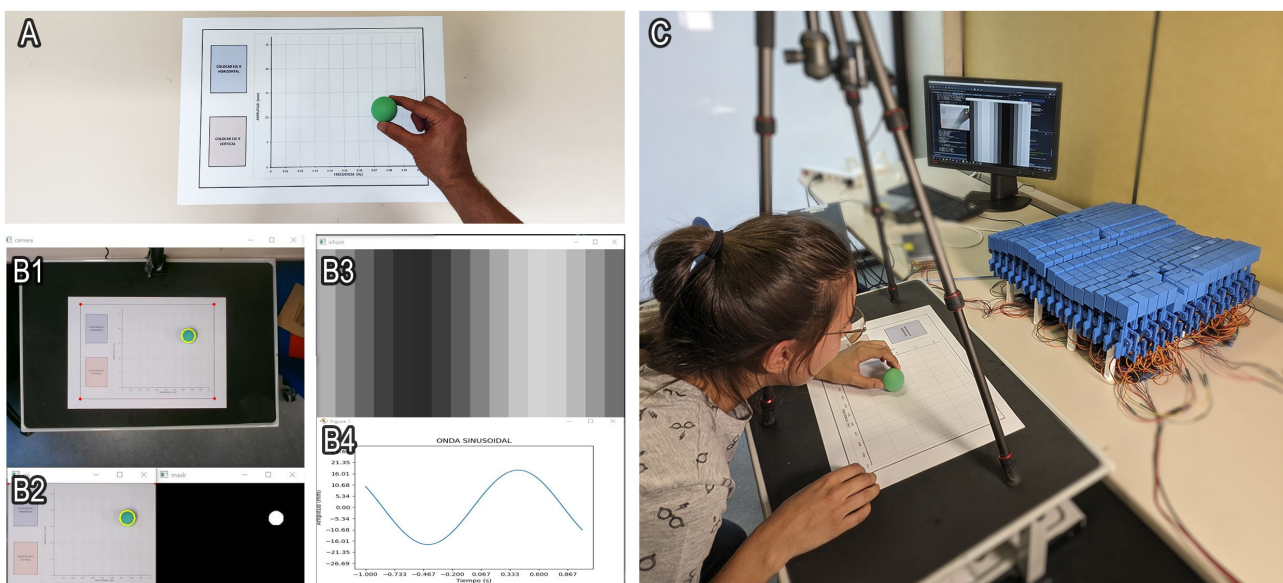


Figure 4.6: A) Input captado por una webcam común y corriente. Infinidad de dashboards pueden ser impresos y personalizados por software en base a la ubicación de uno o varios selectores físicos. B) Proceso de computer vision al que se somete el feed de la webcam. Definición de áreas de interés, corrección de perspectiva y detección de blobs; B1) Feed de la webcam con puntos rojos indicando el área de interés seleccionado por la persona usuaria; B2) Umbralizado y detección de blobs en el área de interés; B3) Mapa de alturas 16x16; B4) Onda sinusoidal representada por el mapa de alturas. C) Persona usuaria experimentando con ondas sinusoidales en un inFORM 16x16. La persona usuaria puede reajustar la orientación, frecuencia y amplitud de la onda en tiempo real.

Esta aplicación pretende demostrar como la visión por computador, en combinación con distintos objetos, permite expandir y personalizar los inputs al dispositivo más allá de la tradicional interacción mediante teclado y/o ratón. Un mismo inFORM podría ser manipulado mediante una variedad de elementos de entrada, cambiando su funcionalidad y configuración en base al elemento escogido por el usuario para su interacción

De la misma forma que en esta aplicación el inFORM se manipula empleando como entrada el desplazamiento de una pelota sobre una “etiqueta”, se podría realizar un aplicación en la que el inFORM fuese enfocado por la webcam y se manipulase empleando como input el desplazamiento de una pelota sobre el propio inFORM. Tomando como base la aplicación “Personalización y renderizado 3D de funciones matemáticas con fines educativos” desarrollada en este proyecto, sería bastante sencillo desarrollar esta nueva aplicación, pues la idea es muy similar. Además, el propio inFORM podría restringir y guiar el movimiento de la mano de la persona usuaria. Vease figura 4.7

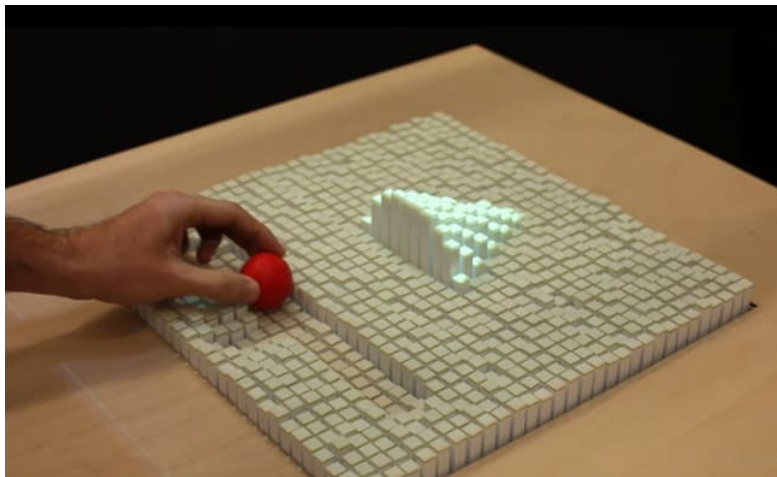


Figure 4.7: inFORM del MIT restringiendo y guiando el movimiento de la mano de la persona usuaria que interactúa directamente sobre su superficie

4.2.4

MINIJUEGO DE LÓGICA EN VERTICAL

Esta aplicación es básicamente un juego que consiste en hacer llegar una pelota desde un punto inicial hasta un punto final. Cuando el usuario comienza una nueva partida, todos los pistones del inFORM suben, excepto los correspondientes al punto inicial y al punto final. El punto inicial es el primer pistón del dispositivo, mientras que el punto final es un pistón elegido de forma aleatoria de entre los pistones de la última fila y los de la última columna. Para hacer avanzar la pelota, el usuario debe indicar, a través de la interfaz de la aplicación, cuales son los pistones que desea que bajen. Cuando el usuario decide finalizar la partida, todos los pistones bajan. Cabe destacar que para poder jugar a este juego, es necesario colocar el inFORM en vertical y diagonalmente. Vease figura 4.8

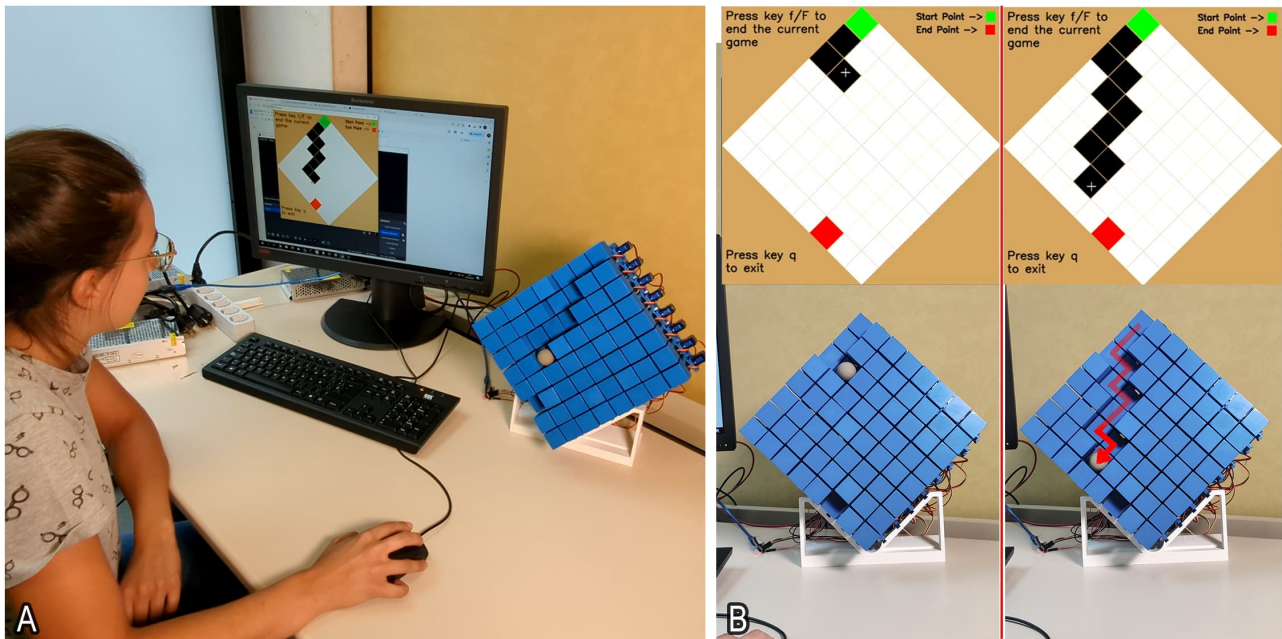


Figure 4.8: A) Jugadora utilizando un inFORM 8x8 en vertical para jugar al minijuego de físicas. B) Dos momentos distintos del minijuego. La pelota sigue el recorrido dibujado por la persona jugadora, siempre y cuando esta haya tenido en cuenta el efecto de la gravedad.

Esta aplicación demuestra que:

- inFORM también puede ser empleado en el ámbito del ocio y el entretenimiento.
- A diferencia de los displays tradicionales 2D, los dynamic shape displays permiten aprovechar su tangibilidad y tridimensionalidad para explotar mecánicas relacionadas con las propiedades físicas del entorno. En este caso, aprovechar la fuerza de la gravedad.

