Generating Shape-Based Building Instructions in Minecraft

Marisa Hudspeth Rhodes College hudmj-22@rhodes.edu

Prashant Jayannavar **Liliang Ren** **Julia Hockenmaier** University of Illinois at Urbana-Champaign {paj3, liliang3, juliamr}@illinois.edu

Abstract

The goal of our research is to develop two interactive agents: an Architect (**A**) which instructs a Builder (**B**) how to build a target structure in Minecraft. We have created a program which generates natural and varied dialogues between **A** and **B**. **A**'s instructions to **B** reference the sub-shapes of the target structure. We hope to use these artificial dialogues to improve our models for both **A** and **B**, which have only been trained on human-human dialogues.

1 Introduction

We consider the Minecraft Collaborative Building Task introduced by Narayan-Chen et al. (2019), in which an Architect (**A**) instructs a Builder (**B**) to create a target structure out of blocks in Minecraft. Narayan-Chen et al. (2019) collected 509 dialogues between human Architects and Builders (termed the Minecraft Dialogue Corpus). These human-human dialogues refer to the target structure in a variety of ways, ranging from high-level instructions referencing the entire target structure ("flower" or "bell tower"), mid-level instructions referencing sub-shapes of the target structure ("row" or "plane"), and low-level instructions referencing individual blocks. In this paper, we focus on generating artificial dialogues that use the mid-level, shape-based Architect instructions. It is our hope that these dialogues can be used to improve the Architect utterance model (Narayan-Chen et al., 2019) and the Builder Action Prediction model (Jayannavar et al., 2020).

2 Background

Previous work has focused on training the Architect utterance model (Narayan-Chen et al., 2019) and the Builder Action Prediction model (Jayannavar et al., 2020) using the human-human dialogues of the Minecraft Dialogue Corpus. However, the small size of the dataset and the complexity of the human-human dialogues led to difficulties in training accurate models.

Narayan-Chen et al. (2019)'s Architect utterance model can identify the colors of blocks correctly, but struggles with spatial relations. In addition, the model can only generate simple, block-by-block instructions. Similarly, Jayannavar et al. (2020)'s Builder Action Prediction model also does well with color identification but is less accurate with numbers and spatial relations.

To improve each models' performance, we have developed a program to generate artificial dialogues between **A** and **B**. These dialogues are obviously not as natural as human-human dialogues; they are simpler, following a set of formulaic templates, and lack the chit-chat of the human dialogues. However, without the "noise" and irrelevant utterances present in the human dialogues, we hope that the artificial dialogues will allow the models to more easily learn the basic elements of giving/following an instruction, especially concerning spatial relations and numbers.

3 Dialogue Generation

3.1 Shapes

The dialogue generation requires a set of target structures to first be created. Each target structure can be composed of any mixture of rows, diagonals, planes, or prisms. This dataset is generated using a program developed by Lambert et al. (2019).

The target structure is then divided into sub-shapes, and at least one instruction is given by **A** per sub-shape. **A** may give two instructions for a single shape if there is too much information to convey using only one instruction. **B** does not always respond to **A**. Sometimes, **B** will acknowledge they have finished building the structure or will ask a clarification question if information is missing from **A**'s instruction. This is intended to mimic the back-

<Builder>	Mission has started.
<Architect>	let's build a blue diagonal going away from you
<Builder>	how big is it?
<Architect>	three by three
<Architect>	so please create a row of yellow blocks 1 block to the right and 1 block behind the last block
<Builder>	how long should it be
<Architect>	4 blocks

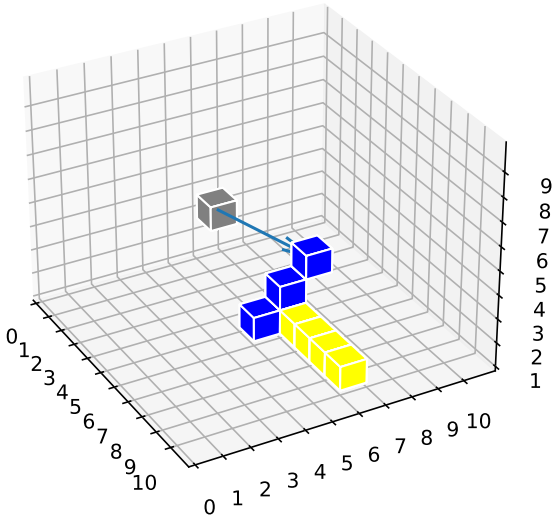

Figure 1: Example dialogue and 3D plot for a row and diagonal

and-forth nature of the human-human dialogues.

