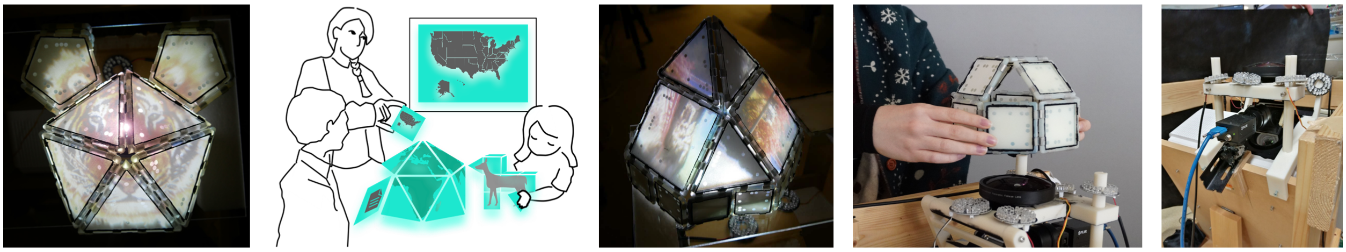
International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2022)
J.-M. Normand and H. Uchiyama (Editors)

OmniTiles - A User-Customizable Display Using An Omni-Directional Camera Projector System

Jana Hoffard[1], Shio Miyafuji[1], Jefferson Pardomuan[1], Toshiki Sato[2] and Hideki Koike[1]

[1]Tokyo Institute of Technology, Japan
[2]Japan Advanced Institute of Science and Technology, Japan



(a) (b) (c) (d) (e)

Figure 1: *(a) Example of polygonal shape creation using OmniTiles. (b) Concept design of our system stationed around a fixed place like a dining table. (c) Tiles displaying an image designated to their ID. (d) Unlit structure held by the user. (e) View onto camera and half mirror.*

Abstract

We present OmniTiles, a manually changeable interface that enables the user to customize their own display. This is achieved by using tiles in basic shapes that are clipped together via magnets. The created structures are then placed on top of a camera-projector set up to track the single tiles and project onto them. The generation of different structures requires no activation mechanism or prior technical knowledge by the user. The 3D printed tiles are robust and cost-efficient, making the system particularly suited for non-experts such as families with children. First, we explain the creation process of our tiles and the implementation of the system. We then demonstrate the flexibility of our system via applications unique to our tile approach and discuss the limitations and future plans for our system.

CCS Concepts

• ***Human Computer Interaction*** → ***Interactive surfaces;*** • ***Projection Display*** → *Omni-Directional System; non intrusive markers;*

1. Introduction

Most people are accustomed to using two-dimensional displays such as smartphones or tablets. While these have the advantage of easily fitting inside pockets and are therefore very portable, these displays also come with several disadvantages: Due to the lack of depth, optical illusions can easily occur and there is little to no haptic feedback for the user.

To compensate for these shortcomings, several three dimensional (3D) and shape changing displays with varying focuses [RPPH12] were introduced over the years. While some concentrate on changing-mechanisms for 3D structures mimicking virtual objects which are then used to support the immersion of virtual reality (VR) environments, others try new ways of creating deformable displays. However, these systems typically provide only constrained ways of changing such as extending, moving along an axis or inflating to alter their size. These mechanisms typically require elaborate activation mechanisms and often have to be redesigned for every new application. This does not allow end-users without technical background to create or design their own display without prior training. Furthermore, the activation mechanisms require responsible handling and are often expensive in set up or maintenance making them unattractive for end-users and usually targeted towards adults only.

We therefore introduce our OmniTiles prototype, a user-

DOI: 10.2312/egve.20221286

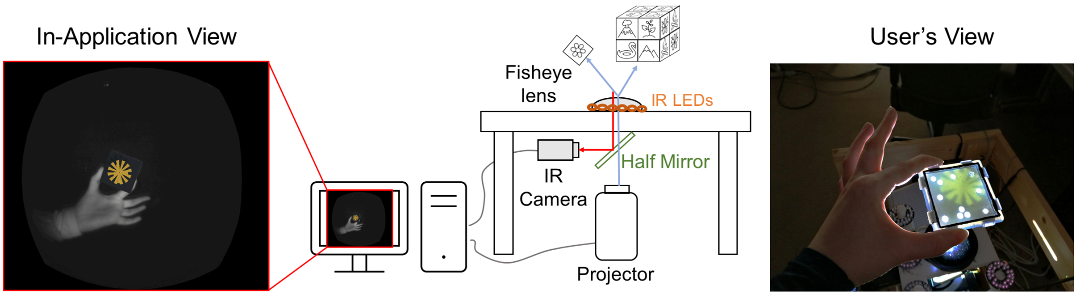


Figure 2: *System overview of the axis aligned projector-camera setup, the user's view and the in-application view.*

customizable display which utilizes tiles in basic polygonal shapes that can be clipped together to create more complex structures. We created triangles, squares, and pentagons of the same side length and bigger triangles and squares of a doubled side length. These tiles are inspired by tile blocks like Magformers[†] that children play with. To display data onto the tiles we use a camera-projector setup underneath a wide angle fisheye lens which allows for simultaneous tracking and projection.

Our design enables multiple users to interact with the system concurrently, allowing them to share information among each other without any constraints. Further, the simple magnet-based clipping mechanism lets the user create and alter the shape of the display without requiring any knowledge of 3D software, electronics or mechanical construction. The structures and hence shapes of the display are only limited by the occlusion-free capturing space of the camera, therefore enabling changes in width and height as well as rotation and shape. The structures created out of the tiles can range from simple shapes like cubes or bowl-like structures to more abstract ones. Multiple users can design and compose a display together supporting each other. The single tiles are 3D printed which makes them sturdy and cost-efficient in creation and maintenance. Our system can be used relentlessly and single components can be produced cost-efficient making it interesting for families with young children.