4.2.5

POTENCIADOR DE UN ENTORNO DESKTOP MEDIANTE COMPUTER VISION

Esta aplicación, mediante el uso de una cámara, detecta la presencia de un teléfono móvil en la superficie del inFORM. Esta aplicación no sólo es capaz de saber que hay un teléfono móvil, sino que también es capaz de saber cuales son las celdas del inFORM sobre las que se apoya el dispositivo móvil. Con esa información, esta aplicación es capaz de desplazar el teléfono móvil hasta un punto concreto de la superficie del inFORM. El móvil es desplazado gracias al movimiento de los pistones del inFORM, que suben y bajan según sea necesario. Vease figura 4.9

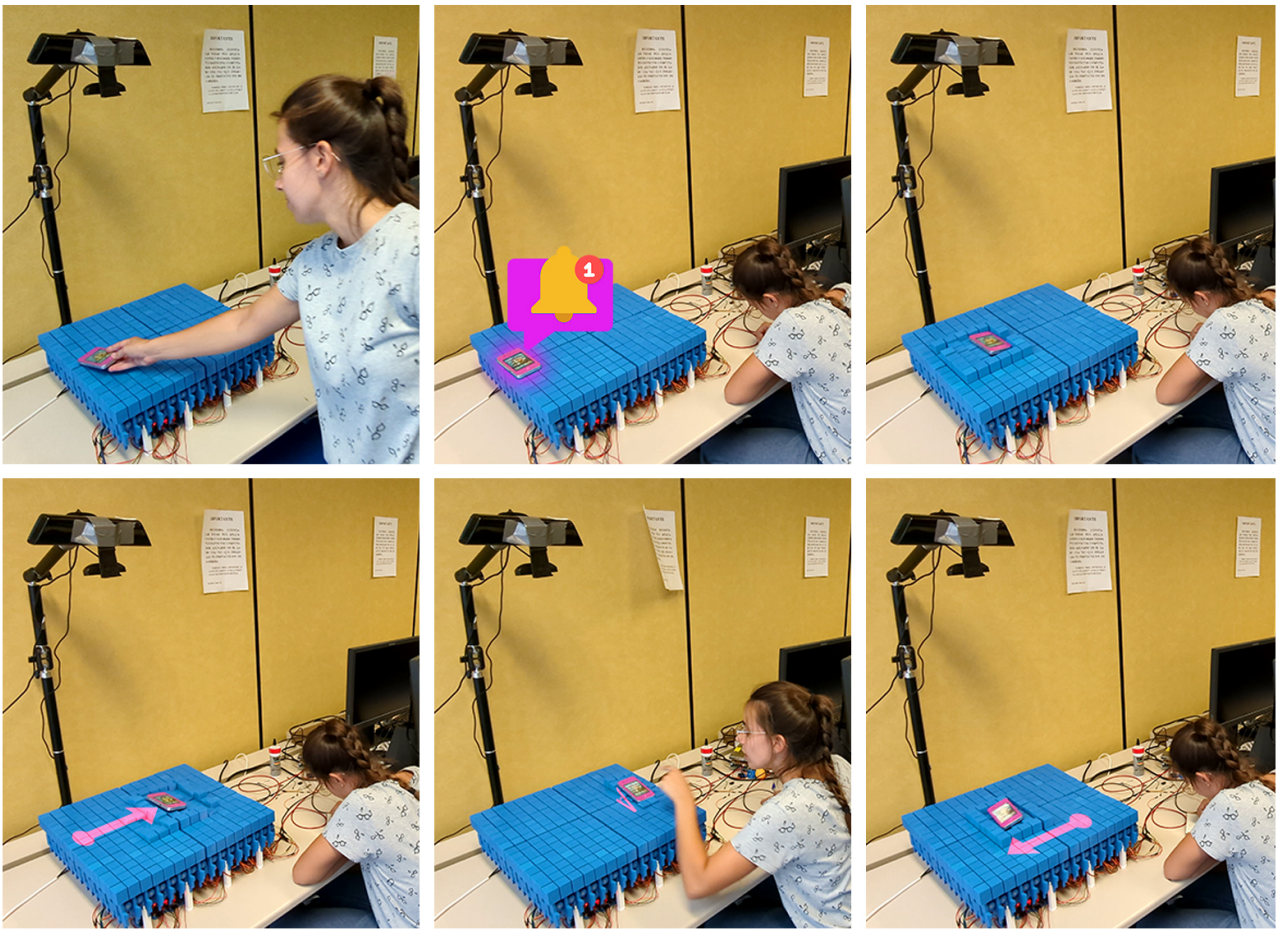


Figure 4.9: Secuencia de imágenes explicativa del desktop potenciado; 1) La persona usuaria deposita su smartphone en un lugar a su elección; 2) Una notificación llega al smartphone; 3) inFORM 16x16 detecta la ubicación del smartphone y cambia su forma para capturararlo; 4) inFORM 16x16 muta progresivamente su forma para desplazar el smartphone cerca de la persona usuaria; 5) inFORM 16x16 inclina el smartphone dejando la pantalla visible y accesible. La persona usuaria puede interactuar con el smartphone; 6) inFORM 16X16 devuelve automáticamente el smartphone a su ubicación original.

Esta aplicación demuestra que:

- inFORM tiene un enorme potencial de aplicación en entornos de escritorio, oficina o taller de trabajo.
- Abre las puertas a una gran variedad de interacciones persona-máquina.
- Puede llegar a ser capaz de distinguir distintos elementos y reaccionar ante su presencia, desplazándolos o rotándolos.

Mecánica y Electrónica



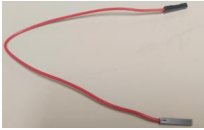


En esta sección se va a explicar toda la información clave relacionada con el diseño, fabricación y montaje del dispositivo inFORM de dimensión 16x16, tanto a nivel mecánico como electrónico.

5.1

BILL OF MATERIALS

A continuación se expone un listado de materias primas y herramientas necesarias para llevar a cabo este proyecto. Este listado tiene como objetivo facilitar la preparación previa del material necesario a aquellas personas que deseen replicar el un inFORM 8x8 o un inFORM 16x16. En secciones posteriores de este documento se detallará el uso que se da a cada elemento.

Foto del componente	Cantidad para 8x8	Cantidad para 16x16	Precio de la unidad (aproximado)	Nombre	Para qué se usa
	1	1	999€	Impresora 3D (Original Prusa i3 MK3S)	Imprimir piezas en 3D
	2	5	17.99€	Filamento PLA 1.75mm (azul)	Como material de impresión
	64	256	2.1€	Micro Servomotor SG90 9g	Mover cada uno de los pistones del inFORM
	128	512	0.1€	Tornillo M2x10mm	Sujetar los servomotores a la estructura que los alberga
	128	512	0.08€	Tuerca M2	Junto con el tornillo M2x10mm, permite sujetar los servomotores a la estructura que los alberga
	1	1	20€	Destornillador	-
	1	1	12€	Aceite de silicona	Como lubricante para poder unir piezas
	4	16	7€	Controlador PWM PCA9685 - 16canales	Controlar los servomotores
	1	1	20€	Soldador	Junto con el estaño, permite soldar los cables, las direcciones I2C en los controladores PWM...
	1	1	9.99€	Estaño	Junto con el soldador, permite soldar los cables, las direcciones I2C en los controladores PWM...
	1	1	8€	Arduino Nano	Como microcontrolador del inFORM

	1	1	3.59€	Cable 2.0 USB A a Mini USB B	Conectar el Arduino Nano con el ordenador
	1	4	23.99€	Fuente de alimentación 5V 30A	Alimentar a los servomotores
	20 (4 negros, 4 rojos, 4 amarillos, 4 verdes, 4 naranjas)	80 (16 negros, 16 rojos, 16 amarillos, 16 verdes, 16 naranjas)	0.12€	Cable Dupont (Hembra-Hembra)	Conectar los controladores PWM en serie
	1	4	8.54€	Cable de alimentación	Conectar las fuentes de alimentación a un enchufe
	4 (2 rollos de cable rojo de diferente grosor y 2 rollos de cable negro de diferente grosor)	4 (2 rollos de cable rojo de diferente grosor y 2 rollos de cable negro de diferente grosor)	16'99€	Cable eléctrico	Conectar las fuentes de alimentación con los controladores PWM

5.2 MECÁNICA

La parte mecánica del dispositivo consta de una serie de piezas: estructura modular de dimensión 2x2, piñón, cremallera, pistón, patas y soporte. La mayoría de estas piezas proceden del proyecto anterior y no se les ha realizado ningún cambio. Sin embargo, algunas de ellas si han sufrido modificaciones. También se han diseñado y fabricado nuevas piezas para cumplir con los requisitos del proyecto. Para el diseño de las nuevas piezas se ha empleado Autodesk Fusion 360. Todas estas piezas se han obtenido mediante impresión 3D, para lo cual se ha empleado la impresora Original Prusa i3 MK3S.

5.2.1 ESTRUCTURA MODULAR DE DIMENSIÓN 2X2

Se trata de una pieza que fue diseñada en el TFG anterior y la cual no ha sufrido ningún cambio durante este proyecto. Esta pieza es la encargada de albergar a los servo motores. Cada estructura es capaz de albergar 4 servo motores y, por tanto, se requieren 16 unidades de esta pieza para montar un módulo de dimensión 8x8. Para fijar el servomotor a la estructura, se emplean dos tornillos M2 10 mm y dos tuercas M2. La estructura cuenta con dos agujeros hexagonales donde deben introducirse las tuercas. Además, cabe destacar que los servo motores se colocan en la estructura en dos niveles: el nivel superior y el nivel inferior. La estructura dispone de dos ganchos para sujetar los cables de los servomotores del nivel superior, de forma que así los cables quedan más ordenados. Ver figura 5.1.

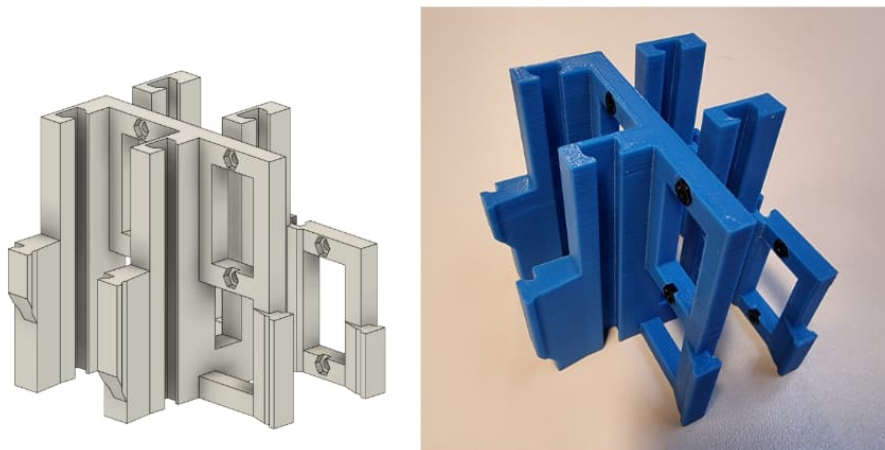


Figure 5.1: Modulo 2x2

Una estructura modular 2x2 puede unirse a otras gracias a las juntas con las que cuenta. Se tratan de juntas de Cola de Milano de 10 mm con una tolerancia de 0.1mm. Dichas juntas permiten a la estructura modular 2x2 escalar tanto en el eje X como en el eje Y. Ver figura 5.2.

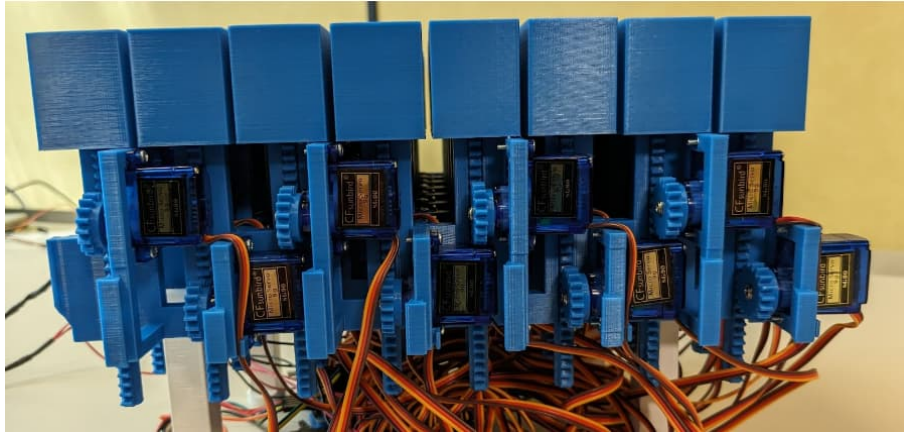


Figure 5.2: Lateral con dos modulos 2x2 ensamblados entre sí. Se observa tambien el sistema de piñones, pistones y engranados.

