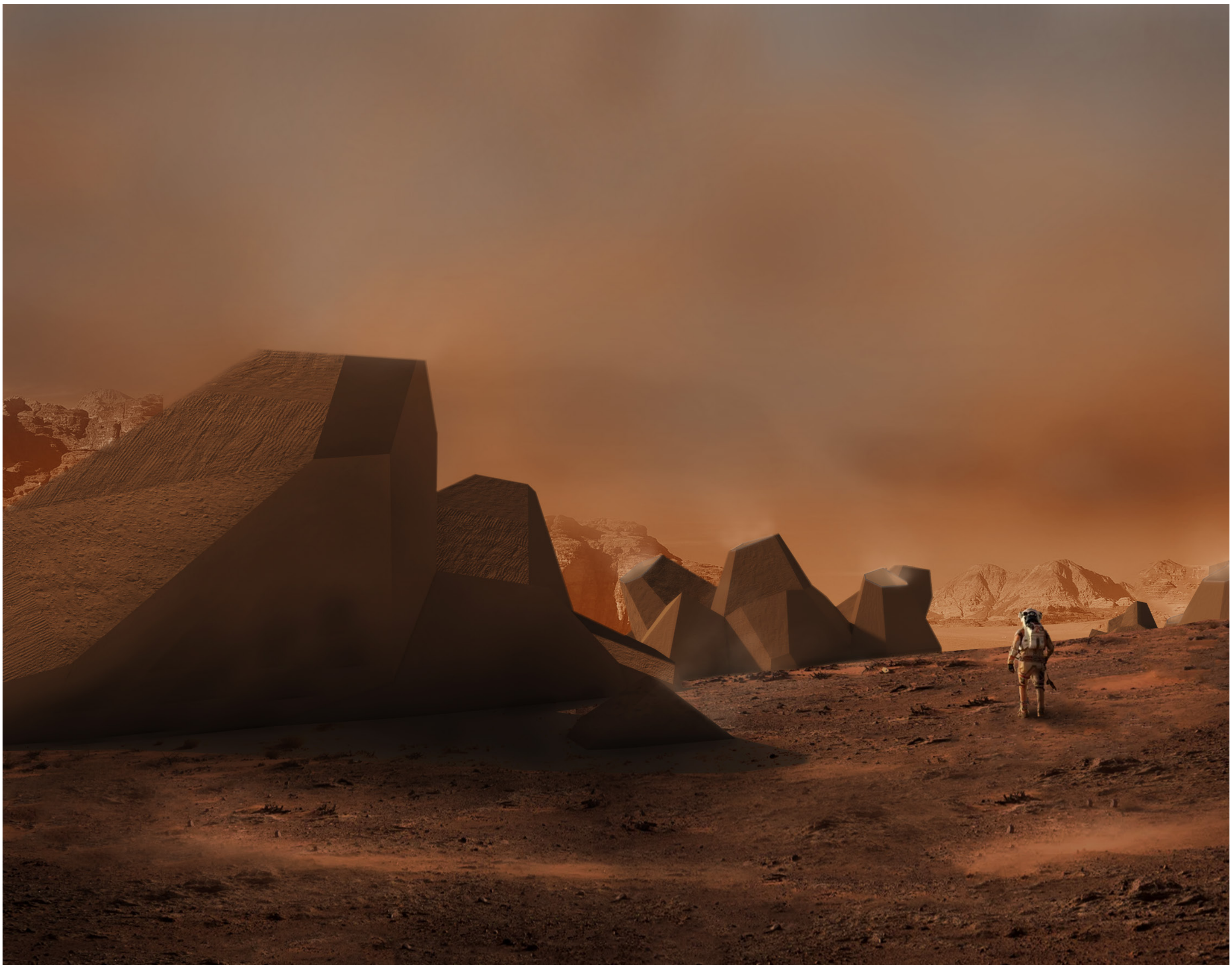
AR0122 1:1 Interactive Architecture Prototypes Workshop (2022/23 Q3)



Communal Housing Typology in Martian Habitat

Group 4:

Coby Bianco 4789717
David Sauer 5835380
Mikolaj Waszkiewicz 5861373
SitChiManSimon 5742617

Coordinators:

Henriette Bier, Arwin Hidding and Vera Laszlo (RB lab); Seyran Kahdemi and Casper van Engelenburg (AiDAPT lab); Micah Prendergast and Luka Peternel (CoR lab)

Part I. Introduction

Living on Mars has become a current exploration with the advancement of aerospace technology. According to Nichols(2017), NASA and SpaceX wanted to send humans to Mars by 2030 and 2024 respectively. Therefore, it is the time to develop architectural habitat solutions for living on Mars. As human effort is limited on Mars, this project aims to provide a voronoi housing scheme using human-and-robot collaboration. It will first identify the challenges and design ideas for martian habitat; then explore the design of communal living units using local materials(regolith); and finally explore the way to realize the design using technologies of Design-to-Robotic-Production-Assembly(D2RP), Computer Vision(CV) and Human-Robot Collaboration(HRC).

Part II. Challenges in Mars

Apart from the lack of construction materials and human labor forces, there are three main challenges in the martian habitat environment this project aims to cope with:

1. The average radiation level in Mars is 40-50 times that of the Earth, which also exceeds the limit that humans can receive by 13 times.("*Radiation*", 2023) The living space needs to be protected from radiation for human health.

2. The average ground temperature on Mars swings from 22° Celsius in daytime and -99° Celsius.("*RoverTemperature Control*", 2023) Apart from using the heating system, the design should reduce the temperature swing to reduce energy loss.

3. Compared to the Earth, Mars is more far away from the Sun. The design need to balance light gain and protection from radiation.

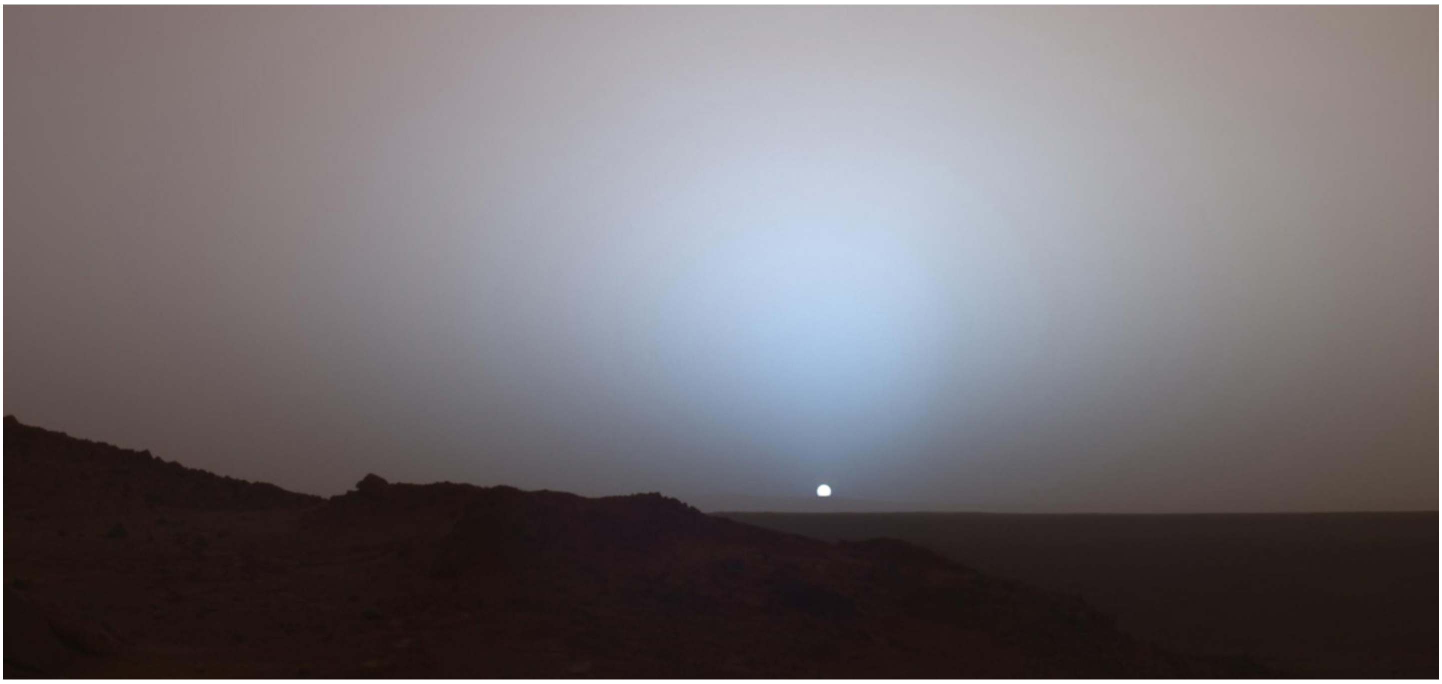


Fig. 1. Lighting condition on Mars ("*What Does a Sunrise-Sunset Look Like on Mars?*", 2019)

Part III. Design

Case studies

Firstly, the design references the study of rhizome 1.0 done by an architectural and aerospace team in Tudelft, which suggested underground constructions using local regolith for radiation and temperature protection. However, the light gain and communal functions of individual underground houses is limited. Therefore, this project also references cave houses in Haddej village, which provided a communal courtyard for light gain and events for the underground houses.(Taylor, 2018) Moreover, the Mars Ice House designed by SEArch and Clouds AO in 2015 suggested using an ice membrane to shield off radiation. As this project focuses on using regolith, the Mars Ice House gives insight into constructing regolith canopies for light gain and radiation protection.

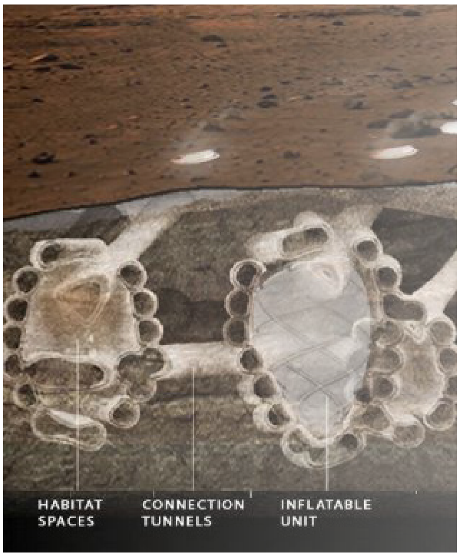


Fig. 2. Rhizome1 (Bier et al., 2023)

Fig. 3. Cave houses in Haddej village (Taylor, 2018)

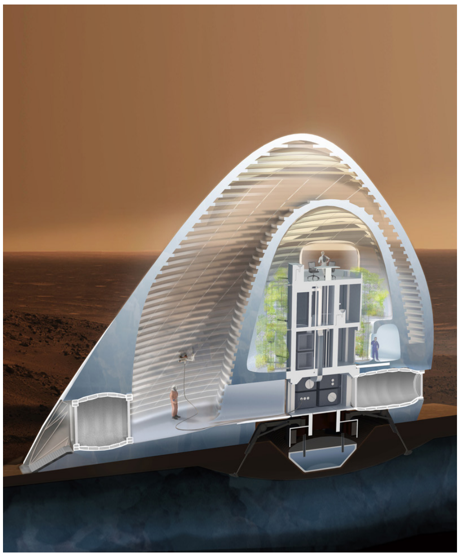


Fig. 4. Mars Ice House ("*Mars Ice House*", 2015)

Concept

In the design, several underground individual houses are connected with the central courtyard, which is further covered by canopies. Different clusters are also interconnected with each other through underground tunnels, which also link different communities. The project also integrates each element with voronoi design, which enables cell customization for different functions while maintaining constructional and structural efficiency.