In summary, our contributions are as follows:

1. Conceptual approach of a user customizable interface for an augmented reality projection space.
2. Basic tiles allow for low cost production and rapid creation of new structures without requiring any technical knowledge.
3. Increased capability in terms of changeable parameters as well as the structure of the display itself
4. Discussion of different use cases and applications as well as limitations and future extensions of the system

2. Related Work

Our related work covers several fields. While we considered 3D and shape-changing displays, and omni-directional projector systems in particular, we also conducted research on suitable tracking and production methods for our tiles.

2.1. Three-Dimensional and Shape Changing Displays

First, we took an extended look into different 3D displays including augmented and virtual reality [ISS*15, OMS18, JYBK20] and different shape changing displays [YOTI14, SIF*20, SII*20], some of which already tried to utilize multiple screens or tiles for their information display [CLW*14, LK12, PM12, RUO01].

Though most people are used to flat 2D displays like smartphones and tablets, there are certain limitations to these displays. For instance, the lack of a third dimension can cause optical illusions and makes it impossible to tell shapes from the surface alone. Everitt et al. [EA17] therefore created a semi-solid surface which can be deformed and lifted by utilizing the ShapeClip modules introduced by Hardy et al. [HWT*15] which can be extended along their height.

Other research concentrates on the overall idea of clipping single modules together. This is either done to extend or change a surface as the user sees fit [RKLS13, TOA*16, LHL*21] or to mimic a physical structure [LGB*16, NULI17]. Tiab et al. Tiltstacks [TBS*18], for example, can change their length along an axis and bend to one side which allows for several surface structures when multiple Tiltstacks are combined. Lu et al. [LHL*21], on the other hand, uses cube shaped ID building blocks which can be stacked on top of each other while keeping track of the single block's ID. This allows one to save information such as position of a character in a game on a single blocks which can then be moved around with said character.

Similar to this idea, Goguey et al. introduced PickCells [GSL*19], small cubic cells that can be arranged by the user. The cells can be used to display information individually or over their connected structure. The use of small touch screens provides good image quality and interaction but makes the system more fragile and expensive.

2.2. Omni-Directional Projection Systems

For tracking and projecting onto our tiles, we further looked into systems which allow for the capture and/or tracking of an extended area, not limited to the front of a standard webcam. Miyashita et al. [MYU*18] for instance, introduced a portable system that allows it to track and project into the environment simultaneously. A camera and projector are attached to the user allowing them to move around

† https://magformers.com/

freely, though, the tracking range is therefore limited to the area in front of the user.

Benko et al. [BW10] constructs a tilted geodesic dome with a onmi-directional projector-camera setup in it. Several users can enter the dome simultaneous and interact with the displayed data via gesture recognition and speech commands. In Maeda et al.'s research [MPSK18] on the other hand, they created a fixed setup consisting of a 360 degree camera and a projector with a wide-angle fisheye lens placed below a table allowing for an increased tracking and projection space in which markers could be detected. Takagi et al. [TSMK21] altered this system to have the camera and the projector axis aligned by using an infrared camera instead of the 360 degree color camera together with a half mirror below the table. Both of these systems used markers for their interactions.

We chose this latter system for our prototype as we think of our system as a support stationed around a sitting place like a desk or a dining table. Therefore, the increased projection and tracking space around the camera and projector is more desirable over the increased mobility of Miyashita et al.'s work [MYU*18]. Additionally, the use of infrared light in the omni-directional system aids in filtering any background noise from our markers.

2.3. Tracking

Over the years, several camera based tracking methods have been introduced. From different marker based tracking systems [PP04, HLW*18] over emitter based tracking systems [WSF18] to gesture input [SKPP19]. For markers, research typically tries to balance between the visibility of the markers for the camera and the user as well as occlusion handling. Markers which are visible to the human eye could distract the user and disturb projection space. Takagi et al. [TSMK21] for instance, uses markers made from retro-reflective material in their system. Since the material is made from a light color, it can still be projected onto without the markers disturbing the image. At the same time, the use of infrared light enables the system to easily track the markers. Tone et al. [TIHS20] used infrared light channeled through a 3D printed objected to recognize and project onto said object. The infrared light emitters and the channels are hidden within the object. The blinking rate of the light is used to track and trigger events unnoticed by the user.

For robust occlusion handling, Narita et al. [NWI15, NWI16] uses a matrix of dots for projection onto a deformable surface. The known matrix enables strong occlusion handling which allows for the calculation of the overall surface shape from the distance of the dots to each other as well as occluded parts. We found that this approach of dot based markers held several advantages for our system, which is why we based our tracking approach on it.

2.4. Systems for Children

Systems targeted towards children require an extra work in the design process to make them robust enough for rigorous interactions. For this reason all of the previously introduced research is targeted towards older children and adults as their single components are too fragile.

There are systems designed for children specifically, but these are mere extensions of dolls and other toys [SHH21, OMS18, JYBK20] who do not offer a display interaction. These studies show a merit in the use of smart toys and augmented reality applications for children in terms of immersive entertainment and creativity stimulation. However, current shape-changing and 3D displays are often targeted towards adults as their single components are too expensive or fragile for families with young children. OmniTiles tries to solve these issues by providing a sturdy, low-cost system.