The instruction templates vary depending on the type of shape being considered. Shapes can be differentiated by their position in the overall build order (whether they are the first shape being built), their dimensions (1D, 2D, or 3D), and their specific shape name.

3.2 Architect Utterances

An Architect's instruction is composed of several elements, which can be arranged in different orders to produce a variety of final instructions. Some of these elements (orientation, direction, dimensions, and color) may be left out of the instruction, as humans do not always include all relevant information.

Introductory Phrase "Intro" phrases are filler words that appear at the beginning of sentences, such as "okay," "next," and "great." It is also possible for this phrase to be an empty string, since not all sentences will start with a filler word. Some phrases are only appropriate for the first shape being built ("first"), while others only work if another shape has already been built ("next," "then," "perfect").

Verb Phrase These can range from simple commands like "make" or "build" to more complex phrases such as "can you add," "you'll want to start with," or "let's create." Again, some phrases can only be used for the first shape being built ("begin by making," "to start with, build").

Orientation The orientation of a shape refers to its position in space relative to the last shape that was built, from the perspective of **B**. The first shape being built, then, will not have any orientation text, since there is no other shape to compare it to. When a shape's position differs from the previous shape in only one or two dimensions, the resulting orientation phrase will be short enough to incorporate into **A**'s final instruction. Examples of such short phrases include "3 up from the last block you placed" and "two blocks to the right and one behind that." However, if the position differs in all three dimensions, the orientation phrase will be separated into its own instruction to avoid producing an overly long or clunky sentence. A long orientation phrase like "1 to the left, 2 under, and one block in front" could result in final instructions like "it's gonna be 1 block to the left, 2 blocks under, and one block in front" or "place it 1 block to the left, 2 blocks under, and one block in front."

Direction This refers to the direction the shape should be built in from **B**'s perspective. Sometimes this information is especially important; if the target shape is a horizontal row, for example, **B** needs to know whether to build it on the X or Z plane. However, it would be unnecessary to specify the direction that a column should be built in, as the name already implies its direction (upwards). Because of this, the direction text is not always included in the final instruction, depending on the shape. The possible directions a shape could be built in are to the left or right, up (not down, since all shapes are built from the bottom up), and towards or away from **B**. The direction phrase will specify at most two directions in which to build

<Builder>	Mission has started.
<Architect>	now we will begin with a 6x4 wall
<Architect>	Perfect, you will want to build a 4X2X2 prism with purple blocks
<Builder>	where should I put it
<Architect>	1 to the left, 1 above, and one behind the last block you placed

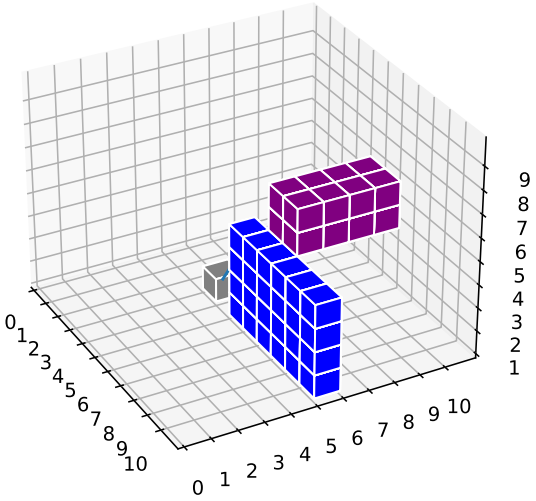

Figure 2: Example dialogue and 3D plot for a plane and prism

the shape to keep the final instruction concise. For prisms, the two directions will be X (left/right) and Z (toward/away). Some example direction phrases are "going to the right of you" and "extending up and towards you."

Dimensions There are multiple ways to describe a shape's size, depending on its own dimensions. For 1D shapes such as rows, the dimension text will only include a single number, such as "four blocks long" or "3 blocks tall." 2D shapes have more variation. They can use any of the following formats: "2x3," "2X3," "2 by 3," "two by three," or "two blocks long and 3 tall." Finally, 3D shapes use the formats "2x3x4," "2X3X4," "2 by 3 by 4," and "two by three by four."

Other Shape Information The final instruction will also include the shape's color and name. The core shape names are row, diagonal, plane, and prism, but these can be replaced with appropriate synonyms. A horizontal row may become a "line," while a vertical row may become a "column" or "tower." Similarly, a horizontal plane could be changed to a "layer," whereas a vertical plane could be a "wall." A horizontal diagonal can only be described as a "diagonal" or "diagonal line," but a vertical diagonal can also be called a "staircase" or "stairway."