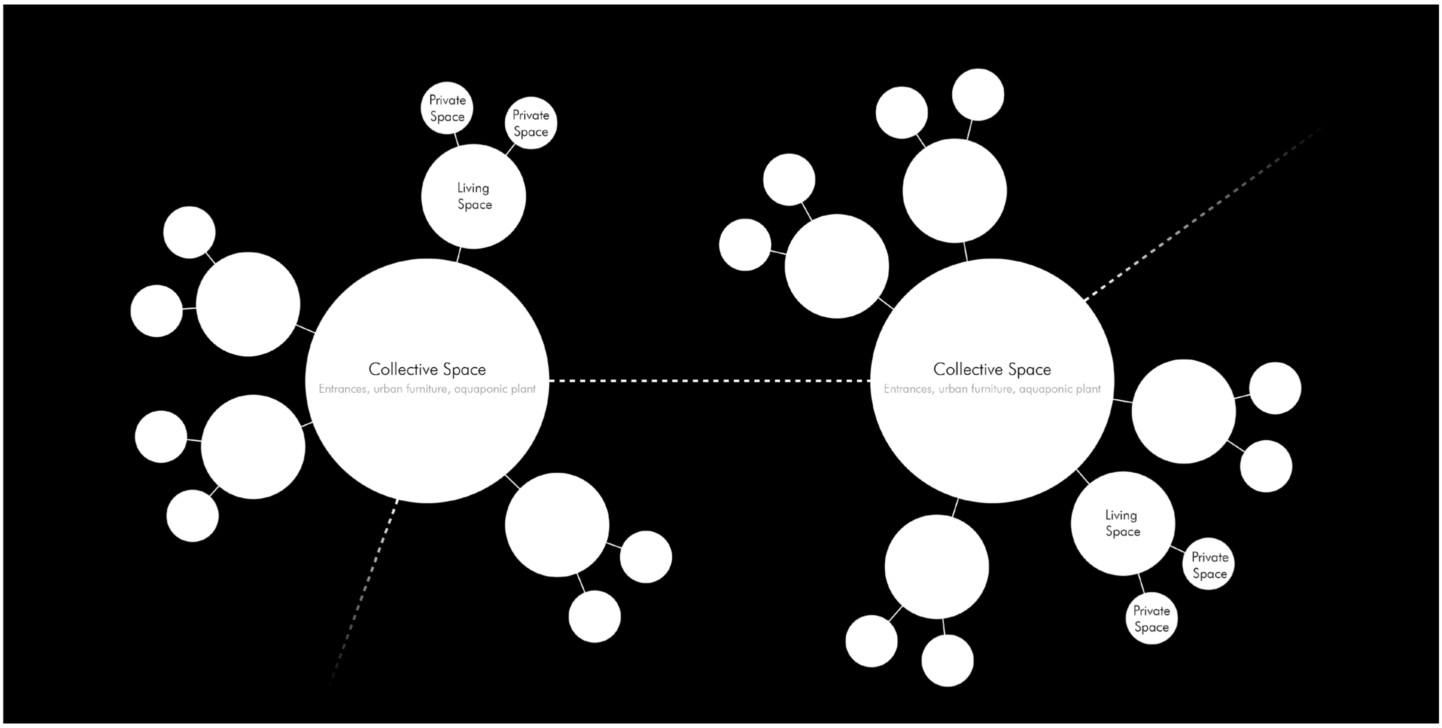


Fig. 5. Conceptual spatial relationships in the habitat

House unit design

The communal habitat design starts with the house unit design. The house unit uses voronoi cell structure, in which dimension and interrelationships of each cell space can be flexibly manipulated using grasshopper script while maintaining the integrated cell structure. A spatial diagram is first considered to arrange the cell spaces according to privacy and light gain. For example, more public entrance and dining room is located towards the communal courtyard; working space protrudes above the ground to receive more natural light; while the most private bedroom, where inhabitants stay for long duration to sleep, is located in the innermost part to protect the inhabitants from radiation. Afterwards, each space, represented by points in the digital model, are used as the input of the grasshopper script, and translated into a voronoi cell that is customized in size and integrated with each other. With the controlled input, the script generates similar options of units, in which the option with best spatial condition is chosen for the design.