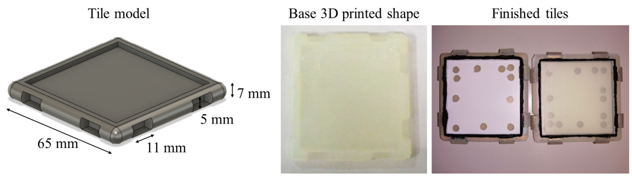


Figure 3: *Overview of our production pipeline: 3D model, 3D-printed tile, and finished tile from left to right.*

3. Hardware

Our system consist of several tiles that can be clipped together into different shapes via magnets incorporated into their sides. They can then be placed on top of an omni-procam system and are tracked via an infrared camera.

3.1. Omni Procam System

Our hardware consists of a projector (HITACHI LP-WU6500J) and a camera (FLIR Grasshopper3) which are both placed below a wide angle fisheye-lens (Opteka OPT-0.2X-37 fish-eye-lens). To align them to the same axis, a half-mirror is placed at the intersection point (see Fig. 2 middle). The camera is capable of detecting light in the infrared spectrum. To increase the capturing range, infrared LEDs are placed around the fish-eye lens to provide better lighting conditions. A filter for visible light is placed on top of the camera's lens to prevent the projector from over-lighting the camera's image.

3.2. Tiles Design

The core mechanic of our system are tiles in basic shapes which can be combined to larger structures inspired by building blocks that children play with. To achieve this, we designed triangle, square, and pentagon shaped tiles with magnets at the sides (see Fig. 1). For a magnetic attraction regardless of the tile's rotation, diametric-directional magnets which can rotate freely along their center axis are necessary.

For cost and time efficient production, the tiles are 3D printed from semi transparent material (Photopolymer resin, durable type (FLUDCL02) from Formlabs). To save material and production time, we designed the sides of the single tiles thicker (7mm) so that the 4mm-diameter magnets can be inserted into 5mm-diameter holes located at two positions on each side for the smaller tiles as depicted in Figure 3 and 4 positions at each side for the larger tiles with doubled side length, while the main projection space in the center is only 3mm thin.

After printing, the square center-shape and a dot pattern are covered for the tracking described in section 4 and the tiles are coated with a white, mat spray for a clear surface. After removing the dots-cover, the magnets are inserted into the sides and a thin, 3D printed blocker is glued to the tiles to hold the magnets in place. After that, a thin line is drawn around the main projection space to help with the outline and rotation detection. The white spray is applied in order to fill the small grooves from the 3D printing which would otherwise cause small self-occlusions (see Fig. 4) resulting in a user and angle dependent image. We refer to the coated area as a clear projection space hereinafter.

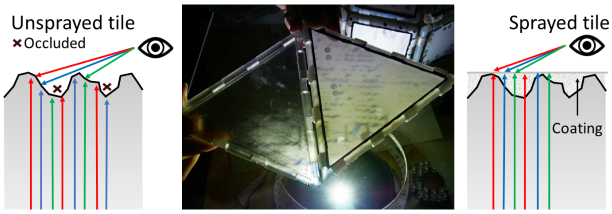


Figure 4: *Unsprayed tile (left) with occlusion due to grooves created by the 3D printing and sprayed tile (right).*

4. Implementation

The implementation of our system consists of several steps. First, we preprocess the image to detect the markers, then we derive the IDs and the rotation. Next, we compare the found markers to previously tracked markers to see if there is a more likely match before sending the markers' information to the Unity3D game engine where it is used for the applications.

Our tracking is based on Narita et al.'s [NWI16] idea to use grouped dots to identify an object and its structure. The number of dots forming a group, so dots which are locally close, provides an entry of a unique matrix which can then be identified (see Fig. 5 left). While Narita et al. used a matrix of grouped dots to cover the entire object, we reduced it to just an array of dots along the tiles' edges to increase the clear projection space. This, however, caused different requirements for our dot detection than Narita et al.'s work. For instance, concentrates Narita et al.'s work on identifying the structure of objects covered by a large unique matrix. Multiple objects are hereby considered to be far enough apart to clearly identify which dots belong to which object. However, we want to identify several smaller objects of known shape and size that might be clipped to each other and therefore very close together. For that reason, we changed the tracking to our needs

4.1. Image Processing and Tracking

4.1.1. Our method

The image is recorded in gray-scale for the tracking since the infrared light omits the color information. For the contour detection, we preprocess the cropped, gray-scale image by applying a 5x5 Gaussian Blur followed by an adaptive Gaussian thresholding to binarize the image. Via OpenCV's contour detection, we then detect possible markers' contours from this binarized image based on the following criteria: First, the outline must be closed and second, its area must be between a minimum and maximum threshold. The thresholds used were empirically determined and are influenced by factors such as the camera's field of view and the used image size. We then use the circularity of the outline to determine the shape of it. Squares have a circularity between 0.75 and 0.85, triangles are below 0.75 and pentagons above 0.85.

In a parallel step, we derive an ID array from the number of dots located close to each other where each group of dots corresponds to one entry of the ID array (see Fig. 5 left). For this, we apply a bilateral filter to the blurred gray-scale image and use it as an input for OpenCV's blob detection to find possible dots of our ID array. We then check which of the found dots lie within an outline candidate and separate them into different lists, containing the dots' positions. From each of these lists, we derive the ID array from the grouping of the dots. Hereby, dots which are closer together than a minimum distance threshold are considered to be in one group and their number is stored in the ID array as an entry, starting with the group which is positioned at a 0 degree angle in a two-dimensional coordinate system (see Fig. 5).