Tiempo de impresión	Metros de PLA
4h 6m	8,94m

5.2.2

PIÑÓN Y CREMALLERA

El mecanismo de piñón y cremallera es un dispositivo mecánico que permite transformar un movimiento rotatorio en un movimiento lineal. Este dispositivo mecánico cuenta con dos piezas: por un lado, tenemos el piñón, el cual se trata de un engranaje circular y, por otro lado, tenemos la cremallera, que es un engranaje lineal. Ver figura 5.3

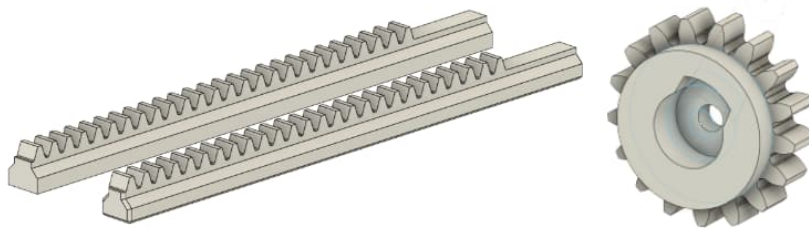


Figure 5.3: Cremallera y piñón.

Ambas piezas fueron diseñadas en el proyecto anterior y no han sufrido ningún cambio durante el actual proyecto. De la cremallera, el parámetro importante de su diseño es que el módulo es 1, mientras que del piñón, los parámetros importantes son que tiene 17 dientes y el módulo es 1. Como bien se puede observar, tanto la cremallera como el piñón, fueron diseñados con el mismo módulo, ya que para que dos engranajes puedan engranar, deben tener el mismo módulo.

Cabe destacar, que el empleo de este mecanismo nos permite transformar el movimiento circular del servomotor en un movimiento lineal y, para ello, será necesario acoplar el piñón al servo motor. Para realizar dicho

acoplamiento, es necesario emplear las “aspas” que vienen con cada uno de los servo motores. Hay que coger una de las mencionadas “aspas”, cortarla dejando sólo 2mm de ella y, posteriormente, introducirla en el agujero que el piñón tiene disponible para introducir el asa. De esta forma, el piñón y el servomotor se pueden acoplar de forma segura. Ver figura 5.4.

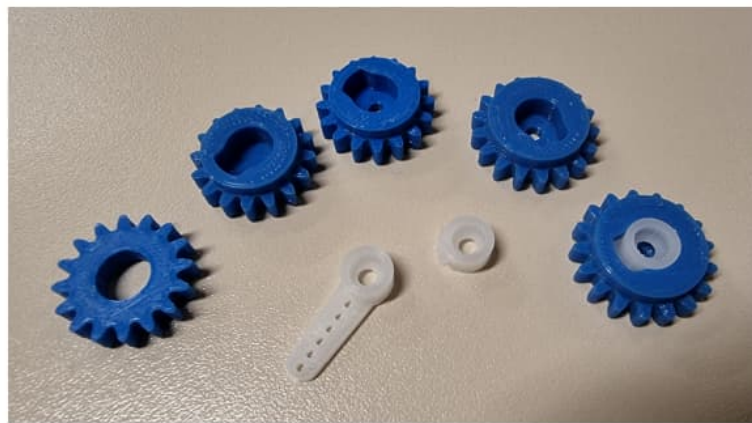
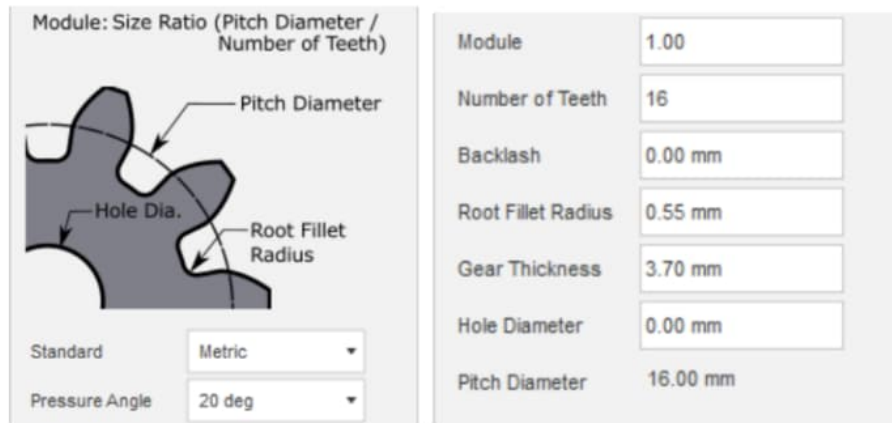


Figure 5.4: Propiedades de piñón y ensamblado con aspa de servo cortada.

Nombre pieza	Tiempo de impresión	Metros de PLA
Piñon	10m	0.39m
Cremallera	20m	0.93m

5.2.3 | PIXEL

El diseño de esta pieza lo obtuvimos del proyecto anterior y no le hemos realizado ningún cambio. Se coloca en la parte superior de la cremallera y es la pieza que actúa en el inFORM. La pieza es un cuboide rectangular que mide 26mm de ancho, 33mm de alto y 26mm de largo. El interior de la pieza está vacío, excepto la columna que permite unir esta pieza con la cremallera. El hecho de que el interior esté vacío permite que el proceso de impresión de la pieza sea más rápido, así como un ahorro en cuanto a la cantidad de filamento empleado. Ver figura 5.5.

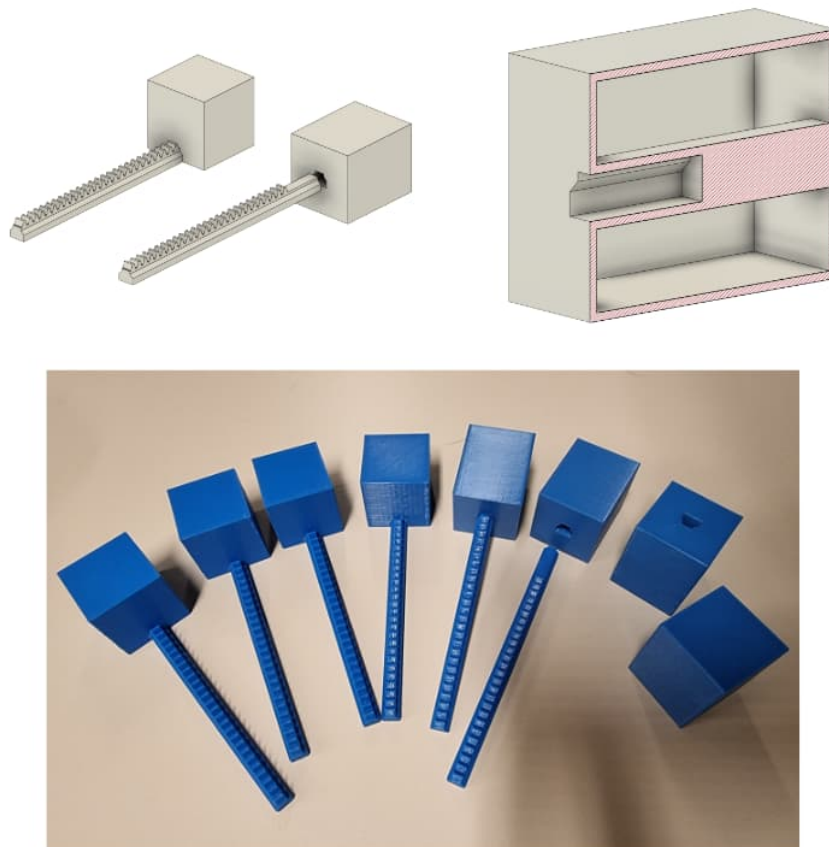


Figure 5.5: Pixel hueco y ensamblado con cremallera.

Tiempo de impresión	Metros de PLA
48m	1.98m

Debido a las cremalleras sobresaliendo por la parte inferior del dispositivo y a la presencia de los controladores PWM y de una gran cantidad de cables, no es viable el uso del dispositivo en horizontal sin estar elevado. Por ello son necesarias las patas, las cuales son las piezas que permiten elevar el dispositivo manteniéndolo en horizontal. Al principio, decidimos emplear las patas verticales diseñadas en el proyecto anterior. Dichas patas miden 9cm de alto y aprovechan las uniones existentes en las estructuras modulares 2x2.



Figure 5.6: Patas originales.

Sin embargo, al tratar de montar el módulo 16x16, nos dimos cuenta de que debíamos diseñar otro modelo de pata que se ajustase mejor a nuestras necesidades. Para montar el módulo 16x16, los 4 módulos 8x8 que lo componen no se unen empleando las uniones existentes en las estructuras modulares, sino que simplemente se colocan uno al lado del otro. Sin embargo, la presencia de las patas verticales diseñadas en el anterior proyecto generaba huecos innecesarios entre los diferentes módulos 8x8, debido a que dichas patas, al aprovechar las uniones de las estructuras, debían colocarse en el borde del módulo 8x8. Ver figura 5.6.

Es por ello que decidimos diseñar otro modelo de pata que evitase la generación de huecos innecesarios entre módulos. Para ello, creímos que la nueva pata no debía acoplarse a la estructura modular por los bordes, como lo hacían las patas verticales del proyecto anterior, sino que debía acoplarse por la parte inferior de la estructura modular. Por lo que diseñamos una pata que mide 8.4cm de alto y que se acopla a la estructura modular por su parte inferior, aprovechando los segmentos en forma de T que contiene la estructura. Ver figura 5.7.

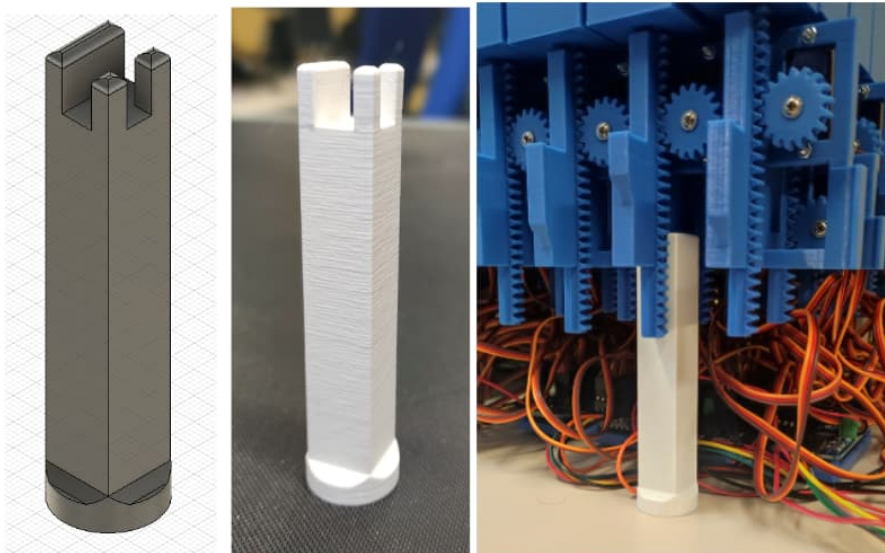


Figure 5.7: Nuevas patas interiores.