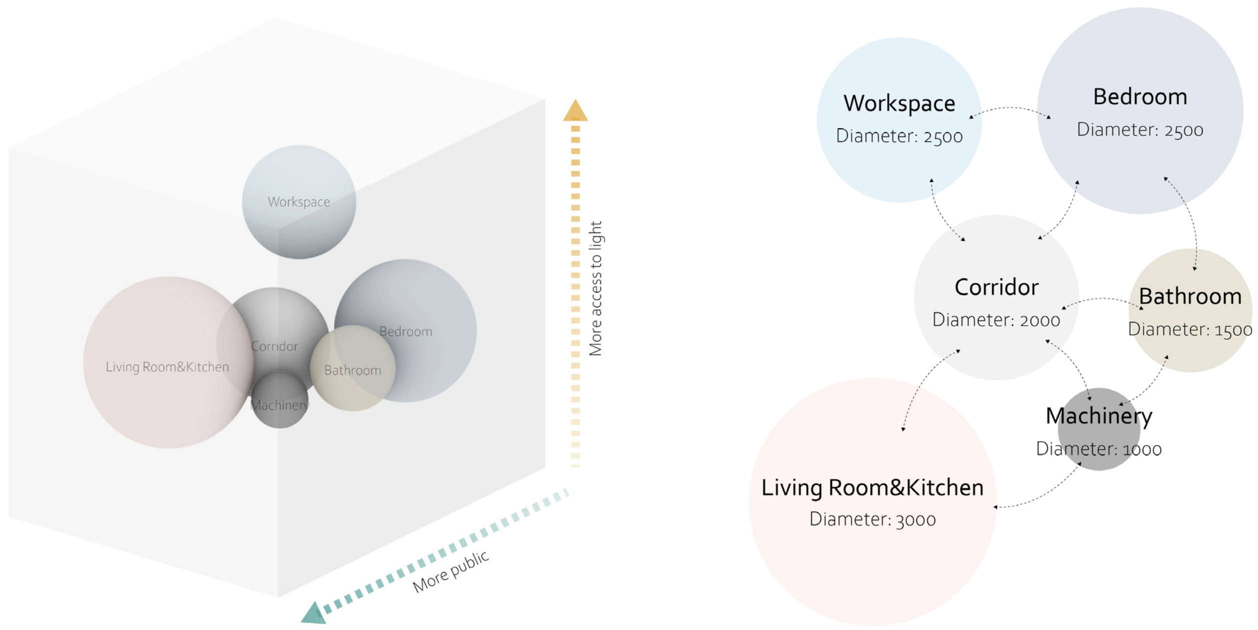


Fig. 6. Conceptual spatial relationships in the house unit

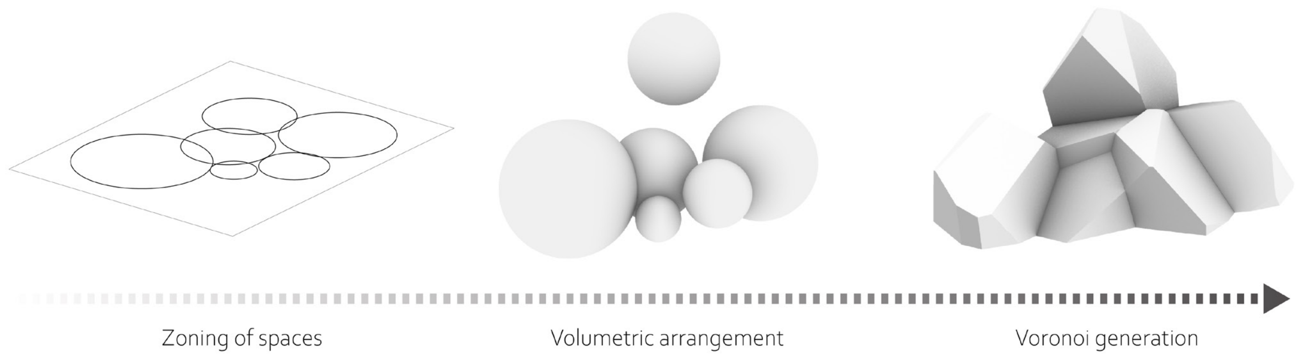


Fig. 7. From input to voronoi structure

Fig. 8. Grasshopper script generates controlled options

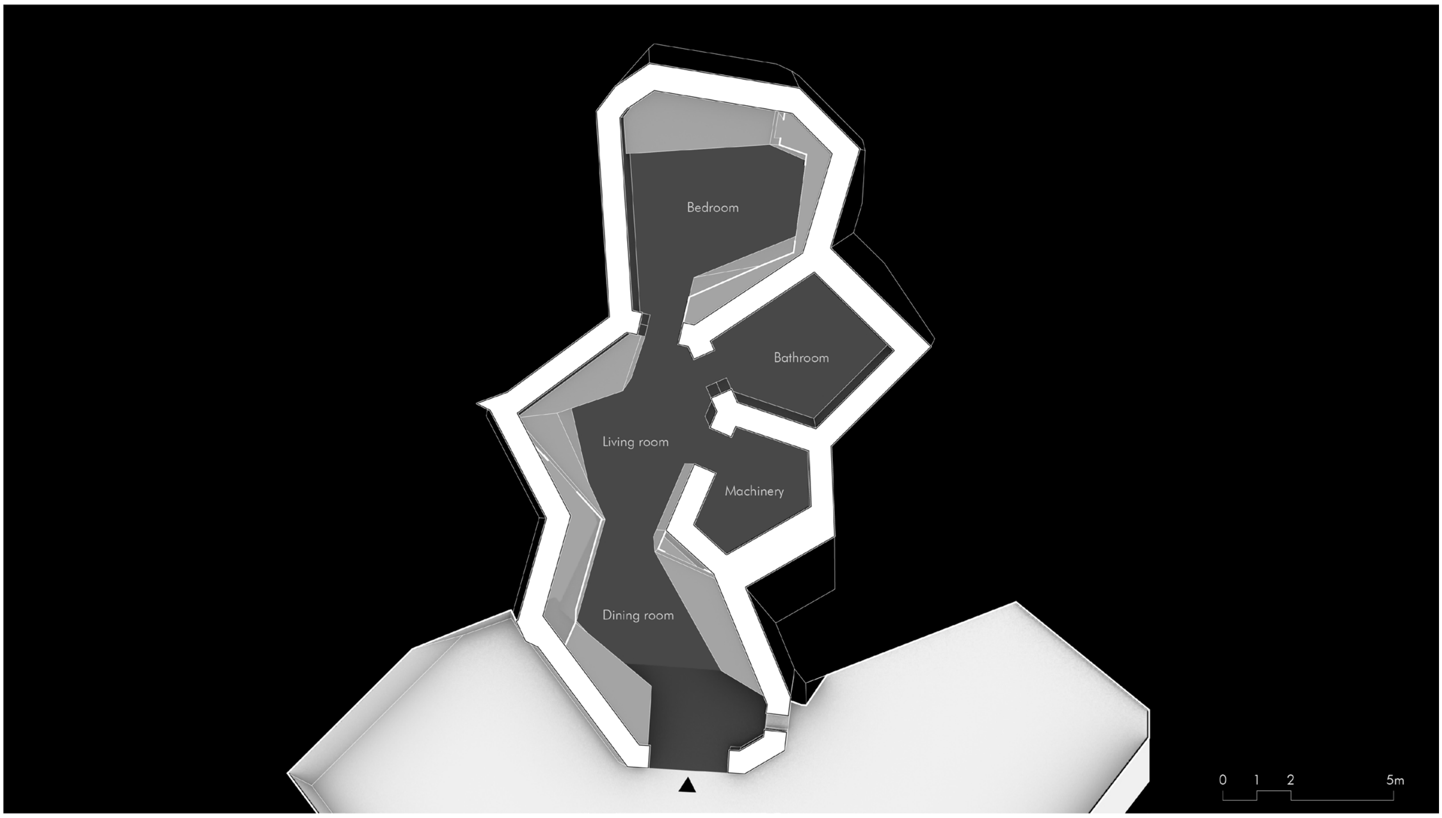


Fig. 9. Plan of voronoi house

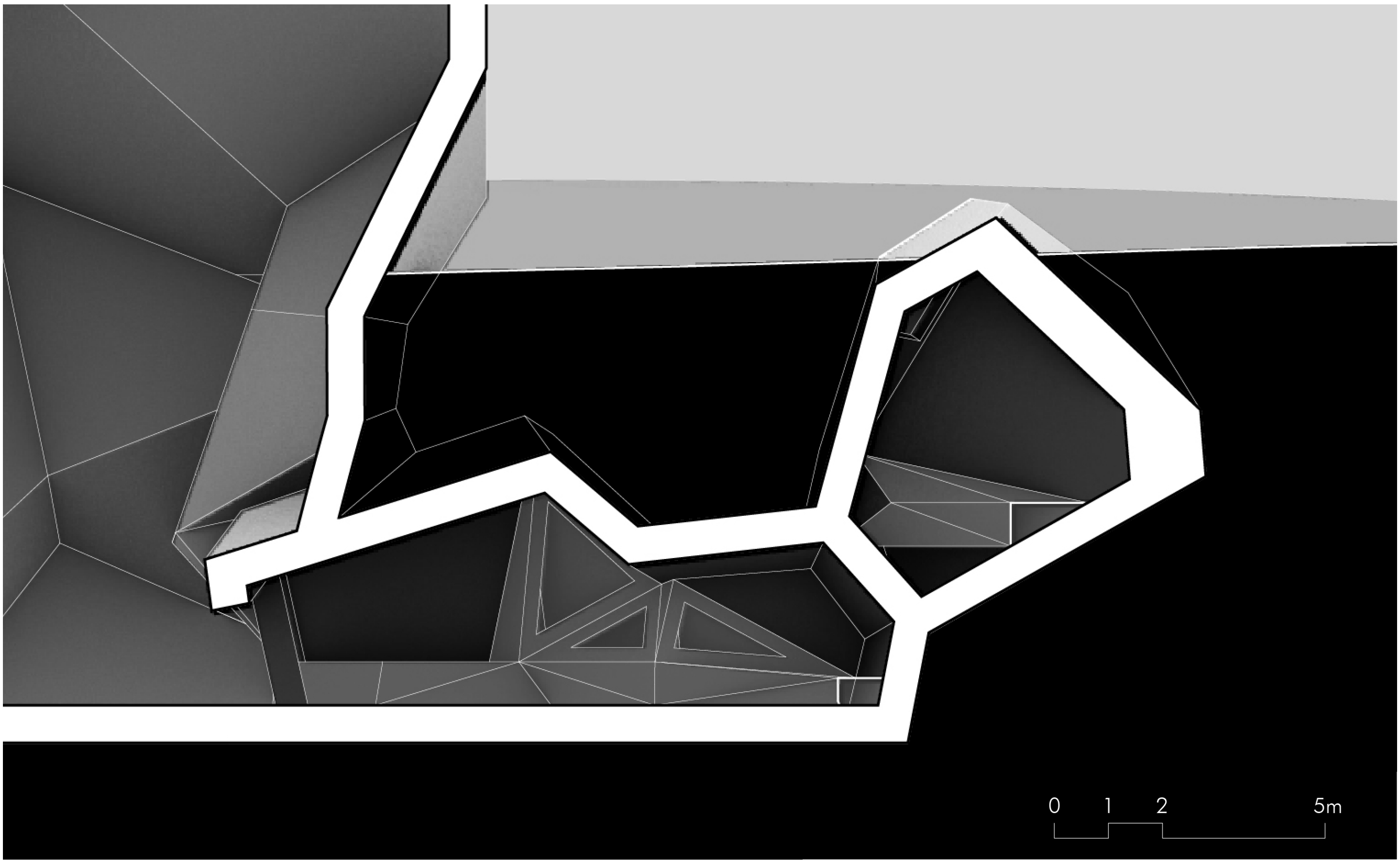


Fig. 10. Section of house with furniture

Furniture design

The furniture adopts continuous polygonal shape to recapture the polygonal gesture of the voronoi house. The polygonal furniture initiates from the edges or vertex of the house, then transforms up and down for different functions, such as tables, seatings, storage and bedroom. In this way, the structure of the house and furniture are united and can be constructed collectively.