We made sure that every ID array of a tile would fulfill the following requirements: First, the array must be unique even when the entries are shifted in order to make sure that no two tiles will have the same ID. Secondly, the array must be asymmetric, so that the tile's rotation and side can be detected. This means that the tile's dots should not be placed in a point-symmetric pattern around the tile since this would not allow for the differentiation between a rotation of 0 and 180 degrees. Further, they should not be placed axis symmetric to the x and y axis of the tile, as the front and the back side could not be distinguished. This last requirement is only necessary as we want to create applications which require the differentiation of the tiles' sides.

4.1.2. Library and Comparison

After deriving this ID array, we compare it to the length of the previously defined ID arrays within our self-created library which we will refer to as library arrays hereinafter. The library is a text file containing the ID followed by the defined array in one line. For the comparison, we want to derive the similarity of the two arrays which is the number of consecutive, matching entries. Ideally, the similarity matches the number of entries in the library array which would provide a single, unique array. However, this is only the first of three possible cases:

1) The found array and the library array have the same length: No occlusion

2) The found array is longer than the library array: additional dots were wrongly detected

3) The found array is shorter than the library array: parts are occluded

In the first case, we can compare the array to the entries in the library without any further steps since we can assume that there was no occlusion. We store the found outline's center position together with its ID and similarity in the list of previously tracked outlines.

For the second case where the found array is longer than the library array, we can assume that there were wrongly detected dots

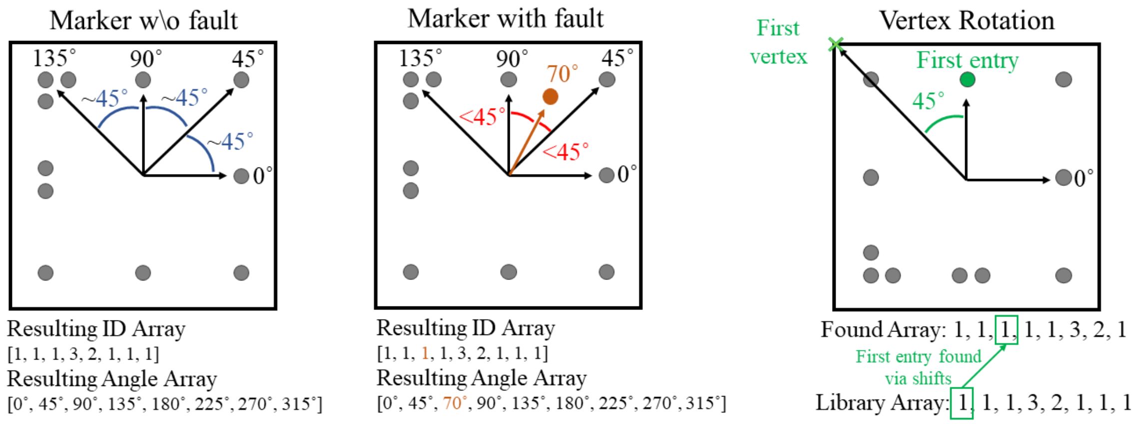


Figure 5: ***Left*** *is a tile without detection fault. The entries of its ID array are derived from the single groups of dots. Here, a single dot refers to an entry of 1, two dots to an entry of 2 and so on. The angle difference to the next groups should be the same.* ***The middle*** *shows a tile with detection fault which results in a smaller angle to the next dot groups.* ***Right****, the vertex calculation and rotation calculation is depicted.*

within the outline. This can be due to irregularities in the surface which cause shadows on the tile. For this case, we take the center position of the found groups of dots and calculate the angle to the tile center. We then store the angle differences between two neighboring dots in a list. Since we placed the dots' groups in regular distances, we can assume that the wrongly detected dot lying between these groups will have a lower angle difference to its neighbors than the other groups (see Fig. 5). Hence, we remove the groups with the minimum distances until we reached the same number of entries as the library array. After that, we proceed in the same way as in case 1.

In the last case, we differentiate how much the lengths differ. If the found array's length falls under the threshold of $T_L = \lceil libraryArraysLength * 0.5 \rceil + \lfloor corners * 0.5 \rfloor$, we do not consider it as a marker. Otherwise, we assume that there was an occlusion. Similar to the second case, we then compare the angle difference of the found dot groups to each other and fill the array with -1 at the maximum differences until the length matches the library array's length.

After ensuring that both arrays have the same length, we return the highest number of consecutively matching entries as the similarity together with all possible library array IDs that matched this sequence. Next, we check in the list of previously tracked outlines if there have been outlines around the same position with a higher similarity. We assume that the position of an outline does not change much between two frames. If there is an outline around the same position with a higher similarity, we check if the stored ID of that outline is contained in the list of possible library array IDs and take it as the most likely outline. We then store the new position but keep the higher similarity in the list of previously tracked tiles.

4.2. Vertex detection

After receiving our potential marker outlines, we extract the corners of them, so that we can send them as vertices to Unity3D. For this, we reduce the outline' point entries by comparing the distance of each point to the line drawn between its neighbors. The points with a minimum distance are then removed from the outline's list since they are likely to be part of the line and not corner.

We repeat this until the number of corners designated by the shape detection is reached. To get the correctly rotated order of the vertices, we use the previous information of our grouped dots angle. Hereby, we compare the library array to the found array to get the entry which should refer to the library array's first entry. We then find the corner closest to that entry and define it as our start corner. From their we store the other corners in counter-clockwise order.

