

Automated Shingling



Team 1, Robot Autonomy (16-662), Spring 2016
Eitan Babcock, Dan Berman, Sean Bryan, Rushat Gupta Chadha,
Pranav Maheshwari

Table of Contents

The Problem.....	2
Background.....	2
What we're doing.....	3
Surface Tessellation.....	3
Shingle Preparation.....	4
Trajectory Generation.....	4
Work Object Definitions.....	5
Final Product.....	6
References.....	6

The Problem

The dFAB lab in the architecture school of Carnegie Mellon University focuses on utilizing cutting edge technology to advance the architect's role in the modern world. As part of this effort, this project aims to provide the dFAB lab with the means of generating, visualizing, and implementing shingle patterns on complex surfaces. This relates to the existing workflow of the lab.

Background

Shingling is currently a very manual labor intensive process. Shingles are purchased as rectangular pieces of wood that must be arranged by hand on a wall or roof. There are a wide variety of surfaces that can be shingled, and many different shapes that can be created.

The most basic shingling is a standard straight row shingle on a flat wall or roof with equal sized shingles, as shown in Figure 1.



Figure 1: Basic shingle pattern



Figure 2: Complex shingle pattern

This is the easiest type for planning and automated execution. Shingles are simply placed next to each other in a row, and then each row overlaps with the row below it. More complex shapes can also be created by varying the shingle placement, as well as varying the surface being shingled, as seen in Figure 2.

This project focusses on this more interesting case. It requires analysis of the wall to be shingled, segmentation to determine shingle sizes and placement, shingle placement planning, and then robotic execution. The robot trajectories are generated relative to the work objects, which in our case are the surface which is being shingled and the pick point for the shingles.

What we're doing

To complete this project, we used an ABB IRB 4400 robotic arm, a custom end effector, Rhino CAD with the Grasshopper add-on and Hal plugin, and ABB Robot Studio. All of these were provided by the dFAB lab.



Figure 3: ABB IRB 4400

Rhino3D is a CAD environment that can be used to work with NURBS curves and surfaces with a mathematical model at the backend, allowing easy manipulation and processing of these elements. Grasshopper is a Rhino plugin which supports visual programming, Python and can be used to perform actions on various Points, Curves and Surfaces. HAL is an add-on within Grasshopper which supports in controlling various robotics platforms by ABB, Kuka, and Universal Robots.

Surface Tessellation

The process of tessellation is essentially breaking down a larger surface into smaller geometric shapes. In our case, this process was driven by various parameters such as shingle overlap, shingle gap, width of shingles and length of shingles. Since the surface being used was curved, the algorithm also had to consider the gap between a shingle and a surface, and keep it within some limits.

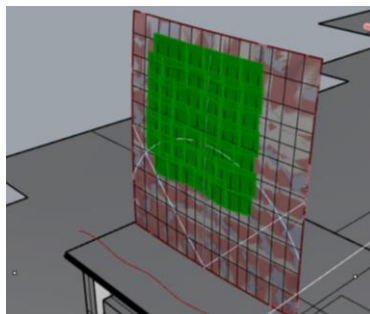


Figure 4: Shingles overlaid on the Surface

Our algorithm discretizes the surface and starts from the bottom right corner. These points are then processed and joined till the maximum permissible size of a shingle, based on all the parameters, is reached. Since the shingle is a flat object and the surface is curved, normals from the surface are made to intersect with the shingle at each step to ensure a maximum gap threshold between the surface and the shingle isn't crossed. Once a row of shingles is complete, processing for the next row starts. This is an iterative process which continues till the extremas of the surface are reached.

Shingle Preparation

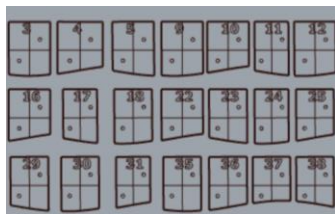


Figure 5: Shingles laid out in grid



Figure 6: Final Shingles

Once all the shingle surfaces are created, they need to lay out in a pattern such that they can be fed into the laser cutter. The shingles are also given a label and have two holes drawn through them so that they can be placed on the shingle. To achieve this we reoriented all the shingles and moved them to multiple grids which are directly fed into the laser cutter as seen in Figure 5. The shingles had a maximum length of 12", maximum width of 11", and thickness of 0.125". The final fabricated shingles can be seen in Figure 6.

Apart from this method, there was an alternative way to approach this problem where the robot interacts with a cutting tool and cuts the shingle. This approach will potentially lead to a more streamlined workflow and avoid the use of a laser cutter.