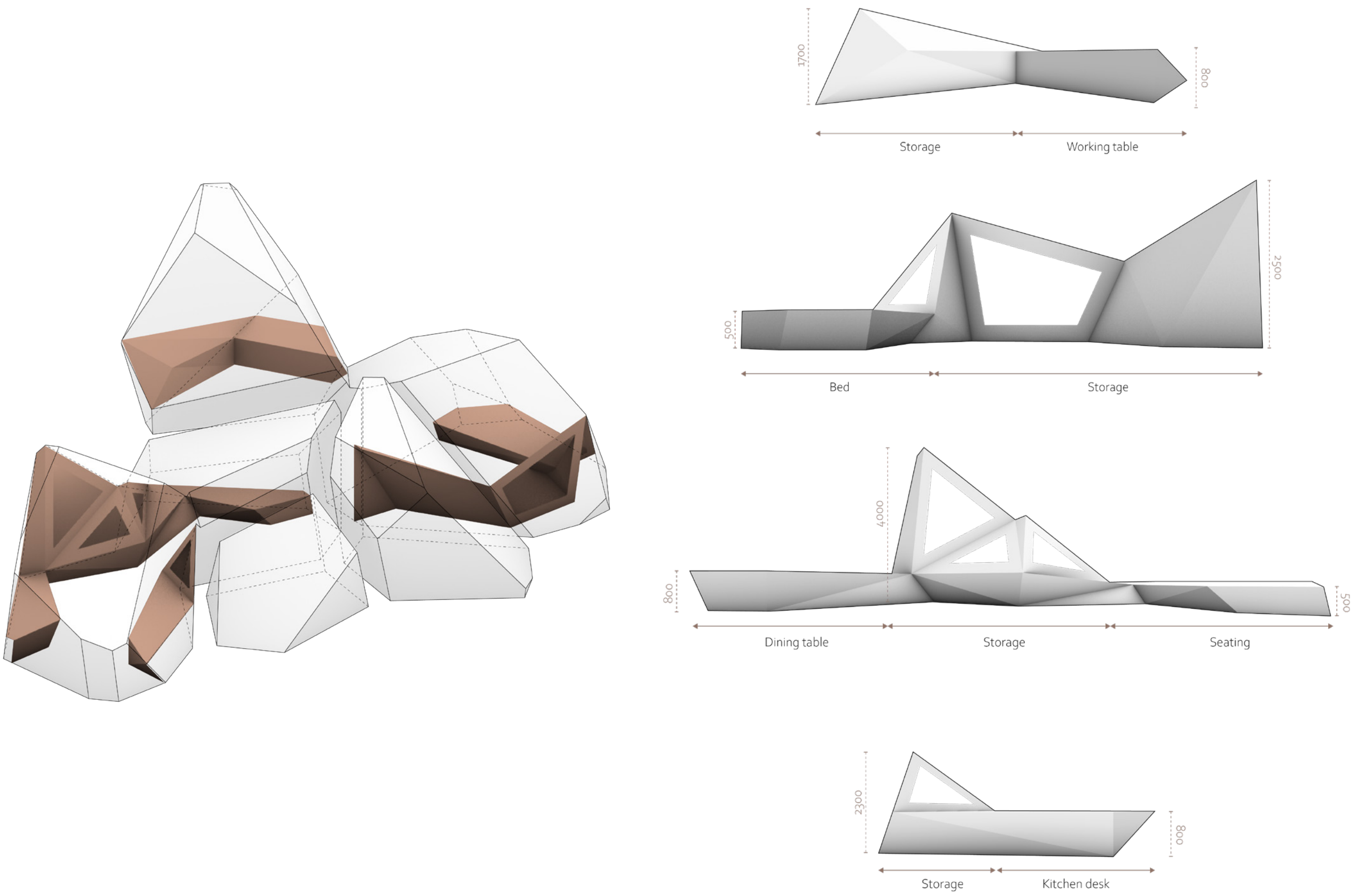


Fig. 11. Polygonal furniture in relation to voronoi house

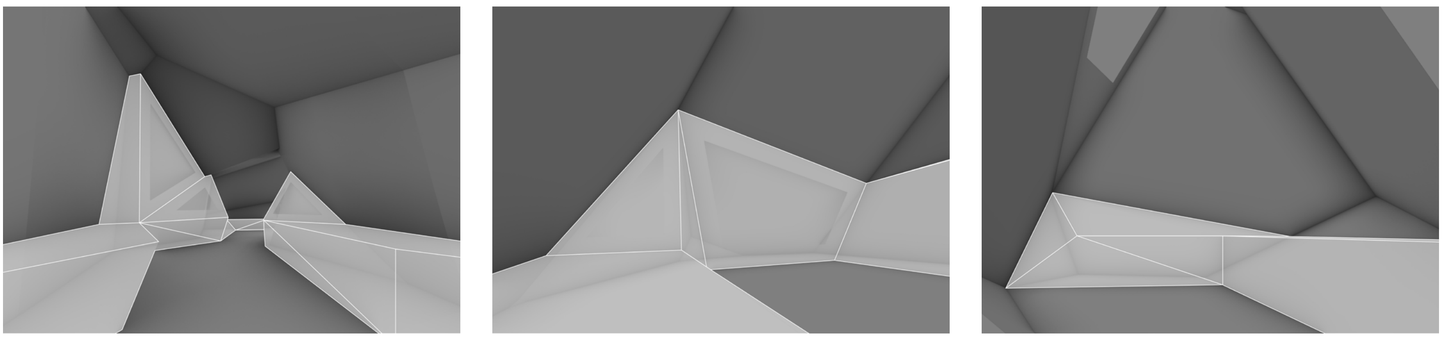


Fig. 12. Furniture in dining room(left), bedroom(middle) and studioi(right)

Canopy design

The canopy, which protects the inner courtyard from radiation, also takes voronoi shape to integrate with the houses. The canopy structure starts from the bottom of the courtyard and rises above the ground. The voronoi cell structure is generated by grasshopper script, in which cells at the bottom are smaller in size to unit with house unit cells, while cells above the ground are larger for structural efficiency and to serve as light tunnels. Cells for light tunnels are also elongated, in which radiation is weakened while bouncing within the tunnel and reaches the courtyard as mild indirect light.