Tiempo de impresión	Metros de PLA
51m	3.27m

Esta pieza permite elevar un módulo 8x8 en vertical diagonalmente. No existía en el proyecto anterior, sino que ha sido diseñada para el actual proyecto. La necesidad de que esta pieza exista la genera una de las aplicaciones desarrolladas en este proyecto, más concretamente el minijuego de lógica en vertical. En dicha aplicación, el usuario debe hacer llegar una pelota desde un punto inicial del inFORM hasta un punto final. El punto inicial es el primer pistón, mientras que el final es una pistón aleatorio de la última fila o de la última columna. Es por ello que es necesario que el soporte coloque al dispositivo en vertical, pues de otra forma no es posible que la pelota vaya avanzando. Además, el inFORM también debe ser colocado en diagonal, pues de otra forma, en determinados casos, no sería posible llegar al punto final. El ángulo que el soporte emplea para poner el dispositivo en vertical no debe alcanzar los 90 grados, pues en dicho caso, existiría una alta probabilidad de que la pelota cayera hacia adelante. Para evitar esto, el soporte ha sido diseñado con un ligero ángulo de inclinación. Además, cabe destacar que el soporte aprovecha las uniones de la estructura modular para acoplarse a ella. Ver figura 5.8.

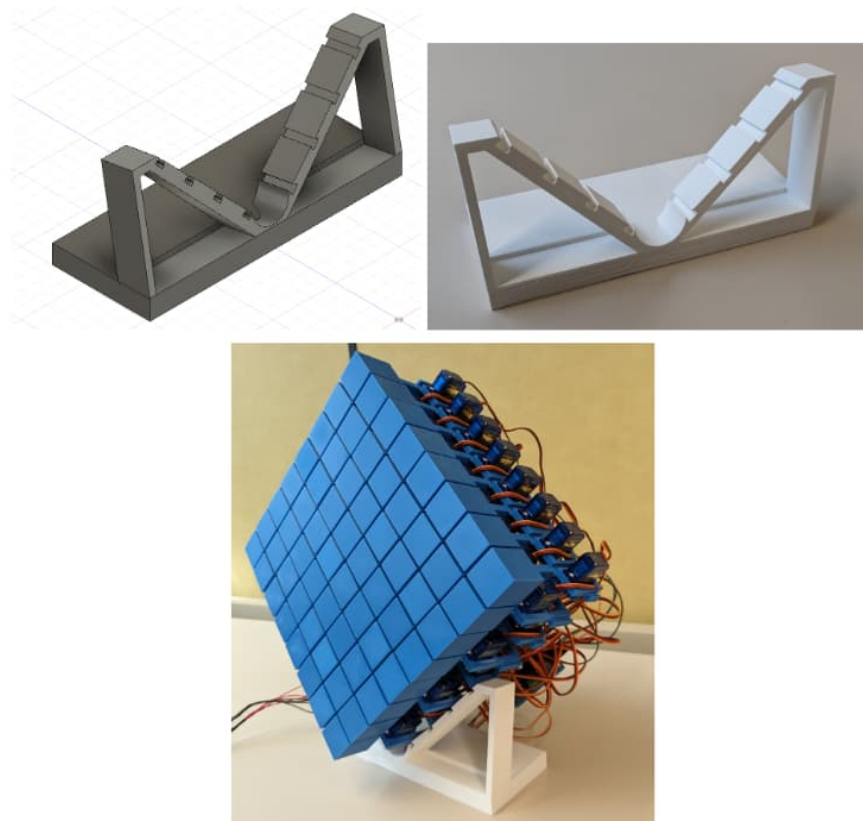


Figure 5.8: Base para ubicar inFORM 8x8 verticalmente en diagonal.

Tiempo de impresión	Metros de PLA
11h 31m	43.32m

5.2.6 RESUMEN DE MATERIAL Y TIEMPOS DE IMPRESIÓN 3D

A continuación, en la tabla, se pueden observar los tiempos de impresión y metros de PLA necesarios para imprimir todas las piezas requeridas en la construcción de un inFORM de dimensión 8x8 y de un inFORM de dimensión 16x16. Cabe destacar que los datos proporcionados no incluyen la impresión de patas o soportes.

Dimensión del inFORM	Tiempo de impresión	Metros de PLA
8x8	148h 48m	352.24 m
16x16	595h 12m	1416.96 m

5.3

ELECTRÓNICA

En esta sección se explicarán los diferentes componentes electrónicos necesarios para que el dispositivo final funcione correctamente.

5.3.1 SERVOMOTOR



Figure 5.9: Micro Servo SG90

Se trata del principal componente en el que se basa nuestro dispositivo. Un servomotor es un motor eléctrico que lleva incorporado un sistema de regulación que puede ser controlado tanto en velocidad como en posición [3]. Existen diversos tipos de servomotor, pero en nuestro caso estamos empleado el Micro Servo SG90 9g [8]. Se trata de un servomotor pequeño y ligero, que tiene un movimiento rotacional de entre 0 y 180 grados (entre 0 y π radianes). Estos servomotores tienen un voltaje de operación de entre 4.8V y 6V, siendo 5V el valor ideal. Al igual que ya se calculó en el TFG anterior, si combinamos el movimiento rotacional con el diámetro de paso de los engranajes, el cual es 17mm, se obtiene que el desplazamiento lineal para cada uno de los pistones es de 26.69mm.

$$\text{Desplazamiento} = \frac{\text{diámetroDePaso} \times \text{movimientoRotacional (en radianes)}}{2}$$

Los servomotores son controlados por medio de una señal PWM (Pulse Width Modulation - Modulación de Ancho de Pulso). La modulación de ancho de pulso es una técnica que modifica el ciclo de trabajo de una señal periódica para transmitir información o para controlar la cantidad de energía enviada [2]. En nuestro

caso, la variación del ciclo de trabajo de la señal nos va a permitir determinar la posición del servomotor. Los servomotores que estamos empleando esperan recibir un pulso cada 20 ms, o lo que es lo mismo, esperan que la frecuencia sea de 50Hz. De esta forma, un pulso de 1.5ms establecerá el servomotor a 0 grados, un pulso de 2ms establecerá el servomotor a 90 grados y un pulso de 1ms establecerá el servomotor a -90 grados.

Estos servomotores se conectan por medio de tres cables:

- Cable Naranja: Señal PWM
- Cable Rojo: Energía (VCC)
- Cable Negro: Tierra (Ground)

5.3.2

ARDUINO NANO

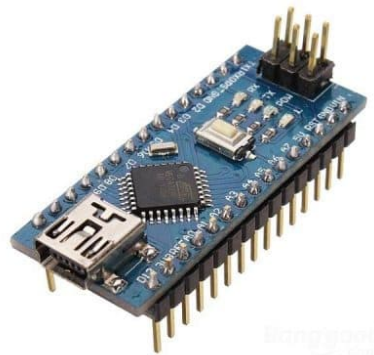


Figure 5.10: Arduino Nano

Es el componente electrónico que hace de microcontrolador. Dicho componente será el que reciba, a través de una conexión serial, la información sobre la altura a la que debe colocarse cada uno de los pistones del INFORM. Posteriormente, enviará dicha información a los servomotores a través de unos controladores PWM.

5.3.3

CONTROLADOR PWM

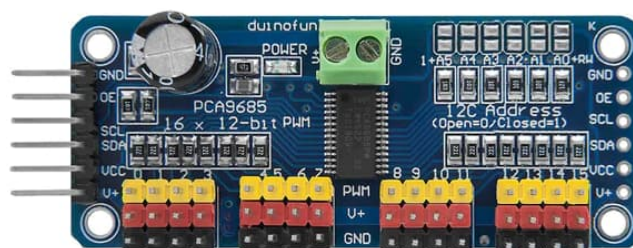


Figure 5.11: Controlador PWM PCA9685

Es el componente electrónico que nos permite controlar los servomotores por medio de una señal PWM. En nuestro caso, estamos empleando el controlador PCA9685 Servo Driver de 16 canales [9], el cual nos permite

controlar individualmente 16 servomotores con 12 bits de resolución. Esto es posible gracias a la presencia de 16 canales, cada uno de los cuales cuenta con 3 pines a los que se conectan los 3 cables de los servomotores: señal PWM, energía (VCC) y tierra (Ground). El controlador PCA9685 cuenta con una bornera para la alimentación de los servos. También cuenta con pines para la alimentación de la parte lógica del controlador: VCC y GND. Además, tiene pines para la transmisión de información en formato I2C: SCL y SDA. Hay otros dos pines más, el OE que permite activar/desactivar todas las salidas rápidamente y el V+ que permite alimentar a los servomotores, aunque este pin no es necesario conectarlo si ya se están alimentando los servos a través de la bornera.

Para conectar un controlador PWM con un Arduino es necesario conectar una serie de pines. A continuación, hay una tabla que resume las conexiones entre los pines del controlador PWM y los pines del Arduino Nano:

Pin Controlador PWM	Pin Arduino Nano
VCC	5V
GND	GND
SDA	A4
SCL	A5

Además, en ocasiones resulta necesario controlar más de 16 servomotores de forma simultánea, algo que se puede realizar encadenando varios controladores PWM. El encadenamiento de controladores PWM permite controlar un mayor número de servomotores empleando un único Arduino Nano. Se pueden encadenar hasta 62 controladores PWM, pudiendo llegar a controlar hasta 992 servomotores con un solo Arduino Nano. Para montar el dispositivo de dimensión 16x16 ha sido necesario encadenar 16 controladores PWM, controlando de esta forma 256 servomotores con un solo Arduino Nano. Para encadenar controladores PWM son necesarias dos cosas. Por un lado, es necesario establecer la dirección I2C de cada uno de los controladores PWM, lo cual nos permitirá usar el mismo bus I2C para enviar información a varios controladores PWM. Para establecer la dirección I2C es necesario soldar los puentes A0, A1, A3, A4 o A5 del controlador. La dirección I2C base es 0x40 y los puentes funcionan como binario, de forma que si soldamos el puente A0 tendremos la dirección 0x41. Para nuestro proyecto hemos usado las siguientes direcciones: 0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F. Por otro lado, para encadenar varios controladores es necesario conectarlos en serie. Para ello, en nuestro proyecto hemos conectado todos los pines excepto el pin V+.