Trajectory Generation



Figure 7: Tool training

Since we used a custom end effector, an important step is training the tool and modelling it in software. To do this, the robot with the custom end effector, is jogged to the same point in the world with 4 different joint configurations (Figure 7). This basically creates 4 spheres (as solutions to the points which satisfy the point) the intersection of these 4 spheres is where the tool frame is placed. The volume of the small intersection space is the error in the tool end effector, which in our case was about 0.4 mm. This generates a homogeneous matrix to be ported to the simulation so the robot solves the inverse kinematics accordingly.

We chose to generate waypoints based on the output of the shingle code and decided to join those way points in straight line without changing the end effector orientation. A total of 6 waypoints were generated for each cycle (Figure 8). The sequence for one cycle is as follows -

1. Point Above Shingle Dispenser
2. Shingle Pick Point
3. Back to Point 1
4. Point Above Foam
5. Place Point on Foam
6. Back to Point 4

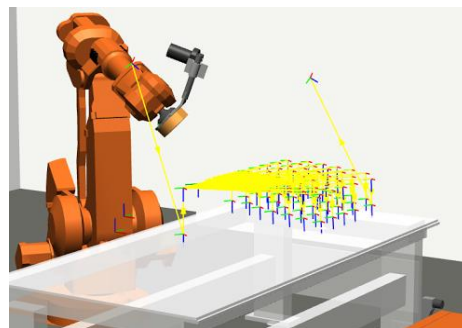


Figure 8: Complex shingle pattern

Work Object Definitions

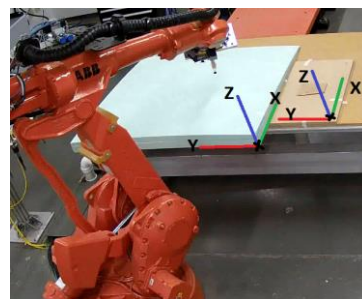
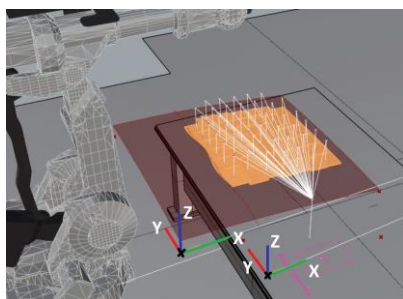


Figure 8: Defining work objects in Rhino (left) and trained origin and coordinate frame on the robot (right)

Since the objects in simulation (foam and shingle dispenser) won't be in the same location or orientation while working with the actual robot, it is important to associate waypoints to these reference objects. To do this, an origin, X axis and orientation was defined on the work objects (Figure 8). After defining this, for every cycle, 3 of the way points were associated with the shingle dispenser while the other 3 are associated with the foam.

Once every waypoint is associated with a work object, the next part is training the coordinate frames on the work objects. This is done by manually jogging the end effector to 3 points (origin, x axis, y axis). This gets rid of all the error that can accumulate because of improper placement of objects.

Final Product

With the inputs for gap, overlap and other parameters provided to us by dFab, the following was achieved on a double curved surface. The video guiding through the pipeline is attached in the references [3].

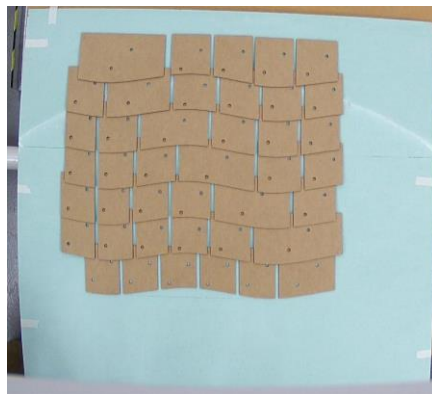


Figure 9: Final Shingled Surface

References

[1] "Cedar Shingled Wall or Roof Section." Cedar Shingled Wall Or Roof Section Stock Photography. N.p., n.d. Web. 04 May 2016. <http://www.dreamstime.com/stock-photography-cedar-shingled-wall-roof-section-image16299182>

[2] "Zinc Shingles, Copper Shingles, Stainless Steel Shingles." SIG Zinc Copper. N.p., 16 Sept. 2015. Web. 04 May 2016. <http://www.sigzincandcopper.co.uk/zinc-shingles-copper-shingles-stainless-steel-shingles-what-you-need-to-know/>

[3] Autonomous Shingling by an ABB IRB 4400 Arm - dFab CMU - https://www.youtube.com/watch?v=rDI49Gicg_g