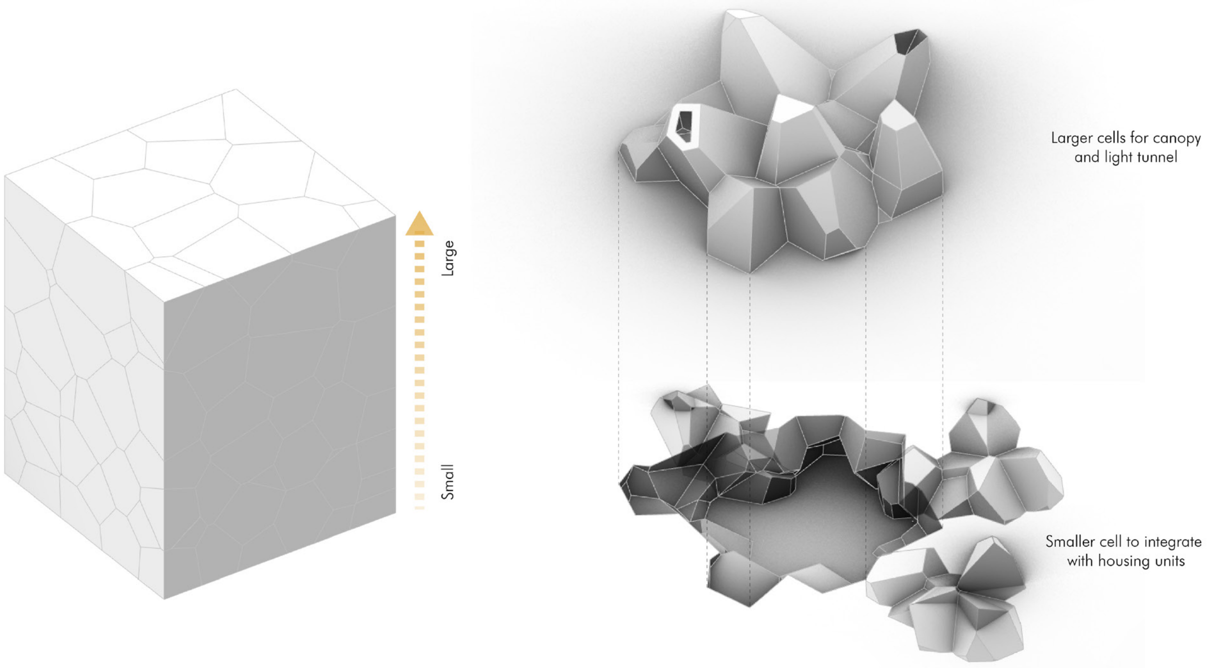


Fig. 13. Generation of gradient cells for canopy and courtyard

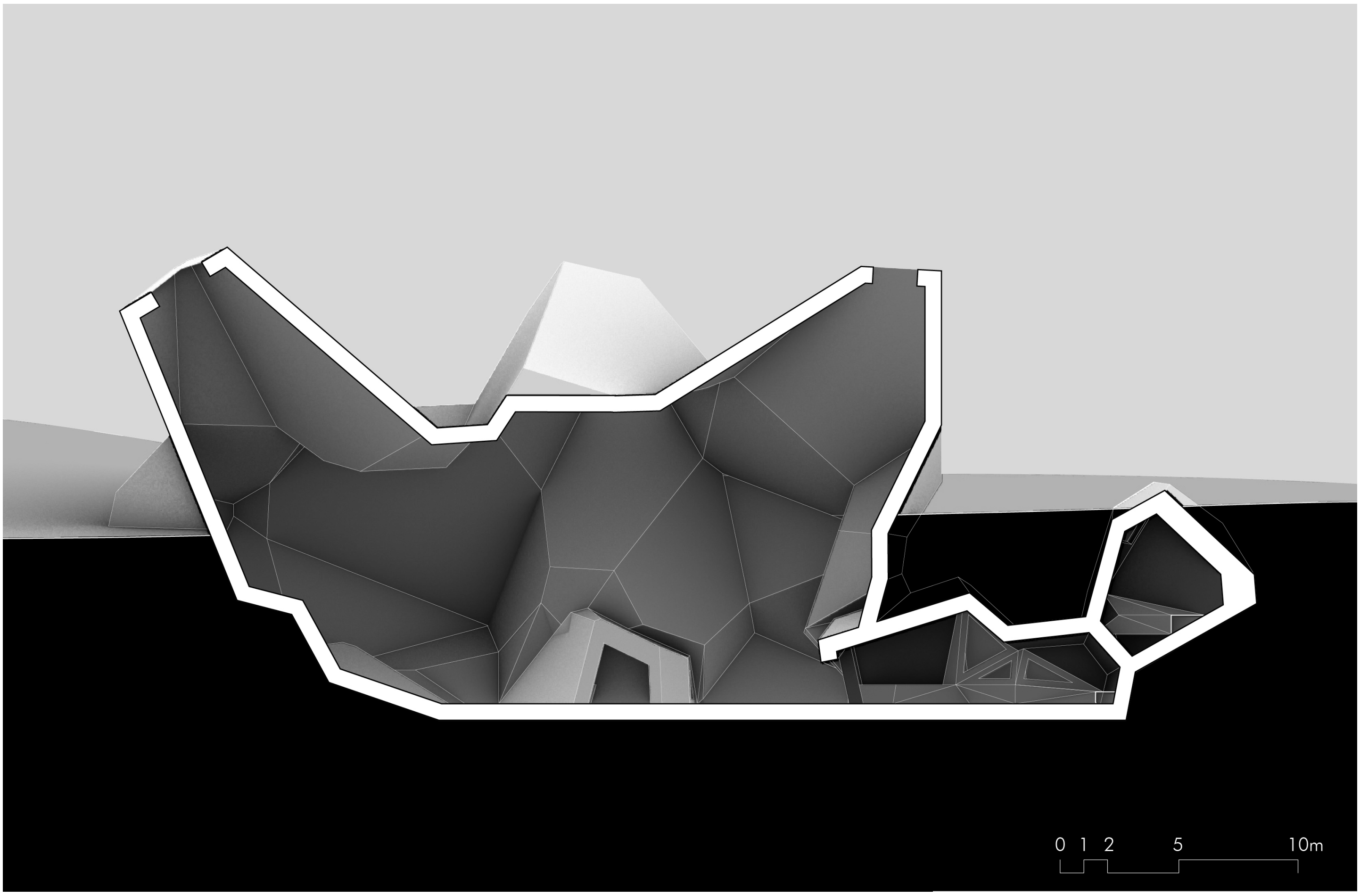


Fig. 14. Section of courtyard, canopy and houses

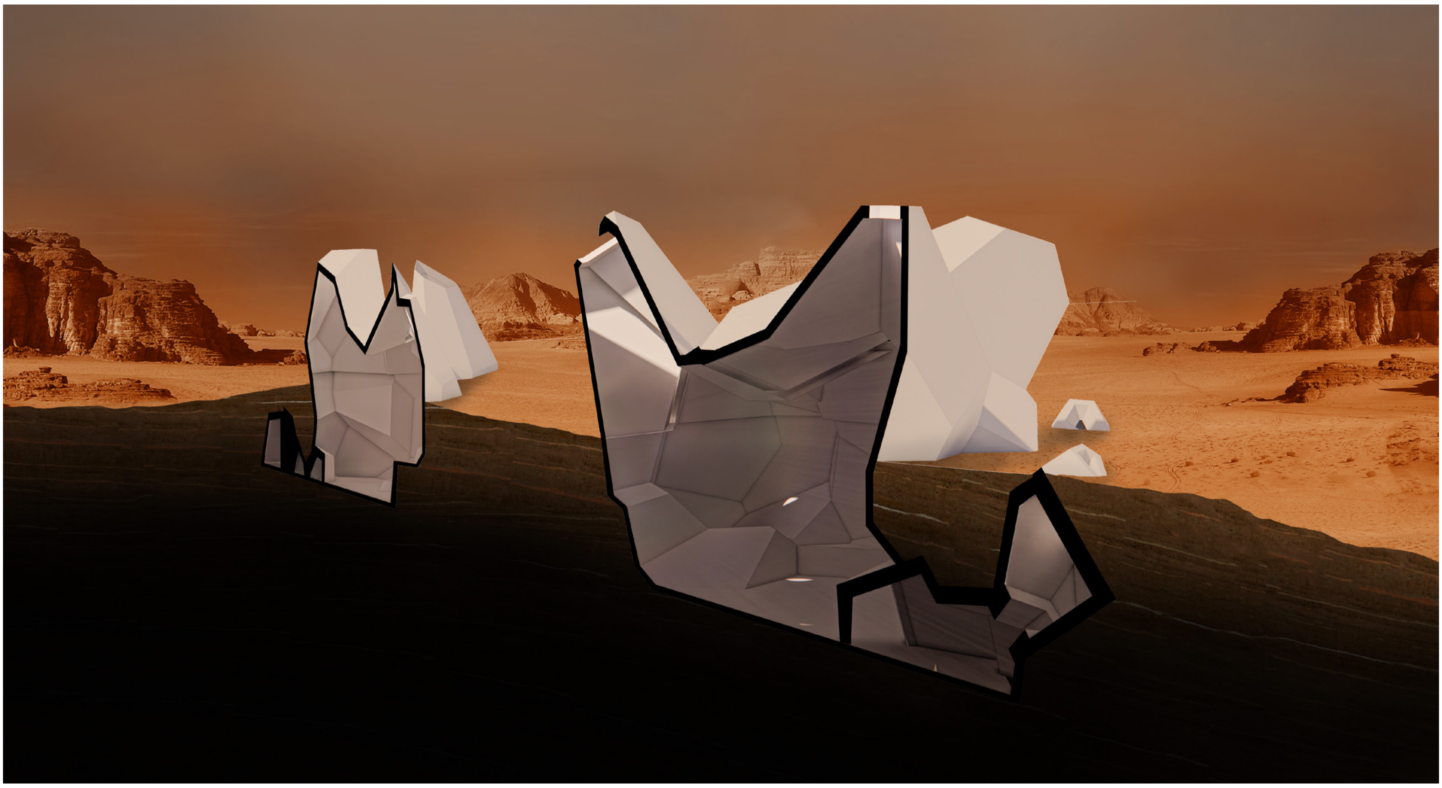


Fig. 15. Perspective section of courtyard, canopy and houses

Courtyard design

More elements are added within the courtyard to enrich its function. For example, cascading voronoi cells are added at the courtyard entrance as stairs, and independent voronoi cells are added on ground as communal seating. Aquaponic plants, as suggested by NASA(2020), are grown in the courtyard using waste from the inhabitants to improve the psychological environment, and provide oxygen which is needed for daily living in Martian habitats. Furthermore, the courtyard ground can also adopt a voronoi pattern, and the pipes which support nutrients from the house to the aquaponic plants, can be installed in the gap between the floor cells.