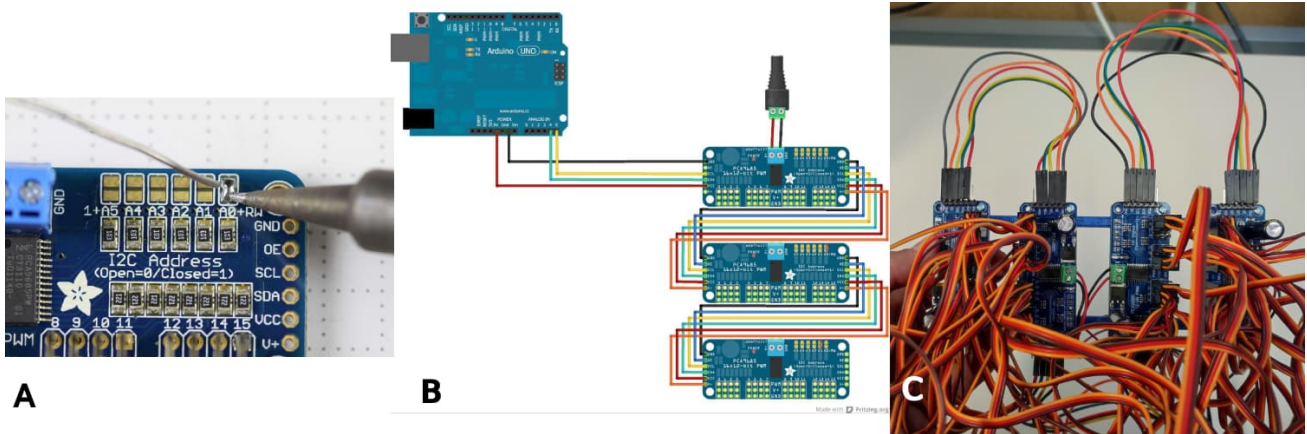


Figure 5.12: A) Soldadura de los puentes de dirección I2C. Imagen obtenida de Adafruit [10] B) Diagrama del encadenamiento en serie de los controladores PWM. Imagen obtenida de Adafruit [10] C) Imagen real del encadenamiento en serie de los controladores PWM

5.3.4

FUENTE DE ALIMENTACIÓN



Figure 5.13: Fuente de alimentación 5V 30A

Es el componente electrónico que se encarga de alimentar a los servomotores para que estos funcionen. Debe conectarse a las borneras de los controladores PWM. Como hemos visto anteriormente, el voltaje de operación ideal de los servomotores empleados en el proyecto es de 5V. Es por ello que la fuente de alimentación empleada debe ser de 5V, con una corriente máxima de 0.4A por servo. Sin embargo, no resulta tan sencillo determinar qué corriente debe tener la fuente de alimentación. En el proyecto anterior, para alimentar todos los servomotores del módulo 8x8 (consumo teórico de 25.6A a 5V) se empleó una fuente de alimentación de 5V y 30A. Para nuestro proyecto necesitábamos saber cuál era la corriente que debía de proporcionar la fuente de alimentación para que todos los servomotores del dispositivo de dimensión 16x16 funcionaran correctamente. Para ello, creamos un script que nos permitía mover todos los servomotores del módulo 8x8 al mismo tiempo. Este script, junto con un multímetro, nos permitió realizar pruebas para averiguar cuál era el consumo de un módulo 8x8 en la situación que más corriente requiere. Las pruebas únicamente se realizaron para un módulo 8x8, puesto que para obtener la corriente necesaria para el dispositivo 16x16 basta con multiplicar la corriente requerida para el módulo 8x8 por 4, pues 4 módulos 8x8 componen un dispositivo 16x16. Tras realizar las pruebas antes mencionadas, llegamos a la conclusión de que lo mejor es emplear una fuente de alimentación

por cada uno de los módulos 8x8. Cada una de las fuentes de alimentación debía de tener las mismas características que la ya empleada en el proyecto anterior. Por tanto, únicamente fue necesario adquirir 3 fuentes de alimentación de 5V y 30A.

VI

Software

En esta sección, por un lado, se introducirán las tecnologías empleadas para desarrollar el software y, por otro lado, se explicará cómo se han desarrollado cada una de las aplicaciones.

6.1

TECNOLOGÍAS EMPLEADAS

Los lenguajes de programación empleados en este proyecto son Arduino(C++), Python y C#. En primer lugar, tenemos el lenguaje de programación Arduino(C++). Este es necesario ya que, como hemos explicado en la sección de “Mecánica y electrónica”, estamos empleando como microcontrolador un Arduino Nano. El lenguaje de programación Arduino comparte similitudes con otros lenguajes como C/C++, aunque es relativamente más simple.

En segundo lugar, tenemos el lenguaje de programación Python, el cual es un lenguaje de alto nivel, que es bastante simple y eficiente, por lo que es adecuado para programadores de todos los niveles. Además, una de las ventajas de Python es que ofrece una gran cantidad de librerías para poder llevar a cabo diversas tareas. Una de las librerías empleadas en el desarrollo de las aplicaciones de este proyecto es OpenCV, la cual resulta muy útil para desarrollar todo lo relacionado con Computer Vision.

En tercer lugar, tenemos el lenguaje de programación C#. Este lenguaje únicamente es empleado en el desarrollo de la aplicación “Interacción remota mediante cámaras de profundidad”, la cual emplea una Kinect para Xbox360. Para desarrollar aplicaciones que empleen la Kinect para Xbox360 se utilizan tres lenguajes: C#, C++ y Visual Basic .NET. De estos tres lenguajes, los más empleados son C# y C++, de entre los cuales yo decidí emplear C#, debido, entre otras cosas, a que ya estaba un poco familiarizada con dicho lenguaje y, además, ofrece una sintaxis más clara y fácil de entender. Cabe destacar que para poder desarrollar aplicaciones que empleen la Kinect para Xbox360 es necesario descargar e instalar el SDK correspondiente proporcionado por Microsoft [7].

Por último, cabe destacar la necesidad de que exista una comunicación de Python y C# con Arduino. Por un lado, Arduino será el encargado de recibir los comandos y emplear dicha información para manejar los servomotores. Por otro lado, serán las aplicaciones desarrolladas en Python o C# las encargadas de crear y enviar dichos comandos a Arduino. Para ello, hemos creado, para cada lenguaje de programación, una clase que nos permita crear los comandos y enviarlos a Arduino. En ambos casos, la clase se llama “ArduinoCommunication”.

INTERACCIÓN REMOTA MEDIANTE CÁMARAS DE PROFUNDIDAD

El "Tangible Media Group" del MIT desarrolló una aplicación de interacción remota mediante cámaras de profundidad. Nosotros nos inspiramos en dicha aplicación para desarrollar nuestra propia versión. Nuestra idea era realizar una aplicación que, gracias al uso de una Kinect para Xbox360, nos permitiera obtener la profundidad a la que se encuentran los objetos para, empleando dicha información, poder representar dichos objetos en el inFORM.

El primer paso para poder desarrollar esta aplicación fue familiarizarnos con la Kinect para Xbox360. Kinect es un dispositivo de detección de movimiento desarrollado por Microsoft e inicialmente pensado para ser empleado con la consola Xbox360, de forma que los usuarios pudieran interactuar con dicha consola empleando gestos y comandos de voz. Dicho dispositivo cuenta con un sensor de profundidad, una cámara RGB, un micrófono, así como un motor para inclinar la propia Kinect. De los componentes mencionados, para desarrollar nuestra aplicación únicamente nos interesaba el sensor de profundidad.

Para poder desarrollar aplicaciones para la Kinect fue necesario instalar el SDK correspondiente. Además, también instalamos el "Developer Toolkit Browser", el cual incorporaba unos cuantos ejemplos de aplicaciones realizadas para la Kinect. De todas esas aplicaciones, hay una llamada "Depth Basics-WPF" que está desarrollada en C# y la cual obtiene la profundidad a la que se encuentran los objetos y los dibuja en la pantalla empleando escala de grises en función de la profundidad a la que se encuentran. Esta aplicación supuso la base para desarrollar nuestra propia aplicación, debido a que, de todos los ejemplos existentes, este era el que más se asemejaba a nuestra idea.

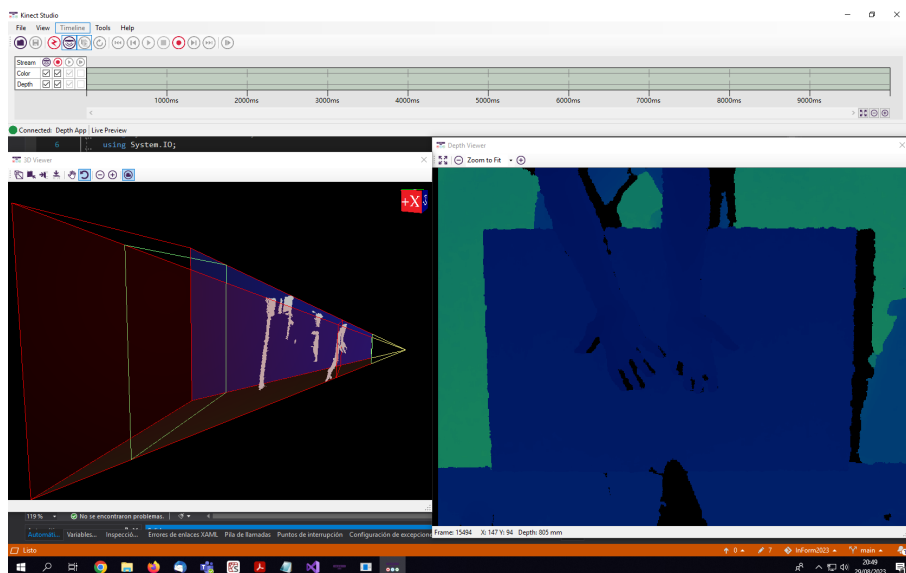


Figure 6.1: Software de asistencia Kinect Studio. Ofrece información en tiempo real sobre el frustum de profundidad capturado por la kinect. Provee información de vital importancia para definir el volumen de interacción.

Una vez que ya teníamos como base una aplicación que nos permitía obtener los datos de profundidad, el siguiente paso era determinar cómo se iban a tratar esos datos o esa imagen captada por la Kinect, pues debíamos de pasar de una resolución de 480x640 (resolución de los frames de la Kinect) a una resolución de 16x16 (resolución del dispositivo inFORM final). Para ello, realizamos los siguientes pasos sobre cada uno de los frames captados por la Kinect (Ver figura 6.2):

1. **Recortar (Crop):** Primero es necesario recortar el frame, puesto que la imagen captada por la Kinect abarca una zona demasiado amplia, lo que nos genera un exceso de información innecesario. Tras recortar el frame nos quedamos con una imagen de resolución 240x240 que incluye los pixels de la zona central del frame obtenido por la Kinect.
2. **Redimensionar:** Tras recortar el frame obtenido por la Kinect, seguimos teniendo una imagen con una resolución demasiado grande si la comparamos con la resolución del inFORM, 240x240 frente a 16x16. Por tanto, para obtener una imagen con la misma resolución que el inFORM, es necesario redimensionar la imagen. Para redimensionar la imagen, lo que hicimos es ir recorriendo conjuntos de pixels de dimensión 15x15 y, para cada uno de esos conjuntos, calcular el valor medio. Ese valor calculado será el valor que se colocará en la posición correspondiente de la imagen resultante de dimensión 16x16.
3. **Establecer mínimo y máximo:** Una vez obtenida una imagen de dimensión 16x16, nos queda un último paso. Esta aplicación está pensada para que se pueda interactuar a distancia, empleando entre otras cosas, las manos. No tiene sentido que el rango de profundidad detectado por la Kinect sea muy amplio, pues eso supondría que el hecho de levantar un dedo de la mano o cualquier movimiento sencillo con la mano no generase un gran cambio de profundidad, de manera que el movimiento no se apreciase en el inFORM. Es por ello que, para establecer el rango de profundidad, se tomaron medidas de las manos de diferentes personas, de forma que así podríamos obtener un rango que nos permitiera que los movimientos sencillos realizados con la mano se apreciaran en el inFORM. Tras realizar todas las mediciones, establecimos que 13cm era una buena distancia entre la profundidad mínima y la máxima. Además, cabe destacar que cuando un objeto se encuentra por debajo de la profundidad mínima se establece como profundidad mínima y si está por encima de la profundidad máxima se establece como profundidad máxima.