4.3. Interaction Design

After calculating the ID and vertices of all found tiles, we store them in a list and send them via ZeroMQ[‡] to the Unity3D engine where we use the information to create and update virtual tile representations. These representations are then captured by a virtual camera and used for the output of the projector underneath the fisheye lens. To compensate for the lens distortion, the camera image is altered by a fisheye projection shader calibrated for the real fisheye lens.

The virtual tiles can then be checked for different behaviors and used for several applications. The Interactions we detect are illustrated in Figure 6 and consist of the following:

- Turning: The Z rotation changes. The change can be taken to update a value similar to a turning nob or for an object whose behavior depends on its rotation, for example, an hourglass
- Clipping: The distance of two tiles falls below a certain threshold. This can be used to initialize a combined behavior.
- Flipping: The ID of the tile's other side was detected. This can be used to display different images on each side of the tile (see Fig. 10) or for activating another behavior like the Turning.

‡ https://zeromq.org/

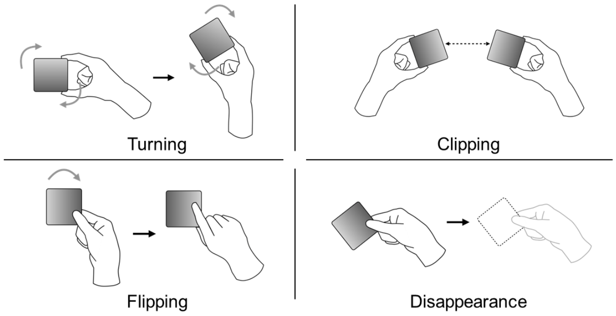


Figure 6: *Interaction design for single tiles.*

- Disappearance: The tile was covered or moved out of the image. This requires to keep track of previously detected tiles and can be used to activate behaviors or projections which should be displayed on a bigger area (see Fig. 9)

Depending on the application, we either change the appearance of the tile itself or place another object onto the tile. Placing another virtual object onto the tile has the advantage that this object can come with its own behaviors. These behaviors can be executed parallel or separate to the tile's behavior. This allows for an easy implementation of behaviors for single tiles and overall behaviors for the entire structure that the tiles are creating.

As another interaction method we created an external infrared LED light that can be seen in Figure 7 on the left. We use background subtraction and a binary threshold to detect the light's position as shown in Figure 8 and send the center of the detected light's area to Unity3D. This allows for applications like drawing and inputs via pointer (see Fig. 13).

Figure 7: *The external LED light (left) and an example application where the user can draw textures onto which animals (right).*

5. Applications

To show the flexibility of possible applications, we implemented some prototype applications based on single tiles and structures of tiles.

5.1. Single Tile Behavior

For the following applications, we used the single tiles rather than a combined structure of tiles. Each tile has a virtual object assigned to its ID which can be used to implement different behaviors. For our

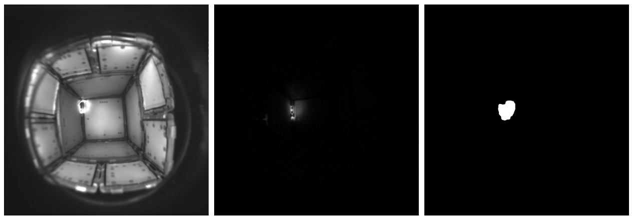


Figure 8: *The camera input image (left), the background subtraction (center) and the binarized imaged used to detect the light (right).*

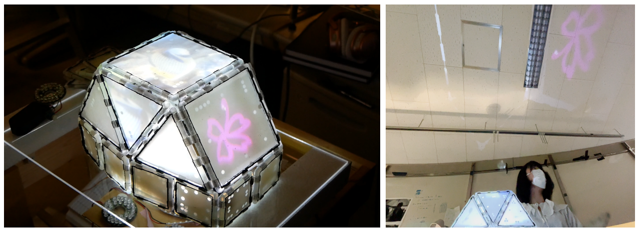


Figure 9: *Video Sculpture taking advantage of the magnification of the projection onto the ceiling when the tile is taken out by the user.*

prototype, we implemented an hourglass which can be physically turned around (see Fig. 12 right), and a changing behavior when certain tiles are clipped together. This changing behavior can be used and adapted easily for different applications. For instance, did we design some simple, nature based combinations which could be used for a children's game like a rainy cloud which can extinguish a fire placed on another tile (see Fig. 12 left).

For a more elaborate utility, we created a set of tiles where one side of the tile will display a Japanese sign and the other one will display an image of the object. In Japanese there are often words which consist of several signs. We therefore used the distance detection to change two connected signs to the object they are representing. In Fig.11 an example for the word volcano can be seen which consists of the signs for fire and mountain. By placing them together the image of a volcano will be displayed. This provides an interesting way of learning another language and could be extended to other educational applications. The physical interaction provides a haptic feedback which might enhance the user's engagement and therefore the overall learning experience [APDF11].

5.2. Structures of tiles

We further implemented applications where the user can create their own structures freely and interact with them either via the tiles themselves or via the external LED light.