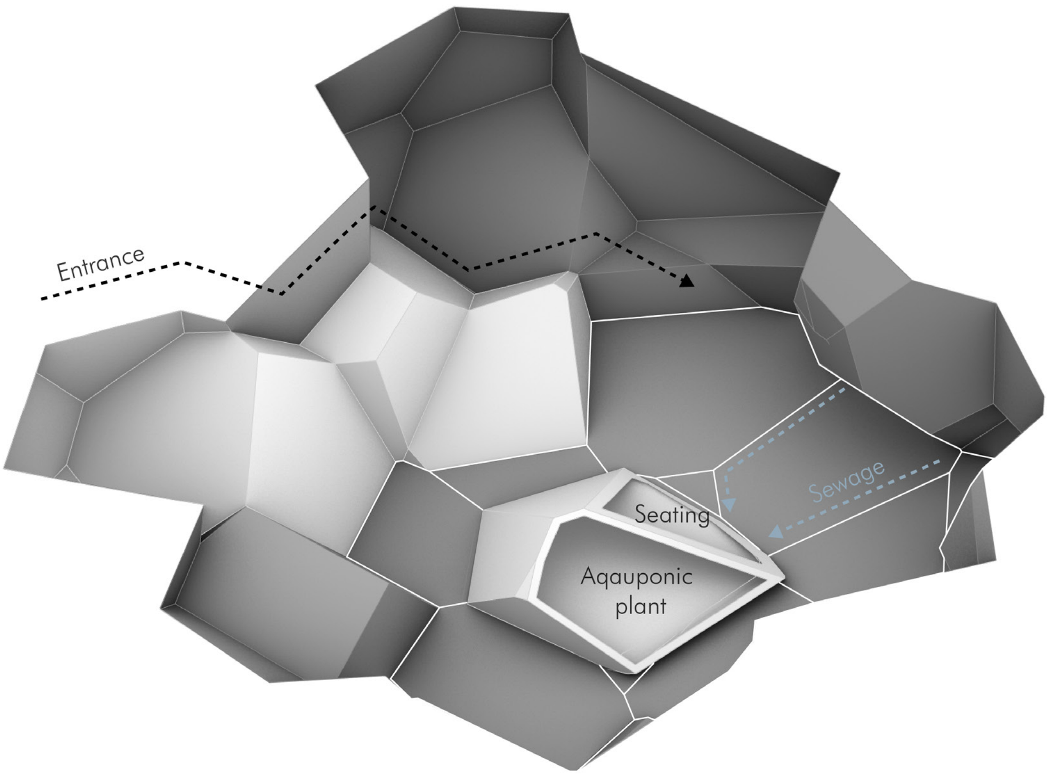


Fig. 16. Functions in the courtyard

Visualization

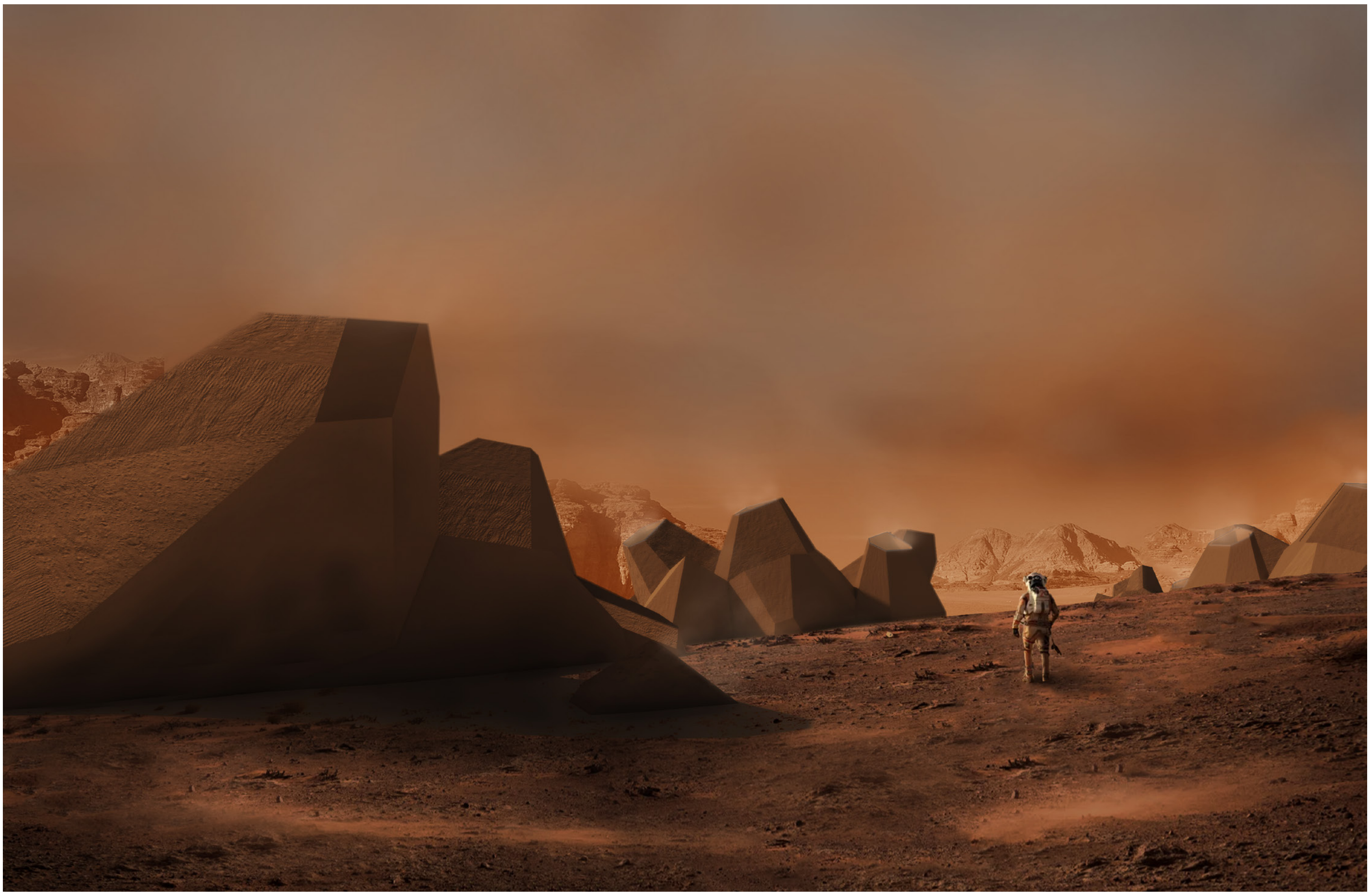


Fig. 16. Exterior view of the voronoi habitat

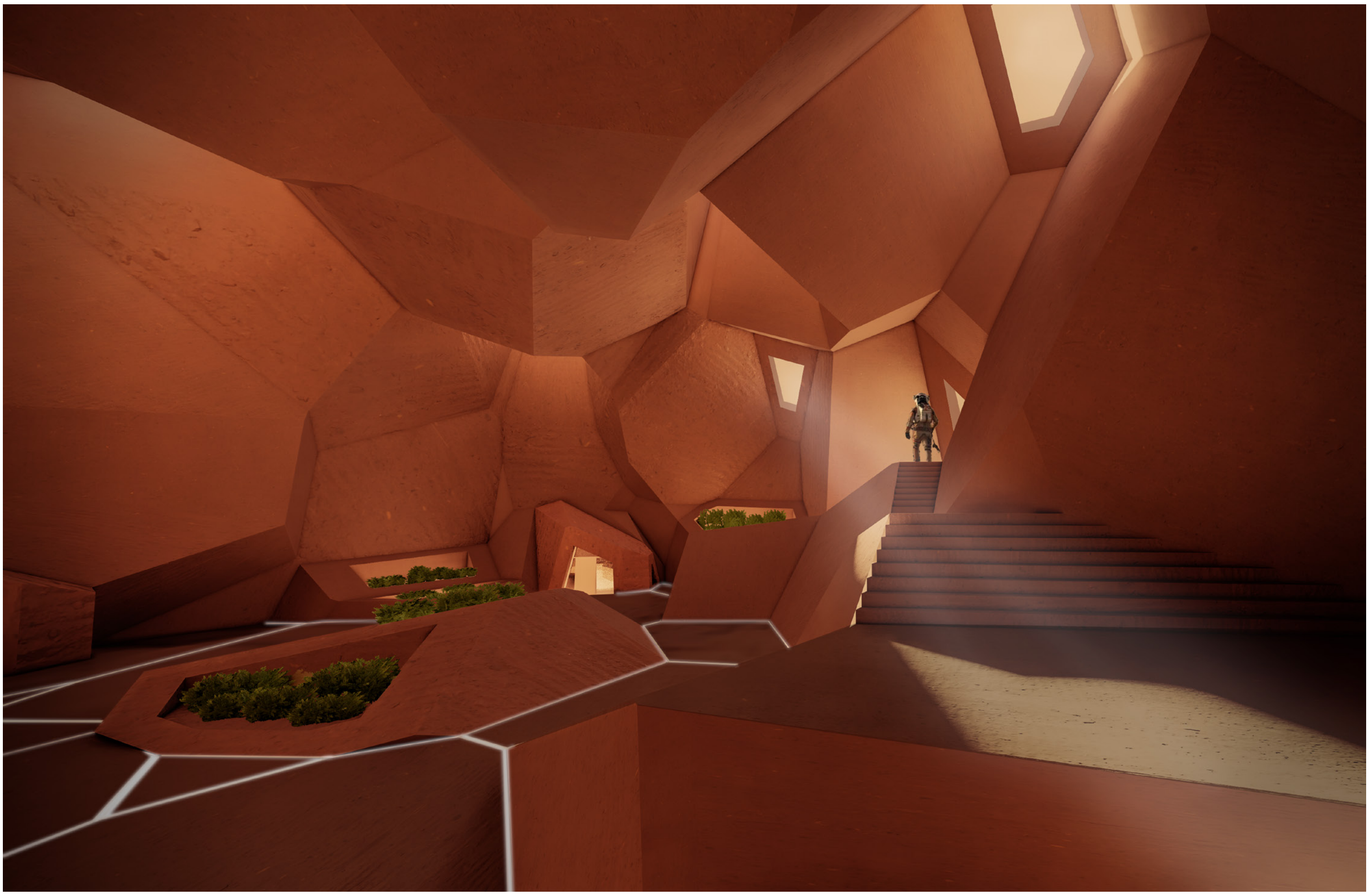
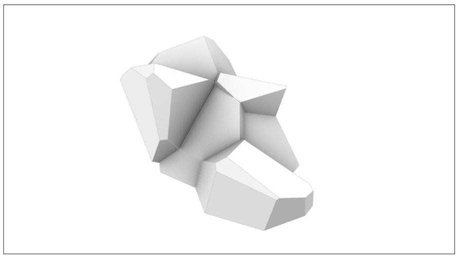


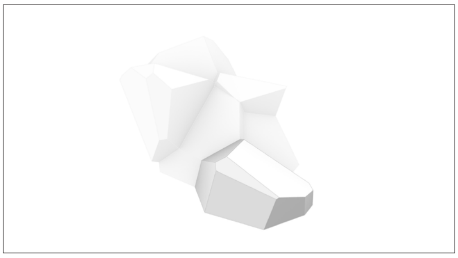
Fig. 16. Interior view of the voronoi habitat

Part IV. Design-to-Robotic-Production(D2RP)

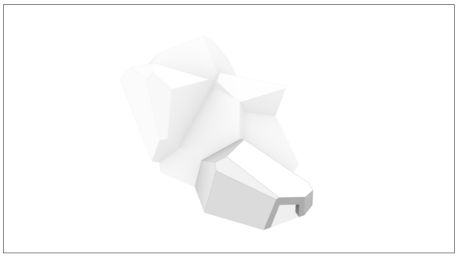
The production step starts by tesselating the habitat and the canopy into smaller voronoi cells such that smaller components can be identified and prototyped. These cells are then grouped together in 3s to fit a form factor. This form factor is determined by the milling step. The bounding box of the initial form that gets milled is the form factor the combined cells are limited to. Once the components are identified the milling pre-processing can begin.



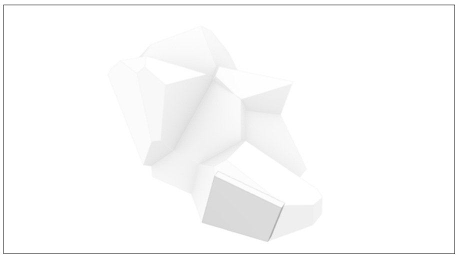
Step1. Selection of house unit



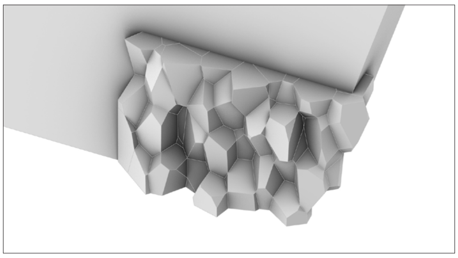
Step2. Selection of individual cell



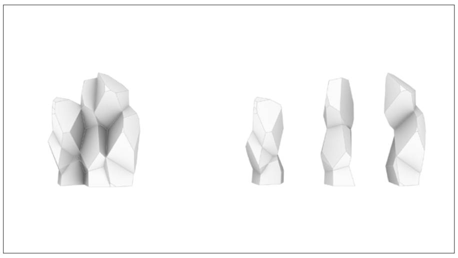
Step3. Shelling of selected cell



Step4. Selection of individual wall



Step5. Generating component cells



Step6. Selection of 3 components

Fig. 17. Step to generate cell components from the house unit

The first step of the milling pre-process is to find the right orientation of the component and create a 'vacuum-seal' mesh for each side; This ensures the milling tool can reach all faces of the component without obstruction. We approach the components from 2 sides, each time milling away half of it leaving the final shape behind. To speed up the process we first want to remove material in large quantities till we have a shape resembling the final form. Then a slower more precise milling pattern can be made to retrieve the final form. The pattern is called a tool path and we generate them in Grasshopper3D.

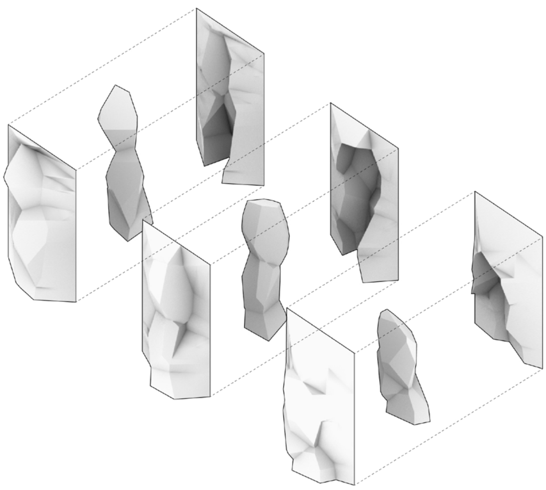


Fig. 18. Meshes for generation of drilling tool path

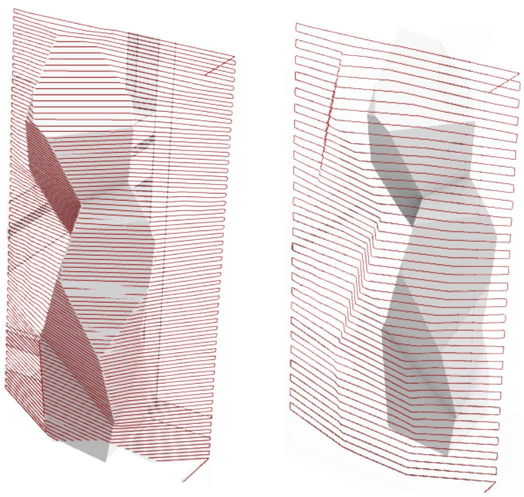


Fig. 19. Precise tool path(left) near the component, and loose tool path(right) to remove excessive foam

The robot that is used for this process is from KUKA and is a 6 axis production robot with no safety sensors. Many endfactors can be attached depending on the task, in this case we have a drilling mill that can drill about 3 cm of foam at a time. Because there are no safety sensors a good workspace is necessary around the robot, and the object the robot can collide with should be modeled and visible when creating the toolpaths in Grasshopper3D. A collision check should be made on every tool path generated to ensure no damage will be made to either the component, the environment, or the robot. It is important that the initial non-milled block is placed in the same coordinate location as the computer modeled bounding box in the reference frame of the robot. Also ensure that the workflow is such that the milled block doesn't need to be moved during any part of the process so that any mis-alignment issues don't ruin the component.

For the fast, low-resolution, material removal we create uniform steps starting from the outside and working our way inwards each time removing about 1 cm of material. Each layer of material that is removed fits the final form of the component more than the previous till the final layer is very close to the final form. We only mill one side of the material for now. Then an additional pass is made with finer detail. The same process is repeated on the other side leaving behind the component.

For acoustic reasons we might want to add additional detail to the faces of the component. For that, we create additional tool paths generated for each face. We want to ensure that where the components come together the surface area is the greatest for the tightest fit and therefore exclude these faces from any process that changes their topography or texture.

Finally for the Human Robotic Interaction that will come later, we need to create 2 holes that will be used for assembling the components together. For this, we choose 2 faces that are suitable and create toolpaths that deepen the topography creating a gripable edge.

Fig. 20. Texturization and production of holes in component

Part V. Computer Vision(CV)