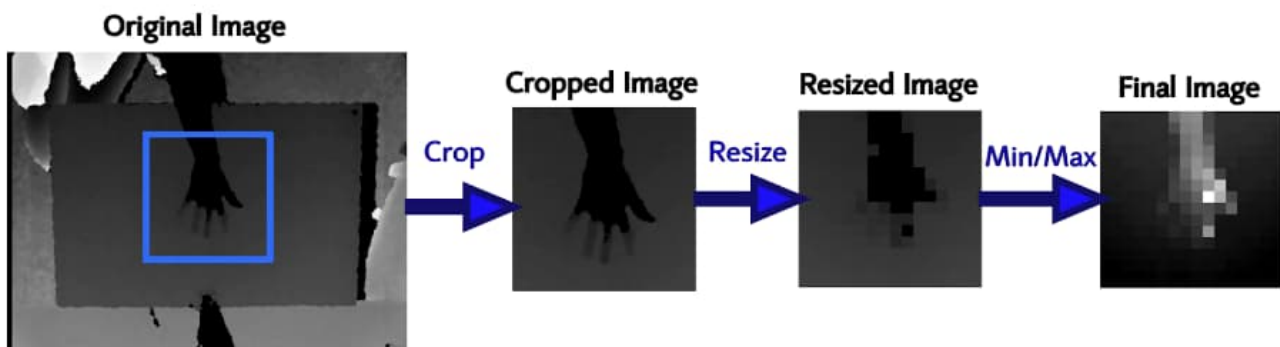


Figure 6.2: Proceso de captura de profundidad y generación de mapa de alturas 16x16.

Además, la aplicación muestra en pantalla la imagen de dimensión 16x16, mapeando los datos de profundidad obtenidos a su valor correspondiente en la escala de grises. Sin embargo, aquí nos surgió un problema,

ya que nosotros queríamos dibujar una imagen de dimensión 16×16 , pero se nos dibujaba con una resolución de 480×640 . Como consecuencia de esto, los datos que nosotros queríamos mostrar en pantalla se dibujaban como pequeños puntos entre los cuales había un degradado. Pero esto no era lo que nosotros queríamos, ya que nuestra idea era que en la pantalla se dibujara una matriz de 16×16 , en la que cada cuadrado representara un dato de profundidad. Para solucionar este problema, en lugar de pintar sólo el pixel al que le corresponde el dato de profundidad, lo que hicimos fue pintar también el conjunto de pixels que se encuentran a su alrededor. El tamaño del conjunto de pixels a pintar por cada datos se calcula viendo el tamaño en el que es necesario agrupar los pixels de la imagen de resolución 480×640 para pasar a una resolución 16×16 y, en este caso, la dimensión de dichos conjuntos debe ser de 30×40 . Ver figura 6.3.

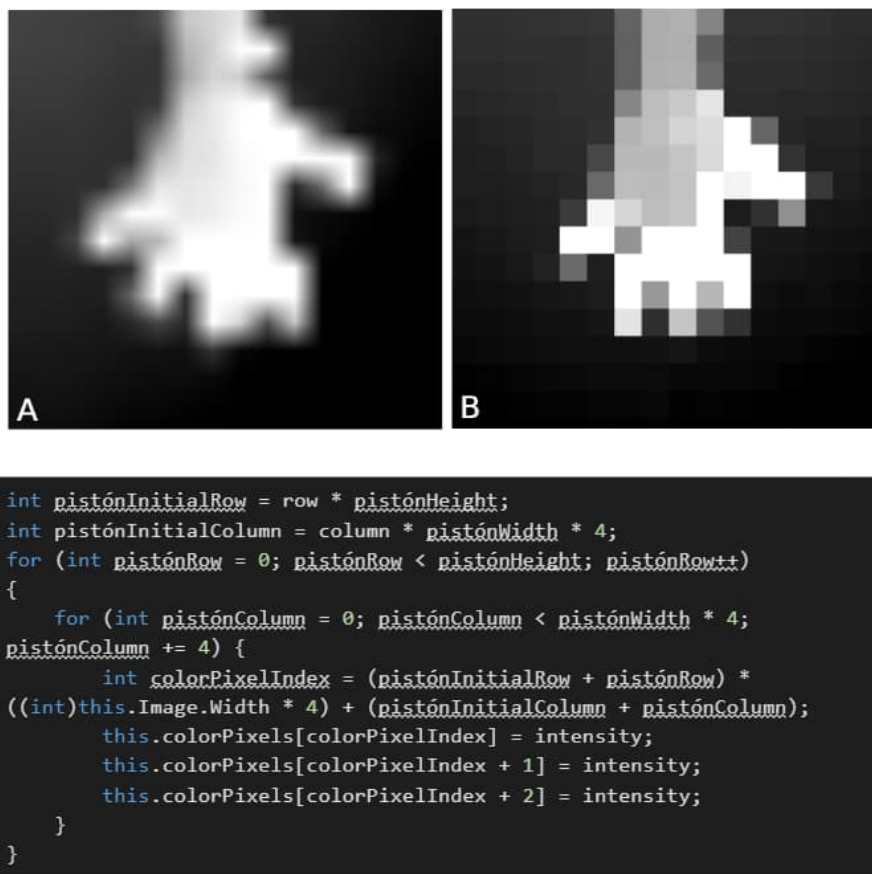


Figure 6.3: Redibujado de pistones para mantener el aspecto pixelizado en pantalla. Se evitan los degradados de grises no discretizables en un inFORM 16×16 .

Una vez que ya fuimos capaces de mostrar la imagen en pantalla y ver que los datos obtenidos eran los correctos, el siguiente paso era enviar la información a Arduino. Sin embargo, al tratar de enviar la información a Arduino fue donde nos surgió otro problema. Nos dimos cuenta de que la información representada en el inFORM no era la correcta. Investigando acerca del problema, llegamos a la conclusión de que la aplicación enviaba la información a Arduino demasiado rápido y este no era capaz de procesarla correctamente. Es por ello que creímos que una buena solución a este problema era introducir una pausa (“sleep”) en la aplicación, de forma que esto ralentizara la velocidad a la que se enviaba la información a Arduino. Esta solución permitió que la información se representase de forma correcta en el inFORM, pero nos bloqueaba la ventana en la que

se debía de dibujar la imagen. Por lo que tuvimos que buscar otra solución. Se nos ocurrió que otra solución podía ser reducir el “Frame rate”. Para ello, lo que hicimos fue quedarnos con 1 de cada 7 frames obtenidos por la Kinect. De esta forma conseguimos que la información se representara correctamente tanto en la pantalla como en el inFORM.

```
if (depthFrame != null && this.frameCounter++ % FRAME_INTERVAL == 0) {
    ... }
```

En la sección “4.2.2 - Resultados - Interacción remota mediante cámaras de profundidad”, puede encontrarse la figura 4.8 representando el resultado obtenido de esta aplicación.

6.3 PERSONALIZACIÓN Y RENDERIZADO 3D DE FUNCIONES MATEMÁTICAS CON FINES EDUCATIVOS

Lo que nos llevó a desarrollar esta aplicación fue nuestro deseo de crear una aplicación que nos permitiera demostrar cómo la visión por computación, en combinación con objetos, permite expandir y personalizar los inputs al dispositivo más allá de la tradicional interacción mediante teclado y/o ratón. La idea inicial era crear una aplicación que, gracias al uso de una cámara, nos permitiera detectar un objeto con el que poder configurar los parámetros de la onda que posteriormente sería renderizada en 3D en el inFORM.

Para desarrollar esta aplicación, tomamos como base una de las aplicaciones resultantes del TFG anterior. Más concretamente, empleamos la aplicación que renderizaba en 3D funciones matemáticas predefinidas en bucle. En nuestra aplicación, no nos interesaba emplear 5 funciones matemáticas diferentes, si no que únicamente nos interesaba renderizar una única onda sinusoidal a la que posteriormente le iríamos variando algunos parámetros. Por lo que ese fue el primer cambio que hicimos.

Después, tuvimos que incorporar visión por computación a nuestra aplicación. Para ello, necesitábamos una cámara y un objeto, el cual sería empleado para configurar los parámetros de la onda. El objeto elegido fue una pelota de color verde. Para detectar dicho objeto en las imágenes obtenidas por la cámara, fue necesario encontrar su umbral de color HSV. También tuvimos que elegir qué parámetros de la onda sinusoidal debía de poder configurar el usuario. Los parámetros elegidos fueron la amplitud, la frecuencia y la dirección del eje x de la onda.

Inicialmente, creímos que era una buena idea emplear la última fila y la última columna del inFORM para establecer los parámetros de la onda desplazando la pelota verde sobre ellas. De esta forma, colocar la pelota en el último pistón del inFORM supondría un cambio de la dirección del eje x de la onda, desplazarla por encima de los restantes pistones de la última columna supondría un cambio en la amplitud de la onda y desplazarla por encima de los restantes pistones de la última fila supondría un cambio en la frecuencia de la onda. Además, cabe destacar que el resto de pistones serían empleados para renderizar la onda. Sin embargo, nos dimos cuenta de que la aplicación, tal y como la habíamos desarrollado, tenía dos desventajas. Por un lado, a la hora de renderizar la onda en 3D se estaban desaprovechando todos los pistones de la última fila y todos los de la última columna. Por otro lado, esta forma de configurar los parámetros de la onda resulta muy poco

intuitiva para un usuario que desconoce la aplicación. Es por ello que decidimos realizar un cambio, introduciendo una “etiqueta” que nos permitiera establecer los parámetros de la onda desplazando la pelota verde sobre ella. Además, la cámara pasaría a capturar la “etiqueta” en lugar de capturar el inFORM. La “etiqueta” es un folio DIN A3 que contiene un gráfico y dos rectángulos. El eje Y del gráfico representa la amplitud en milímetros, mientras que el eje X del gráfico representa la frecuencia en hercios. De esta forma, si se desplaza la pelota verde por encima del gráfico, se puede cambiar tanto la amplitud como la frecuencia de la onda. Por otro lado, los dos rectángulos existentes en la “etiqueta” nos sirven como botones para cambiar la dirección del eje x de la onda, de forma que colocar la pelota encima de ellos será como haber pulsado un botón. Además, cabe destacar que uno de los botones sirve para colocar el eje x de la onda en vertical y el otro sirve para colocarlo en horizontal.