3.3 Builder Utterances

B can respond to **A** in two ways. If no information is missing from **A**'s instruction, **B** has a low (adjustable) chance of indicating that they are finished building the shape. **B** may say "I'm done," "okay," "finished," or similar phrases. However, if information is missing from the instruction, **B** has a higher chance of asking for clarification. Whether information is missing from **A**'s instruction is hard-coded so that we can produce the correct response from **B**. **B** may ask about the color, size, orientation, or direction of the shape. Some example questions about the shape's color are "what color," "what color is it," "what color blocks," and "what color should it be." Questions about size, orientation, and direction have a similar set of variants. There is a 50% chance that a question mark will be added to the end of a question.

3.4 Statistics

In a dataset with 10,000 dialogues, there were 56,930 utterances (33,922 Architect and 23,008 Builder). The average dialogue had 5.7 utterances. 3.4 of those were Architect utterances, with an average length of 11.8 tokens, and 2.3 were Builder utterances, with an average length of 3.3 tokens.

The associated target structures ranged in complexity, with a minimum of 4 blocks, maximum of 328 blocks, and average of 38.8 blocks. There were 5044 rows, 4918 diagonals, 5008 planes, and 5030 prisms total.

4 Building Target Structures

The order in which **B** places blocks within a shape can also vary, depending on the shape. Some shapes can be built in multiple ways, and certain block orders are more optimal than others.

<Builder>	Mission has started.	<Builder>	Mission has started.
<Architect>	now we'll build a 3X4X2 yellow prism	<Architect>	Okay, now build a rectangular prism that is two by four by three
<Builder>	which way	<Builder>	what color should it be?
<Architect>	extending to the left of and away from you	<Architect>	yellow
<Architect>	ok, now 1 to the right, 1 above, and one in front of the block you just added add a vertical wall that is four by six	<Architect>	now make a 4x6 wall with orange blocks going up and away from you
<Builder>	what color should it be	<Architect>	it should be one to the left, 2 below, and 1 block behind
<Architect>	orange	<Builder>	ok

Figure 3: Randomized Dialogue Comparison

4.1 Starting Block

First, it will be determined which block **B** should start with. Potential starting blocks are always in the bottom corners of the shape. From these, the block that is chosen to be the starting block is the one with the shortest distance to the last block that was placed in the previous shape. If there is no previous shape (so, this is the first shape being built), the starting block will have the largest distance from the first block to be placed in the next shape.

4.2 Block Order

After the starting block is chosen, the overall order in which all the blocks of the shape will be placed is decided. Rows and horizontal diagonals will only have one possible build order - straight out from the starting block. However, planes can be built either by column or by row. In addition, if a plane is being built by row, **B** may also use a zigzag pattern to alternate which side of the shape each row starts on. Prisms are divided into either horizontal or vertical planes, each of which is built using the same technique. These potential build orders were designed to mimic the many ways human builders constructed shapes.

4.3 Intermediate Blocks

Some shapes require that intermediate blocks be placed between each block of the shape and then removed. In Minecraft, a block can only be placed if one of its sides is directly touching the side of another block. Vertical diagonals and floating shapes both need to have intermediate blocks placed during their construction. If a shape is floating, we first attempt to find a neighboring block in another shape that can be used to place an intermediate block. If no neighboring block can be found, a column of intermediate blocks will be built up to the first block of the floating shape and then removed.

5 Discussion

Our artificial dialogue generation has some limitations. Although the dialogues we generate are varied, they still are not as natural as those produced by humans. Currently, we only generate instructions based on a set of simple shapes, but human Architects also consider individual blocks and more advanced shapes.

Another area for improvement involves **B**'s action sequences. Unlike **A**, who sometimes makes mistakes by leaving information out of their instructions, **B** always creates the correct target structure and never misplaces a block. Just as **B** can ask clarification questions of **A**, **A** should also be able to correct any mistakes **B** makes while building. The human dialogues generally have more of this back-and-forth than our artificial dialogues have.

At the moment, we are focusing our efforts on Narayan-Chen et al. (2019)'s Minecraft Collaborative Building Task, but this general idea of developing agents who can give and receive instructions has the potential for much broader applications. One area of interest would be to move from Minecraft into the real world: as robots become more ubiquitous in our homes and workplaces, the need for robots which understand human instruction becomes more apparent.