5.2.1. Photo-Sculpture

Our first application displays different photos from a selected folder. On each tile one photo is displayed. By changing the physical arrangement of the tiles including the rotation, the user can create their own photo-sculpture (see Fig. 1 c). The displayed photo can be changed by flipping the tile which will select the next image

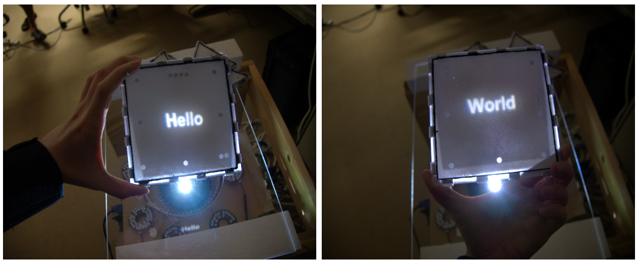


Figure 10: *Same tile displaying different images on each side.*

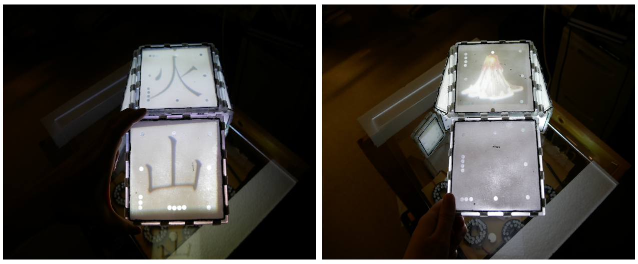


Figure 11: *Two tiles displaying the single Japanese signs to write volcano meaning fire and mountain (left) become a volcano when clipped together (right).*

in the folder. A similar approach can be used for other applications. For example could each member of an online meeting be placed one tile and the rotation or flipping could be used to mute the single participants.

5.2.2. Video-Projection

Similar to the first application, the user can create a sculpture where each tile shows a movie or video. The video is paused as long as the tile it belongs to is tracked. However, when the user takes out the tile so that it is not tracked by the camera anymore, the movie will start playing. For this application, we take advantage of the circumstance that the projection's size increases when displayed onto a further object (see Fig. 9 right). By taking out the tile, the video can be projected in a larger scale onto another surface like a wall. Additional functionalities could be implemented with other tiles of the same sculpture. For example, could one tile be used to increase or decrease the sound volume when taken out or to rewind.

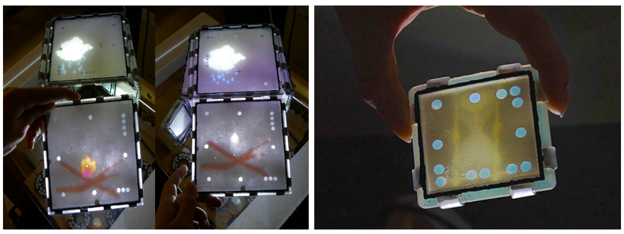


Figure 12: *Fire being extinguished upon combining with rainy cloud (left) and hourglass which can be physically turned.*

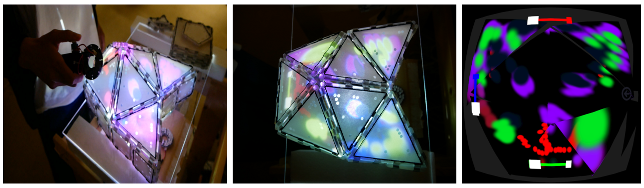


Figure 13: *User drawing onto a structure (left), the final image (center) and the in Unity view of the drawing (right).*

5.2.3. Drawing

This application makes use of the external LED light. The user can first create a structure and then draw onto it. They can choose the color and brush size via sliders on the side of the structure that can be activated with the external light. It is further possible to exchange the drawn color for a texture and assign additional behavior to it. In Figure 7 on the right, the colors were exchanged for grass and water textures onto which animals like fish and rabbits would then spawn.

5.2.4. Game

As a last application, we created a small game where the user can create their own map. To show this, we designed a simple space-shooter where the user can move around and shoot at each other's space-ships (see Fig. 14). Though, the principle of map creation is applicable to several games.

First, the user sets up the structure. Then, the external light can be used to draw different obstacles or grounds onto the structure. In our case, we provided three types: A ground that would speed up the player while moving above it (red), one that slows them down (blue) and one that is not passable (stone) and destroys the player upon touching. The players can then move their ship but also take out the tiles that other players are on to destroy them. In a first test-play with members of our laboratory, we noticed that the users turned the drawing phase of the map already into a game where the one holding the external LED light tried to capture the players by drawing the not passable ground onto them. Furthermore, the users would move around the three dimensional display, increasing their physical exertion.

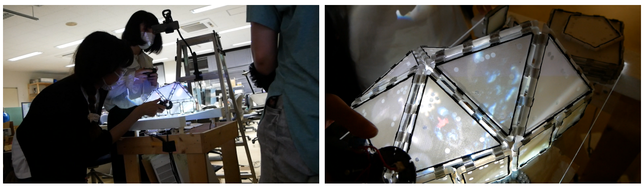


Figure 14: *Two user interacting with the system where one tries to capture the other user's ship (left) and a closeup of a user shooting at one of the structures (right)*

6. Discussion

The variety of possible applications shows the high customization potential of our system. The physical interaction and sturdiness of the system allows for experimentation and creativity by the user. Nonetheless, there are still considerations and limitations we would like to discuss hereinafter.

6.1. Computational Time Improvement

First of all, we are currently running our whole system as a single process which can cause a noticeable delay in displaying the correct tiles position and content. The length of the delay depends on the amount of tiles visible in the image since for every tile, the needed steps for the tracking have to be performed after each other. To get rid of this dependency and to improve the overall runtime, we would like to run them in parallel by using multiprocessing.

6.2. Increasing Projection Space

Though our tracking is designed to optimize the projection space in the center of the tile, the magnets and dots for the ID recognition are still decreasing the clear projection surface. While the dots are displaying the information but with a lesser quality, the magnets are blocking the projection light completely. Therefore, we would like to investigate more into alternative connections like stacking. The sides of the tiles could be designed in a way so that they will connect to each other by their shape itself. We need to pay attention to the robustness of the material though, as some shapes might break off easily, especially, when frequently attached and detached. Hence, we need to carefully balance between the increase of the projection space and the usability.