En la sección “4.2.3 - Resultados - Personalización y renderizado 3D de funciones matemáticas con fines educativos”, puede encontrarse la figura 4.6 representando el resultado obtenido de esta aplicación.

6.4

MINIJUEGO DE LÓGICA EN VERTICAL

La idea inicial era crear un juego que nos permitiera mostrar que el inFORM también puede ser empleado en el ámbito del ocio y del entretenimiento. Se nos ocurrió la idea de crear un juego que utilizase el inFORM en vertical y diagonalmente y consistiera en hacer llegar una pelota desde un punto inicial del inFORM hasta un punto final de este. De esta forma, se podría demostrar que el inFORM también es capaz de explotar mecánicas relacionadas con las propiedades físicas del entorno, ya que en este caso estamos aprovechando la fuerza de la gravedad para hacer caer la pelota.

Para desarrollar esta aplicación, tomamos como base la aplicación de dibujado desarrollada en el proyecto anterior. Hubo que hacer algunos cambios en la interfaz gráfica para que la aplicación en lugar de dibujar en pantalla un cuadrado que representara al inFORM, dibujara un rombo, ya que para esta aplicación el inFORM se debe colocar en vertical y diagonalmente. De esta forma, mostrando en pantalla un rombo en lugar de un cuadrado, se contribuye a que la aplicación sea más intuitiva para el usuario. La parte lógica de este juego consta de 3 partes:

1. **Inicio de la partida:** Antes de comenzar una partida, todos los pistones deben estar abajo. Un usuario puede comenzar una partida pulsando la tecla “s” o “S”. En el momento en el que se inicia una partida, todos los pistones suben, excepto el pistón correspondiente al punto inicial y el correspondiente al punto final. Al comenzar la partida, el punto final es elegido de forma aleatoria y corresponde a un pistón de la última fila o de la última columna del inFORM. El punto inicial es siempre el mismo y se corresponde con el primer pistón del inFORM.
2. **Partida:** Durante la partida, el usuario debe seleccionar en la pantalla cuáles son los pistones a bajar para conseguir que la pelota llegue al punto final. Un pistón bajado no se puede volver a subir durante la partida. Cabe destacar que el cambio realizado en la interfaz gráfica de la aplicación hizo que el proceso de detección del pistón seleccionado por el usuario se complicase. Cuando se tiene un cuadrado, conocer el pistón que se está seleccionando es bastante trivial, ya que, para obtener la fila únicamente hay que dividir la coordenada Y del punto seleccionado entre el alto de cada pistón, siendo la fila la

parte entera del resultado y, para obtener la columna sería lo mismo pero empleando la coordenada X y el ancho de cada pistón. Sin embargo, no resulta tan trivial detectar el pistón que se está seleccionando si se emplea un rombo para dibujar el inFORM en la pantalla. En nuestro caso, cada vez que el usuario realiza una selección en la pantalla, es necesario recorrer los pistones hasta que se encuentra el seleccionado o hasta que se han recorrido todos. Además, para comprobar si un pistón está seleccionado o no, es necesario comprobar si el punto seleccionado por el usuario está dentro de alguno de los dos triángulos que forman el rombo.

3. **Fin de la partida:** El usuario puede decidir finalizar la partida en cualquier momento y, para ello, debe pulsar la tecla "F" o "F". En el momento en el que se finaliza una partida, todos los pistones bajan.

En la sección "4.2.4 - Resultados - Minijuego de lógica en vertical", puede encontrarse la figura 4.6 representando el resultado obtenido de esta aplicación.

6.5 POTENCIADOR DE UN ENTORNO DESKTOP MEDIANTE COMPUTER VISION

La idea inicial era crear una aplicación que, mediante el uso de una cámara, detectara un objeto en la superficie del inFORM y lo desplazara gracias al movimiento de los pistones. Para comenzar a desarrollar la aplicación era necesario decidir qué objeto se iba a desplazar por la superficie del inFORM. Decidimos que el objeto a desplazar iba a ser un teléfono móvil.

En primer lugar, la aplicación debía de ser capaz de detectar el teléfono móvil a desplazar, así como las celdas del inFORM ocupadas por dicho dispositivo. Para detectar el teléfono móvil en las imágenes obtenidas por la cámara, fue necesario encontrar su umbral de color HSV (Ver figura 6.4), así como definir una zona de interés en las imágenes captadas por la cámara, de forma que pudiéramos centrarnos únicamente en la parte de las imágenes que nos interesaba.

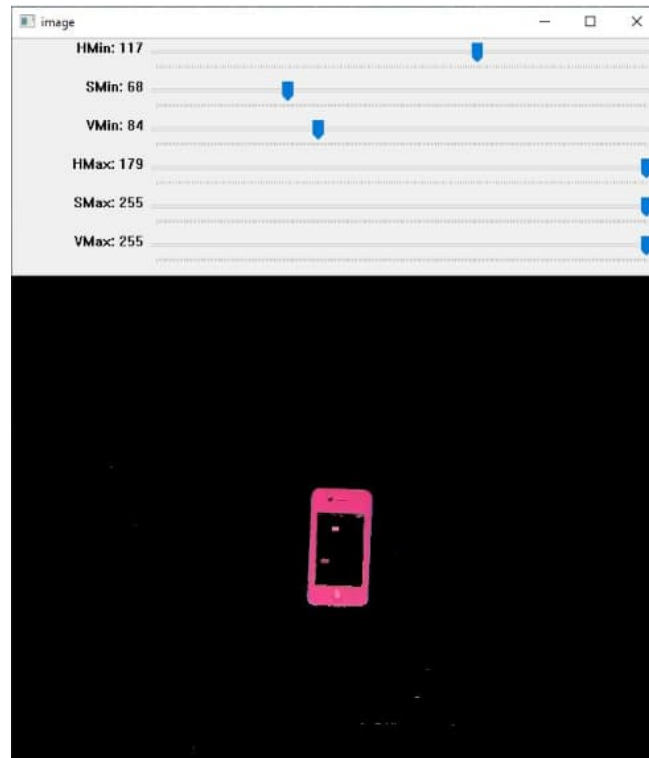


Figure 6.4: Aplicacion propia auxiliar que nos facilita encontrar el los valores de umbralización HSV óptimos.

Una vez detectado el dispositivo, fue necesario procesar las imágenes de la zona de interés para averiguar las celdas del inFORM sobre las que estaba colocado el teléfono móvil. Como la zona de interés únicamente abarcaba el inFORM 16x16, podíamos saber que coordenadas X e Y de la imagen correspondían a cada una de las celdas. Además, al haber detectado el teléfono móvil en las imágenes, conocíamos las coordenadas X e Y de sus esquinas. Por tanto, con esta información se podía averiguar de forma sencilla las celdas del inFORM ocupadas por el dispositivo móvil.

```

1 def updateOccupiedCells(xPixelSize, yPixelSize, rectPoints):
2     xRectCoordinates, yRectCoordinates = zip(*rectPoints)
3     xMinRect = min(xRectCoordinates)
4     xMaxRect = max(xRectCoordinates)
5     yMinRect = min(yRectCoordinates)
6     yMaxRect = max(yRectCoordinates)
7     for x in range(0, N_COLUMNS):
8         for y in range(0, N_ROWS):
9             xMinCell = x * xPixelSize
10            xMaxCell = xMinCell + xPixelSize
11            yMinCell = y * yPixelSize
12            yMaxCell = yMinCell + yPixelSize
13            fullyOccupied = xMinRect <= xMinCell and xMaxRect >= xMaxCell and
14            yMinRect <= yMinCell and yMaxRect >= yMaxCell
15            topBottomCenterOccupied = (yMinCell < yMaxRect <= yMaxCell or
16            yMinCell <= yMinRect < yMaxCell) and xMinRect <= xMinCell and
17            xMaxRect >= xMaxCell

```

```
15     rightLeftCenterOccupied = (xMinCell < xMaxRect <= xMaxCell or
xMinCell <= xMinRect < xMaxCell) and yMinRect <= yMinCell and
yMaxRect >= yMaxCell
16     if xMinRect == xMaxRect or yMinRect == xMaxRect:
17         rectCornerInsideCell = False
18     else:
19         rectCornerInsideCell = (xMinCell < xMinRect < xMaxCell or
xMinCell < xMaxRect < xMaxCell) and (yMinCell <= yMinRect < yMaxCell
or yMinCell < yMaxRect <= yMaxCell)
20     partiallyOccupied = topBottomCenterOccupied or
rightLeftCenterOccupied or rectCornerInsideCell
21     if fullyOccupied or partiallyOccupied:
22         occupiedCells[x,y] = True
23     else:
24         occupiedCells[x,y] = False
```

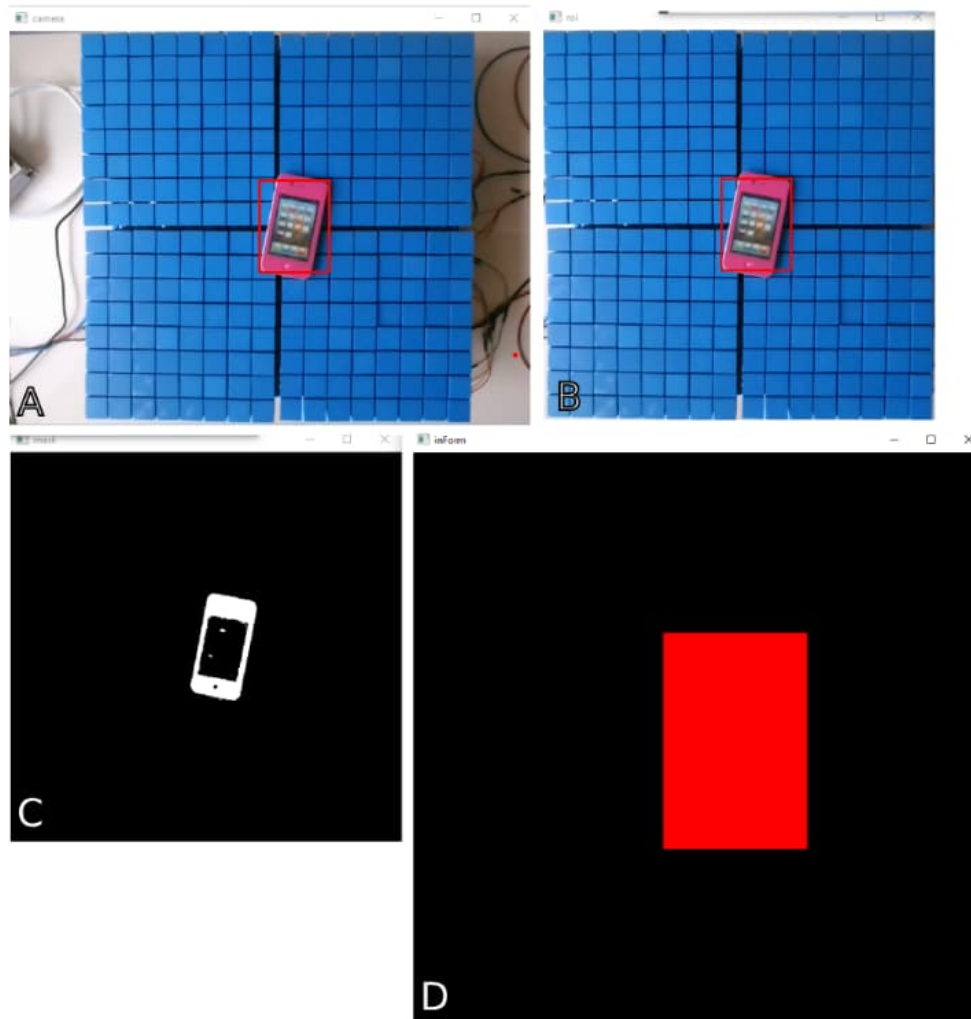


Figure 6.5: Proceso de reconocimiento y localización del smartphone mediante Computer Vision. Todas estas ventanas se visualizan en tiempo real en el PC para depurar su correcto funcionamiento. A) Feed directo de la webcam. El usuario puede clicar en 4 puntos de la imagen para definir una región de interés que posteriormente será empleada para corregir perspectiva; B) Región de interés sobre la que se está trabajando. Un rectángulo rojo indica la detección del smartphone; C) Umbralización HSV; D) Pistones del inFORM sobre los que se ha detectado la presencia del smartphone.