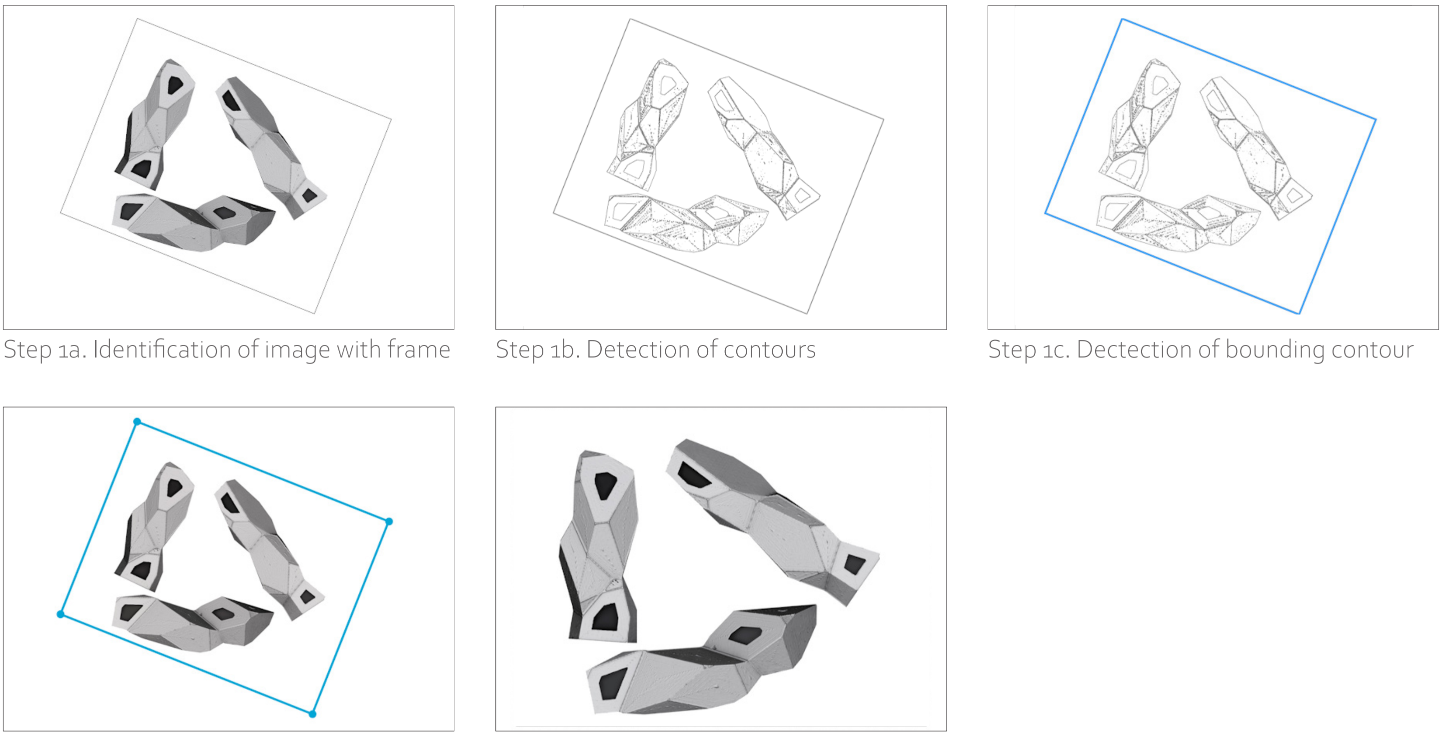
Computer Vision (CV) is an important component to working with robots that need to interact with their surroundings. Robots that blindly follow commands such as the one used for the component milling don't use computer vision, but robots that are intended for more complex and responsive interaction with their surroundings do. There are more sensors that are important for a robot to interact with its environment and we will see more of this kind of robot in the HRI chapter. CV allows for the robot to receive an understanding of the situation so that decisions and commands can be made autonomously. In this project we want that the robot can identify the components, identify where they can be grabbed, and then have it grab the components and assemble them. For this we need to create a CV algorithm. We will be using two libraries, the IAPC library which is based upon the CV2 library.

CV process

Aligment of real and virtual image

To start the algorithm we need a practice image that simulates the conditions the robot will need to face; a white table with white foam components that are roughly milled. The table has a frame outline around the components and an image is taken of the components from above. The frame should be identifiable and complete in the image and ideally, the frame should fill the image and be aligned such that the table coordinates are conventionally orthogonal. (step 1a)We then insert the image into our CV workflow and process it in a way that returns meaningful inputs for the robot.

An edge contour algorithm is run on the image (step 1b). In this case, it is the Canny edge detector algorithm. In this algorithm, a gradient of contrast is used to determine an edge. The difference in gradient is a parameter that can be fine tunes to match the image. On the newly created edges we run a contour finding algorithm called findContours and we set it to find only the most external contours (step 1c). There are multiple setting that can be defined for findcontours but we want the most external contours because we want to identify the edge of the frame. The contour is returned with a lot of vertices and line segments, this is unnecessarily complicated so we match the points to a box polygon (step 1d). This polygon and the image are warped to fix any misalignments between the frame and the camera (step 1e). The pixels of the image can then be converted into physical units using the frame as a reference object. At this point the image can already be used to guide any robot that is also connected to the coordinate system of the frame, it just won't be fully autonomous. We will continue to process the image so that a robot can interact with the components more autonomously.



Step 1a. Identification of image with frame

Step 1b. Detection of contours

Step 1c. Dectection of bounding contour

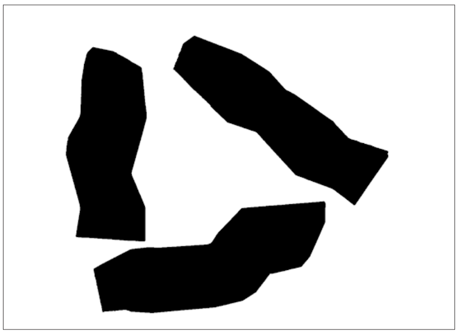
Step 1d. Contour into polygonal box

Step 1e. Alignment of frame to computer

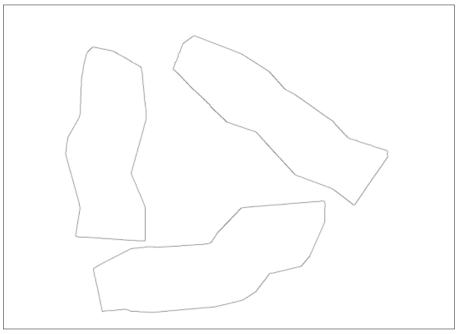
Fig. 21. Python steps to align and calibrate real and virtual image

Isolation of a single component

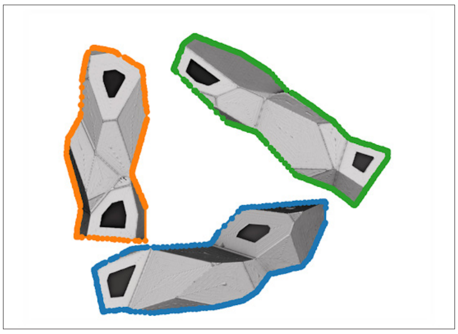
We first isolate the components by using a grayscale contrast threshold; Any value that is whiter than 240 is made white, and any value darker than 240 is made black (figure 2a). Running an edge detection and a findcontours becomes trivial after this step (figure 2b and 2c). The contours are simplified to make the later steps cleaner (figure 2d). From here we just select a contour and use it as a mask to isolate the respective component (figure 2e).



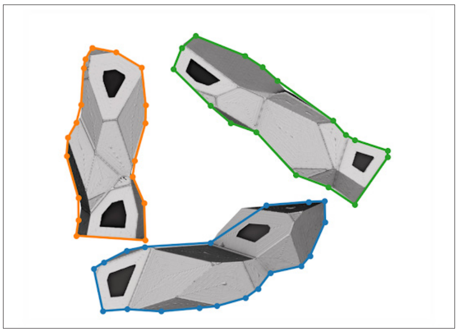
Step 2a. Conversion of component image into black figure



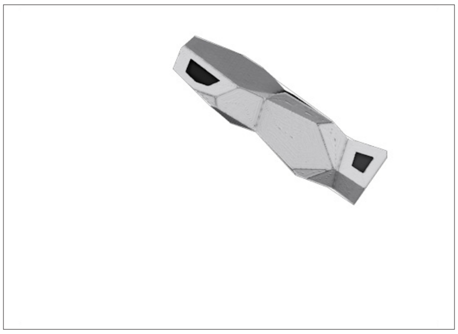
Step 2b. Detection of component edge



Step 2c. Detection of component contours



Step 2d. Simplification of contours

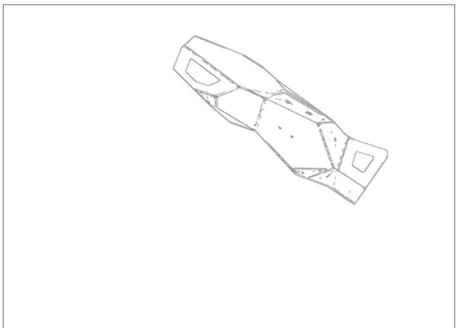


Step 2e. Isolation of single component

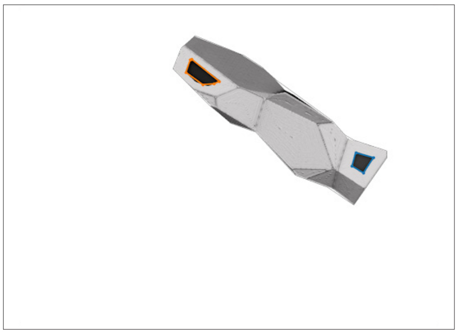
Fig. 22. Python steps to isolate a single component

Dectection of holes and generation of grabbing vector

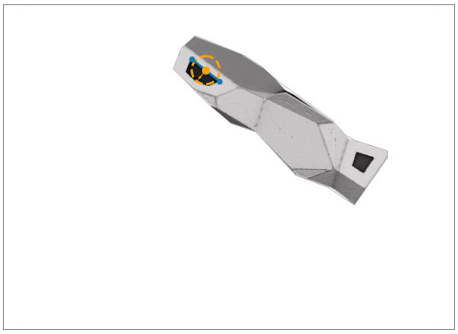
We run an additional edge detection algorithm and again the findcontours (figure 3a). In this step a bluring of the image before the edge detection can be useful to reduce the complexity of the edges and eliminate false gaps. We identify the holes by looking for contours with an area between two values (figure 3b). And finally, as inputs for the robot we create a grabbing vector. We do this by creating a vector between 2 points, the contour centroid and the midpoints of the longest edge of the contour (figure 3c). .



Step 3a. Detection of component contours



Step 2b. Detection of component holes



Step 2c. Creating grabbing factor

Fig. 23. Python steps to detect holes and generate grabbing vector

Part VI. Human-Robot-Collaboration

After the components are recognized through computer vision, now they can be integrated in place through Human-Robot-Collaboration(HRC). According to the experiences, human, robot, camera, computer and components are what needed for HRC.

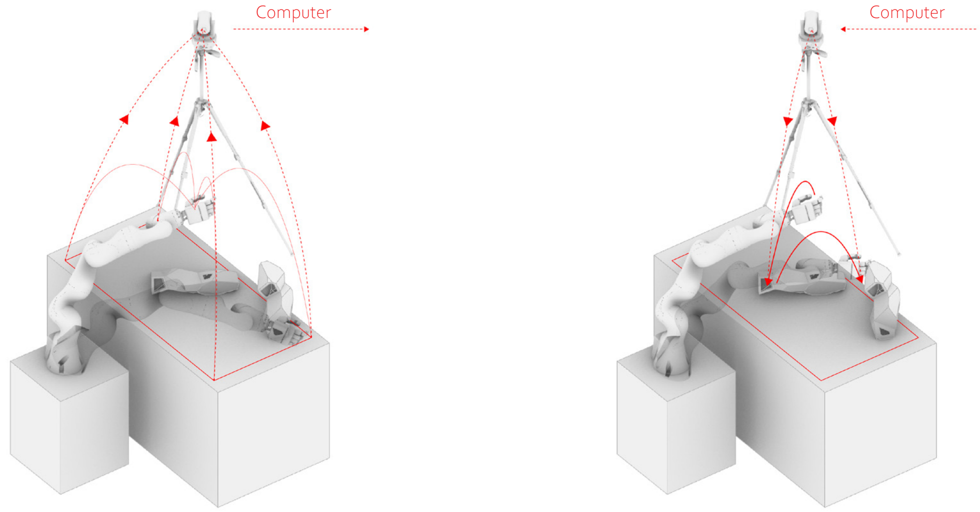


Fig. 24. Relationships between robot, camera, computer and components

Steps of HRI and instructions for robot

1. The robot needs to know the exact location of the frame and table. This can be done by directing the robotic arm(hand) to the vertices of the frame, and marking the location of corresponding vertices in images captured by the camera in the computer.