6.3. Alternative Materials

So far, the structure of created displays with our system needs to be visible by the camera-projector set-up. However, alternative materials for the center part of the tiles could increase the projection space in interesting ways. For instance, a half-mirror placed inside the structure could funnel the light of the camera-projector set-up to previously occluded tiles like in Figure 15. This would allow for even more complex structures. Though, it needs to be determined if the light reaching through the half-mirror is still bright enough to create a clear image.

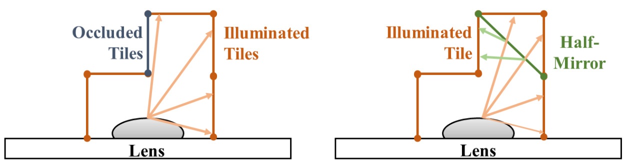


Figure 15: *An occluded tile (left) can be projected onto by using a half-mirror at the position seen on the right side.*

Another material consideration is polymer dispersed liquid crystal (PDLC) foil which turns transparent when connected to electricity and is opaque otherwise. This change can be performed within milliseconds not perceptible by the human eye which would allow to seemingly project onto the foil and the surroundings behind it at the same time.

A last consideration are small LCD screens. These would have the disadvantage of reducing the overall display space since the surroundings cannot be projected onto but would increase the mobility and image quality of the system. Ideally, our system will incorporate all of these materials in different tiles that can be exchanged and combined depending on the situation and the user's needs.

6.4. Alternative Setups

To deal with the over-lighting that the projector would cause for a standard RGB camera, we are using an infrared camera with some infrared LEDs around the fisheye lens. While this solves the problem of the over-lighting, it decreases the range of the system since light does not reach objects which are further away. Moreover, any color information is lost which limits the tracking possibilities. Therefore, we would like to switch to an RGB camera and a high-speed projector. This would allow us to synchronize them, so that the camera will only capture during the frames in which the projector's image will be turned to black.

6.5. Interaction Methods and Evaluation

Our current system provides an interaction based on the tiles themselves and an external LED light. However, we would like to include other input methods as well. For instance, could holes in the tiles' structure be used to capture the users themselves, therefore allowing gesture recognition as an input. Furthermore, as alternative to the external light, we would like to include a touch based input. When a constant ambient light shines through the tiles, the user's finger can be detected as a shadow on them. However, the current coating cannot be applied homogeneous enough to always ensure the finger to be detected which is why we are experimenting with alternative coatings for the tiles like tracing paper.

To confirm the practicality and intuitiveness of our system, we still need to conduct a user study. Since our system is targeted towards non-experts and in particular families, we would like to conduct a user study with children. We think that our tiles are suited well for children since they are made from robust and inexpensive material that can withstand a rigorous handling typical for young children. In the study we would like to investigate whether our implemented interaction methods are comprehensible and enjoyable for a young user unfamiliar to the system. Further, we would like to observe if and which other interactions the user will try to perform with the tiles that we have not thought of.

7. Conclusion

We present OmniTiles, a user customizable display consisting of base tiles which can be arranged freely by the user via a magnetic clipping system. Our system combines omni-directional projection displays with the high customizability of shape changing interfaces while enabling various behaviors for single tiles. The introduced clipping and tracking provide a sturdy low-cost system that allows the user to generate and change complex structures without any prior training. This makes the system particularly suited for families with young children who can experiment with the tiles freely. We demonstrated the variety of our system by showing several applications and interaction methods. We further discussed ways to improve and extend the system as well as future plans.

References

[APDF11] ARNAB S., PETRIDIS P., DUNWELL I., FREITAS S. D.: Enhancing learning in distributed virtual worlds through touch: a browser-based architecture for haptic interaction. In *Serious Games and Edutainment Applications*. Springer, 2011, pp. 149–167. 6

[BW10] BENKO H., WILSON A. D.: Pinch-the-sky dome: freehand multi-point interactions with immersive omni-directional data. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. 2010, pp. 3045–3050. 3

[CLW*14] CASALEGNO F., LIM Y., WINFIELD C., SILVESTER K., LOWE M., KIM S., ZAMAN C. H.: Lume-building identity, displaying content, and engaging users through network of interactive display. In *Proceedings of The International Symposium on Pervasive Displays* (2014), pp. 192–193. 2

[EA17] EVERITT A., ALEXANDER J.: Polysurface: a design approach for rapid prototyping of shape-changing displays using semi-solid surfaces. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (2017), pp. 1283–1294. 2

[GSL*19] GOGUEY A., STEER C., LUCERO A., NIGAY L., SAHOO D. R., COUTRIX C., ROUDAUT A., SUBRAMANIAN S., TOKUDA Y., NEATE T., ET AL.: Pickcells: A physically reconfigurable cell-composed touchscreen. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–14. 2

[HLW*18] HAN S., LIU B., WANG R., YE Y., TWIGG C. D., KIN K.: Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–10. 3

[HWT*15] HARDY J., WEICHEL C., TAHER F., VIDLER J., ALEXANDER J.: Shapeclip: towards rapid prototyping with shape-changing displays for designers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 19–28. 2

[ISS*15] IBAYASHI H., SUGIURA Y., SAKAMOTO D., MIYATA N., TADA M., OKUMA T., KURATA T., MOCHIMARU M., IGARASHI T.: Dollhouse vr: a multi-view, multi-user collaborative design workspace with vr technology. In *SIGGRAPH Asia 2015 Emerging Technologies*. 2015, pp. 1–2. 2