Una vez conocida la ubicación del teléfono móvil, ya sólo nos quedaba desplazarlo. El desplazamiento del teléfono móvil consta de los siguientes pasos:

1. **Iniciar desplazamiento:** Cuando se inicia un desplazamiento, la aplicación debe obtener el punto de origen y el punto de destino. Cabe destacar que los desplazamientos se van a basar en el último pistón ocupado por el teléfono móvil, es decir, el pistón inferior derecho de los ocupados. De esta forma, el punto de origen será el último pistón ocupado por el teléfono móvil en el momento en el que se inicia el desplazamiento, mientras que el punto de destino será el punto al que debemos llevar la parte inferior derecha del dispositivo móvil.
2. **Desplazar al punto de destino:** Para realizar el desplazamiento del teléfono móvil hasta el punto destino, la aplicación lo que hace es comprobar cual es la última celda ocupada por el teléfono móvil

y compararla con la celda correspondiente al punto destino, para determinar si es necesario continuar desplazando el dispositivo y, en el caso de que sea necesario continuar el desplazamiento, determinar la dirección hacia la que hay que mover el dispositivo.

3. **Inclinar:** Una vez que el teléfono móvil ha alcanzado el punto de destino, la aplicación cesa su desplazamiento y lo inclina durante unos segundos. El objetivo de inclinar el dispositivo es que el usuario vea el teléfono móvil y sea capaz de interactuar con él.
4. **Devolver al punto de origen:** Cuando el teléfono móvil deja de estar inclinado, la aplicación lo devuelve al punto de origen. El desplazamiento del teléfono móvil al punto de origen se realiza de la misma manera que el desplazamiento al punto de destino, lo único que cambia es el punto al que debe llegar el dispositivo móvil.
5. **Finalizar desplazamiento:** El desplazamiento del teléfono móvil finaliza cuando este llega de nuevo al punto de origen.

En la sección "4.2.4 - Resultados - Potenciador de un entorno Desktop mediante Computer Vision", puede encontrarse la figura 4.9 representando el resultado obtenido de esta aplicación.

VII

Consejos para makers valientes

En esta sección se pueden encontrar una serie de consejos útiles para aquellas personas que decidan aventurarse en la construcción de un inFORM. La lista de consejos es la siguiente:

- Antes de imprimir una gran cantidad de piezas 3D de un mismo modelo, es recomendable realizar una impresión de una única pieza para asegurarse de que esta se imprime correctamente con la configuración empleada para la impresión. Así, en caso de que la impresión no se haya realizado correctamente, la cantidad de material gastado será menor.
- La modificación de las tolerancias de las impresiones 3D puede provocar que las piezas no encajen, que queden demasiado ajustadas o que queden demasiado flojas. Por lo que es recomendable tener mucho cuidado a la hora de modificar las tolerancias.
- La altura o debe ser la misma para todos los pistones. Para ello, es necesario que a la hora de montar los módulos, los servomotores estén en la posición correspondiente a la altura o de los pistones. Además, también será necesario colocar los pistones lo más abajo posible.
- El máximo largo de pulso es el que permite colocar a los servomotores en la posición correspondiente a la altura o de los pistones. Es recomendable que el máximo largo de pulso empleado para el funcionamiento del inFORM sea menor que el empleado para el montaje de los módulos. Si se emplea el mismo largo de pulso en ambos casos, algunos pistones podrían quedar demasiado justos en el nivel o, lo que provocaría que los servomotores nunca llegasen a la posición correspondiente al máximo largo de pulso. De esta forma, los servomotores estarían continuamente tratando de llegar a la posición correspondiente al máximo largo de pulso, por lo que al final acabarían quemándose. Ocurriría lo mismo si el largo de pulso empleado para el funcionamiento del inFORM fuera mayor que el empleado para el montaje de los módulos.
- A la hora de unir piezas, especialmente si lo que se van a unir son módulos 2x2 o conjuntos de módulos 2x2, es bastante recomendable emplear algún lubricante que facilite la unión de las piezas. En ocasiones, si no se emplea lubricante y las piezas a unir están demasiado ajustadas, podrían llegar a atascarse o romperse.
- Las fuentes de alimentación que van a ser empleadas para alimentar a los servomotores es mejor probarlas primero con un único componente para posteriormente avanzar de forma incremental. De esta forma, en el caso de que una fuente de alimentación no funcione correctamente, la cantidad de componentes electrónicos estropeados será menor que si directamente se conecta la fuente con todos los componentes electrónicos a los que va a alimentar.

VIII

Trabajo Futuro

Mejoras en el dispositivo inFORM

El dispositivo inFORM puede ser mejorado en varios aspectos. En primer lugar, se podría tratar de aumentar la resolución del inFORM 16x16 construyendo e incorporando más módulos. Un aumento de la resolución permite aumentar la claridad, el detalle, el realismo y la precisión del contenido representado en el inFORM. En este caso, conviene reevaluar los medios y protocolos de comunicación para mantener un framerate usable. Más pistones implicaría mayor flujo de información.

También se podría tratar de reducir el tamaño de los pistones, lo que aumentaría la densidad de información en el inFORM. En este caso habría que reorganizar los servos para juntarlos todavía más, hacer uso de servos más pequeños, o aproximarnos a la idea del inFORM original utilizando cables Push-Pull.

Se podría aumentar el movimiento vertical de los pistones empleando unos servomotores mejores que tengan un mayor ángulo de rotación, o empleando un sistema de piñones asimétrico que amplíe el movimiento lineal resultante.

Además, se podría rediseñar la estructura que soporta los servomotores, creando un diseño que permita reparar o cambiar los servo motores sin necesidad de desmontar el inFORM. También se podría diseñar una estructura más sencilla o fácil de ensamblar y desensamblar.

Interfaz de interacción Web

Se podría emplear un ESP32 para crear un hotspot wifi y servir en él una página web que permita interactuar con el inFORM de la misma forma que lo hacen las aplicaciones desarrolladas, las cuales se ejecutan desde un ordenador. De esta forma un usuario podría conectarse desde su móvil o tablet al hotspot wifi, visualizar la página web e interactuar con el inFORM sin la necesidad de un ordenador. Esta es una idea especialmente interesante de cara a utilizar el inFORM 16x16 en un museo o demostración.

Proyector

Se podría colocar un proyector encima del inFORM, así como material reflectante en la parte superior de los pistones, para proyectar contenido en la superficie del inFORM. El empleo de un proyector podría resultar muy útil en la aplicación de interacción remota mediante cámaras de profundidad, ya que se podría dibujar en la superficie del inFORM la imagen detectada con la cámara RGB que posee la Kinect.

Sensórica

Se podría añadir a los pistones del inFORM sensórica que permita detectar cuando un pistón está siendo tocado, de forma que se pueda interactuar con el inFORM tocando directamente los pistones. Podría realizarse mediante computer visión, o empleando una matriz de sensores capacitivos que reaccionasen al tacto.

Estudios de usuario

Se podrían realizar estudios de usuario evaluando la capacidad del display de renderizar información sobre formas geométricas. Podría medirse con que frecuencia las personas usuarias son capaces de distinguir correctamente la figura geométrica renderizada. Esto ayudaría a definir qué tipo de renderizados tangibles serían válidos de cara a extender experiencias VR.

Framerate

En alguna de las aplicaciones desarrolladas, tuvimos que reducir el framerate para enviar la información más despacio a Arduino, ya que si no este no era capaz de procesarla correctamente. Sugerimos realizar un análisis de flujo de información en las comunicaciones para dar con el cuello de botella y buscar una solución que permita aumentar el framerate de ejecución. Hipotetizamos que utilizar un arduino nano por cada módulo 8x8 podría aliviar este problema de comunicación. También sugerimos que parte del calculado del mapa de alturas lo realice el propio arduino de manera que, por ejemplo en lugar de enviar información de los 256 pistones para renderizar una onda sinusoidal, podría enviarse simplemente información sobre su frecuencia y amplitud, encargándose el propio arduino de calcular las alturas de los pistones.

Tutorial

Por último, siguiendo el espíritu maker de este proyecto, nos encantaría realizar un tutorial guiado mediante texto y video sobre cómo construir nuestra réplica de inFORM y publicarlo en <https://www.instructables.com/member/UpnaLab/>

Bibliography

- [1] Aug. 2021. URL: https://es.wikipedia.org/wiki/MIT_Media_Lab.
- [2] Oct. 2022. URL: https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_ancho_de_pulsos.
- [3] July 2023. URL: <https://es.wikipedia.org/wiki/Servomotor>.
- [4] URL: <https://upnalab.com/>.
- [5] URL: <https://tangible.media.mit.edu/contact-admission/>.
- [6] URL: <https://tangible.media.mit.edu/project/inform/>.
- [7] URL: <https://www.microsoft.com/en-us/download/details.aspx?id=40278>.
- [8] URL: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf.
- [9] URL: <https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>.
- [10] Bill EARL. *Adafruit PCA9685 16-channel servo driver*. URL: <https://learn.adafruit.com/16-channel-pwm-servo-driver/chaining-drivers>.
- [11] Sean FOLLMER et al. “inFORM: dynamic physical affordances and constraints through shape and object actuation.” In: *Uist*. Vol. 13. 10. Citeseer. 2013, pp. 2501–988.
- [12] Daniel LEITHINGER et al. “Shape displays: Spatial interaction with dynamic physical form”. In: *IEEE computer graphics and applications* 35.5 (2015), pp. 5–11.
- [13] Gaurav SINHA, Rahul SHAHI, and Mani SHANKAR. “Human computer interaction”. In: *2010 3rd International Conference on Emerging Trends in Engineering and Technology*. IEEE. 2010, pp. 1–4.
- [14] Yuxiang XU. *Budget inform, a dynamic shape display*. Jan. 2021. URL: <https://academica-e.unavarra.es/handle/2454/44307>.