2. For safety, the robot needs to know its working area. Therefore we define certain mid-air node points, which direct the moving path of the robotic arm along those points to control its movement. Also, we should also limit its moving speed to reduce damage in any accidental cases.

3. Robot needs to know both the exact and relative position of the cells to integrate them. For example, to move a cell in the right towards the cell in the left, the hand of the robot should grab the right side of the components to prevent crashing of the arm to the left cell. Moreover, the robot should slow down when it is approaching the target cell.

4. As in-accuracy occurs during the translation of 3D vision in camera into 2D control frame in computer, pointing the component hole in the computer does not bring the robotic arm to the exact location of the hole. The robot hand stays above the holes, and calibration of height error needs to be instructed with human collaboration.

5. After the robot hand reaches the hole, it needs to be instructed by humans regarding how to grab the hole, and how much force to grab and lift the component.

Fig. 25. HRI process in the experiment

Reflection in the practical assignment

We have seen that there is a large difference between the two KUKA robots, one used in the milling process and one used in the HRI process. The HRI robot is very responsive to external influences. The safety measures set in place to avoid causing harm were demonstrated to work in our practical despite it always being a false call. The versatility and control of the motion of the HRI KUKA was also demonstrated. We saw that different axes of motion can be isolated for easy manipulation. In our practical we saw that the settings of the KUKA were not perfect and it didn't hold objects perfectly and sometimes it succumbed to its own weight thinking that something was pushing it downwards. Here again we saw that the robot did not fully know its surroundings, when asking the robot to grab components there was a point where the robot joints were positioned a bit awkwardly and the lower arm collided with the table, requiring us to lift up the joint. This was likely because the scenario was quickly set up. We attempted to implement our CV code into the system but the code was not resilient enough to deal with the low contrast of the styrofoam component and the white background we made for it. 2 things could have been improved to make the situation better, have a more contrasting background or colored component pieces. The drilling of the holes in our components could have been drilled after the components were given a color, in that way the contrast between the holes and the component itself would be greater. The color of the component could also allow for greater contrast with white background.

Robot to improve construction tasks and its limitation

As we have learnt in the course, robots do what humans can do at their maximum by overcoming the physical limitations of humans. In this sense, robots bring a lot of benefits to the architectural industry. Firstly, it can handle tasks that are dangerous, physically demanding or require high precision. For example, robots can lift the heavy regolith components during construction in reality, and manufacture the precise polygonal shape of each component. Secondly, robots don't tire like humans do. They can handle tasks repetitively in a faster and more accurate way while reducing error, such as how they can produce voronoi components non-stop in the D2RP process. This can help to increase the overall productivity of construction projects and reduce the physical workload on human workers. Thirdly, robots can work for tasks in different scales, such as both the texturization of component cells in small scale, and the construction of large-scale voronoi structure. By using robots to handle some of the physical tasks involved in construction, workers can focus more on the intellectual and supervisory aspects of their jobs. This can help to improve the overall quality of the work being done, while also reducing the risk of injuries and accidents on the job site.

Meanwhile, robots currently have certain limitations as they are not intellectuals in nature. They cannot think, but rely on human instruction. In the HRI process, remarkably in-place human assistants are required. For example, they need human instruction to identify the working area and to grab the holes, while humans also need to present to avoid any errors and accidents. This illustrates that complete replacement of humans in on-site-work is not yet possible. Moreover, robots perform blindly by algorithm, but not by thinking of the situation independently. Therefore it can not duel with any unpredicted and uncertain situation. This is why it's important to introduce robots as assistants to human workers rather than trying to replace them entirely. In this course, we understand that human-robot-communication is achieved through a computational programme, in which we develop the skills to translate humans' thinking into precise data by using digital models, drawings and scripts, and let robots to maximize and perform our thinking in reality. This collaboration mode of human thought and robot do,will be conducive to the future tasks for architectural development.

HRC for other construction tasks

In regards to human-robot collaboration the tasks which seem to have the most potential for this kind of cooperation are usually physically demanding and repetitive. There are a variety of tasks in the construction industry which would benefit from such cooperation.

One example task for future use of HRI is bricklaying or paving, which is highly precise but repetitive work, and where workers are often prone to injuries over working long hours in unfavorable conditions. The instructions for robots can be relatively simple as bricklaying is repetitive work use of robots, while the use of robots can improve the precision of brick structure, which reduces changes of damage and structural failure. Meanwhile, robots can also assist in laying out complicated and decorative paving patterns according to digitally fed design, while humans can concentrate on tasks such as providing material for the robot.

Another example is that HRI can be applied in parametric construction. The complexity of parametric design makes human construction challenging and time consuming. The precise understanding of 3D space by robots offers the potential to construct complicated and varying parametric components. This can result in better craftsmanship and improve the elegance of parametric design.

Part VII. Conclusion and suggestions for further investigations

To conclude, we experienced the complete workflow of Design-To-Robotic-Production and Assembly. We firstly designed the martian voronoi community habitat through digital measures, and studied how to realize the design through D2RP, CV and HRC. This process provided valuable insight into the application of digital and robotic assistance on architecture design. In each process, the tasks were done in a simplified way on a small prototype scale. The challenges identified in each simplified task gives insight into further exploration direction to apply D2RP&A in the complex situation in reality.

Firstly, for the voronoi design, the design process illustrates opportunities to create houses with customized space cells in an integrated structure. However, as each cell depends on each other, flexibility is a concern. It is relatively controllable for individual houses which only use a few cells. The inflexibility would be more illustrated when we envision larger housing complexes constructed with a large number of cells, as it would be hard to maximize the function of each cell, which is interdependent and competing with each other. Also, it would be more complicated to evaluate which cell option, generated by grasshopper script, is better, as there are too many variations in each option. Therefore, more advanced techniques to integrate and maximize functions of more cells can be further inverstigated.

Afterwards, computer vision relies on detecting the outline of target objects(cells). In the Martian environment, the dust, lack of light and environment shadow can blur the vision of the object, and obstruct the computer from detecting the objects' outline. In our working progress, manual processing of cell photos taken in the actual environment is required for the script to detect the cell. Meanwhile, inaccuracy also occurs during the conversion of 3D images into 2D data. Therefore, improvements in python script, such as more accurate outline detecting method using convolution, and technology of 3D scanning, can be incorporated for the future use of Computer Vision.

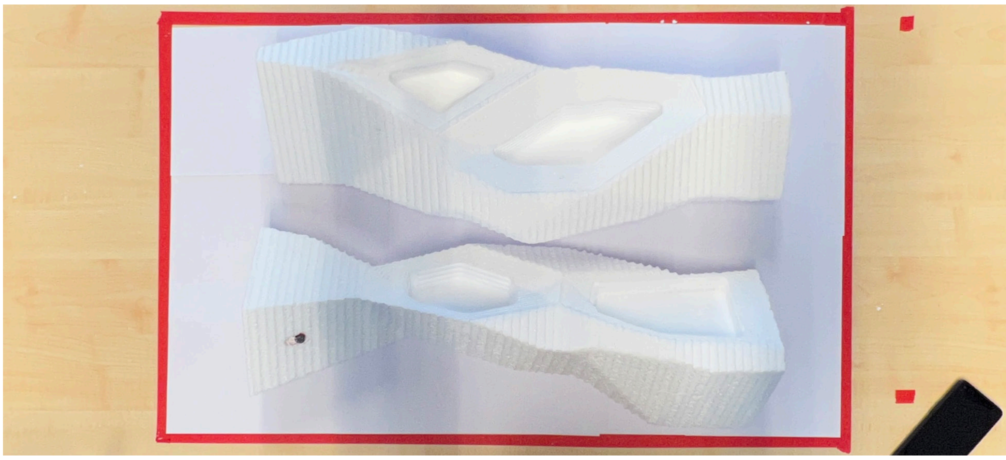


Fig. 26. Shadow in reality obstruct the detection of voronoi components

Lastly, as mentioned, the Human Robot Interaction demonstrated that robots still require human assistance. This challenges its application because robots on Mars have to adapt to the conditions themselves, when the help of humans is limited or avoided. As human instructions are always necessary for robots, apart from improving the independence of the robot itself, human-robot-interaction can be also considered in a way to replace human's in-place instructions with remote instructions.

Nevertheless, with the great benefits brought by computational and robotic design, it is worthwhile to further investigate its application in reality.

References

Nichols, M. R. (2017, May 3). If we successfully land on Mars, could we live there? Astronomy.com. Retrieved April 17, 2023, from https://astronomy.com/news/2017/05/could-we-live-on-mars

Radiation. Marspedia. (2023). Retrieved April 17, 2023, from https://marspedia.org/Radiation#:~:text=The%20 average%20natural%20radiation%20level,times%20the%20average%20on%20Earth.

The Rover's temperature controls. (2023). NASA. Retrieved April 17, 2023, from https://mars.nasa.gov/mer/mission/rover/temperature/#:~:text=In%20fact%2C%20the%20ground%20temperatures,degrees%20 Fahrenheit)%20per%20Mars%20day.

NASA. (2019, May 17). What does a sunrise-sunset look like on Mars? – NASA solar system exploration. NASA. Retrieved April 17, 2023, from https://solarsystem.nasa.gov/news/925/what-does-a-sunrise-sunset-look-like-on-mars/

Rhizome 1.0: Development of an autarkic design-to-robotic-production and -operation system for building offearth habitats. Robotic Building (RB). (2023). Project Team in TuDelft. Retrieved April 17, 2023, from http://www.roboticbuilding.eu/project/rhizome-development-of-an-autarkic-design-to-robotic-production-and-operationsystem-for-building-off-earth-habitats/

Taylor, A. (2018, February 27). The last families living in Tunisia's underground houses. The Atlantic. Retrieved April 17, 2023, from https://www.theatlantic.com/photo/2018/02/the-last-families-living-in-tunisias-undergroundhouses/554426/

Mars ice house. (2023). Space Exploration Architecture. Retrieved April 17, 2023, from http://www.spacexarch.com/mars-ice-house