[JYBK20] JIMBU M., YOSHIDA M., BIZEN H., KAWAI Y.: Proposal for an interactive dollhouse with multiple sensors and projection mapping. In *Proceedings of the 8th International Conference on Human-Agent Interaction* (2020), pp. 236–238. 2, 3

[LGB*16] LINDLBAUER D., GRØNBÆK J. E., BIRK M., HALSKOV K., ALEXA M., MÜLLER J.: Combining shape-changing interfaces and spatial augmented reality enables extended object appearance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 791–802. 2

[LHL*21] LU C.-Y., HSIEH H.-W., LIANG R.-H., LEE C.-J., YANG L.-C., XUE M., GUO J.-L., HSIEH M.-J., CHEN B.-Y.: Combining touchscreens with passive rich-id building blocks to support context construction in touchscreen interactions. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–14. 2

[LK12] LI M., KOBBELT L.: Dynamic tiling display: building an interactive display surface using multiple mobile devices. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (2012), pp. 1–4. 2

[MPSK18] MAEDA K., PIEKENBROCK M., SATO T., KOIKE H.: A tabletop system using an omnidirectional projector-camera. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (2018), pp. 311–314. 3

[MYU*18] MIYASHITA L., YAMAZAKI T., UEHARA K., WATANABE Y., ISHIKAWA M.: Portable lumipen: Dynamic sar in your hand. In *2018 IEEE International Conference on Multimedia and Expo (ICME)* (2018), IEEE, pp. 1–6. 2, 3

[NULI17] NAKAGAKI K., UMAPATHI U., LEITHINGER D., ISHII H.: Animastage: hands-on animated craft on pin-based shape displays. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (2017), pp. 1093–1097. 2

[NWI15] NARITA G., WATANABE Y., ISHIKAWA M.: Dynamic projection mapping onto a deformable object with occlusion based on high-speed tracking of dot marker array. In *Proceedings of the 21st ACM symposium on virtual reality software and technology* (2015), pp. 149–152. 3

[NWI16] NARITA G., WATANABE Y., ISHIKAWA M.: Dynamic projection mapping onto deforming non-rigid surface using deformable dot cluster marker. *IEEE transactions on visualization and computer graphics 23*, 3 (2016), 1235–1248. 3, 4

[OMS18] OZAKI H., MATOBA Y., SIIO I.: Can i gettoyin? a box interface connecting real and virtual worlds. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–4. 2, 3

[PM12] PLA P., MAES P.: Display blocks: cubic displays for multi-perspective visualization. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. 2012, pp. 2015–2020. 2

[PP04] PARK H., PARK J.-I.: Invisible marker tracking for ar. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality* (2004), IEEE, pp. 272–273. 3

[RKLS13] ROUDAUT A., KARNIK A., LÖCHTEFELD M., SUBRAMANIAN S.: Morphees: toward high" shape resolution" in self-actuated flexible mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), pp. 593–602. 2

[RPPH12] RASMUSSEN M. K., PEDERSEN E. W., PETERSEN M. G., HORNBÆK K.: Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), pp. 735–744. 1

[RUO01] REKIMOTO J., ULLMER B., OBA H.: Datatiles: a modular platform for mixed physical and graphical interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), pp. 269–276. 2

[SHH21] SCHOLZ E., HONAUER M., HORNECKER E.: sssnake: a remote multiplayer toy for children. In *Interaction Design and Children* (2021), pp. 563–567. 3

[SIF*20] SUZUNAGA S., ITOH Y., FUJITA K., SHIRAI R., ONOYE T.: Coiled display: Make everything displayable. In *SIGGRAPH Asia 2020 Emerging Technologies*. 2020, pp. 1–2. 2

[SII*20] SUZUNAGA S., ITOH Y., INOUE Y., FUJITA K., ONOYE T.: Tuve: A shape-changeable display using fluids in a tube. In *Proceedings of the International Conference on Advanced Visual Interfaces* (2020), pp. 1–9. 2

[SKPP19] SINHA K., KUMARI R., PRIYA A., PAUL P.: A computer vision-based gesture recognition using hidden markov model. In *Innovations in Soft Computing and Information Technology*. Springer, 2019, pp. 55–67. 3

[TBS*18] TIAB J., BORING S., STROHMEIER P., MARKUSSEN A., ALEXANDER J., HORNBÆK K.: Tiltstacks: composing shape-changing interfaces using tilting and stacking of modules. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (2018), pp. 1–5. 2

[TIHS20] TONE D., IWAI D., HIURA S., SATO K.: Fibar: Embedding optical fibers in 3d printed objects for active markers in dynamic projection mapping. *IEEE transactions on visualization and computer graphics 26*, 5 (2020), 2030–2040. 3

[TOA*16] TAKASHIMA K., OYAMA T., ASARI Y., SHARLIN E., GREENBERG S., KITAMURA Y.: Study and design of a shape-shifting wall display. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (2016), pp. 796–806. 2

[TSMK21] TAKAGI L., SATO T., MIYAFUJI S., KOIKE H.: Real-time projection of lip animation onto face masks using omniprocam. In *ACM SIGGRAPH 2021 Posters*. 2021, pp. 1–2. 3

[WSF18] WALTER V., SASKA M., FRANCHI A.: Fast mutual relative localization of uavs using ultraviolet led markers. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* (2018), IEEE, pp. 1217–1226. 3

[YOTI14] YAO L., OU J., TAUBER D., ISHII H.: Integrating optical waveguides for display and sensing on pneumatic soft shape changing interfaces. In *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology* (2014), pp. 117–118. 2