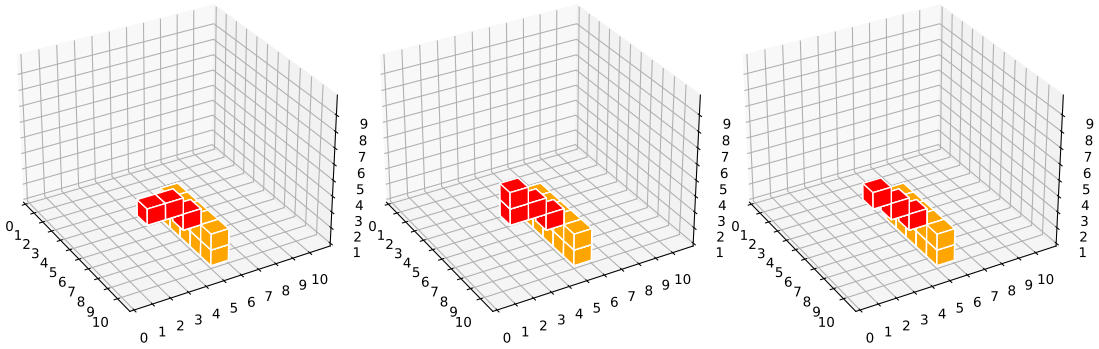

Figure 4: Addition and removal of an intermediate block

6 Conclusion and Future Work

There are a few additional features that could reduce the gap between our artificial dialogues and the human dialogues. Incorporating more variance, additional simple sub-shapes (L's, U's, T's, rings), typos, and the ability for **B** to make mistakes into the artificial dialogue generation would all bring the artificial dialogues closer to the human dialogues.

After refinement of the artificial dialogue generation, the next step is using the artificial dialogues to train the Architect and Builder models. Developing an Architect model that could recognize and create instructions for more advanced shapes (flowers, belltowers, giraffes, etc) would be very difficult, and perhaps impossible considering the amount of shapes there are to account for. By contrast, having **A** generate simple block-by-block instructions is a much simpler task which we are currently working on. We plan to use artificially generated block-by-block instructions in combination with the shape-based instructions discussed in this paper to further improve the Architect utterance model and the Builder Action Prediction model.

Acknowledgements

I would like to thank my mentors, Prashant Jayannavar and Julia Hockenmaier, as well as CRA-WP's DREU Program.

References

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online. Association for Computational Linguistics.

Charlotte Lambert, Ariel Cordes, Elli Kaplan, Prashant Jayannavar, and Julia Hockenmaier. 2019. Virtual world context encoding for grounded dialogue in minecraft. Unpublished.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

A Sentence Templates

Listed are the sentence templates used to generate the randomized instructions. Words surrounded by braces are replaced with appropriate synonyms or phrases.

The shape types refer to whether the shape is the first shape being built. A shape type of "any" means the corresponding template would work for any shape; a shape type of "other" means the template would work for any shape except the first shape.

A.1 Shapes

Shape Name	Synonyms
row_X	row line
row_Z	row line
row_Y	column tower pillar
r_prism	rectangular prism prism cube
square_horizontal	square plane layer horizontal square horizontal layer horizontal plane
square_vertical	square plane wall vertical square vertical plane vertical wall
plane_horizontal	plane layer horizontal plane horizontal layer
plane_vertical	plane wall vertical plane vertical wall
d_horizontal	diagonal line diagonal
d_vertical	diagonal line diagonal staircase stairway

A.2 Synonyms

Word	Synonyms
make	make build create add
making	making building creating adding
start	start begin
place	place put add
placed	placed put added
great	great good perfect cool okay ok
done	done done! I'm done ok okay okay, done finished I'm finished
above	above on top of up from
below	below down from under underneath
toward	toward towards
tall	tall high
extending	extending going
x	X x by

you will	you will you'll
we will	we will we'll
you are	you are you're
we are	we are we're

A.3 Intro Phrase

Shape Type	Template
any	okay, now ok, now now so
first	first
other	next next, then {great}! now {great},

A.4 Verb Phrase

Shape Type	Template
any	{make} please {make} {subj} {make} {subj} want to {make} let's {make} can you {make}
first	{start} by {making} {start} with let's {start} with let's {start} by {making} {subj} {start} with {subj} {start} by {making} {subj} want to {start} with {subj} want to {start} by {making} to {start} with, let's {make} to {start} with, {subj} {make} to {start} with, {make}

A.5 Direction Phrase

Num Directions	Template
1	{extending} {direction1} you
2	{extending} {direction1} and {direction2} you

A.6 Orientation Phrase

Num Orients	Template
1	{d1} block{s1} {direction1} {d1} {direction1}
2	{d1} block{s1} {direction1} and {d2} block{s2} {direction2} {d1} block{s1} {direction1} and {d2} {direction2} {d1} {direction1} and {d2} block{s2} {direction2} {d1} {direction1} and {d2} {direction2}
3	{d1} block{s1} {direction1}, {d2} block{s2} {direction2}, and {d3} block{s3} {direction3} {d1} {direction1}, {d2} {direction2}, and {d3} {direction3} {d1} block{s1} {direction1}, {d2} {direction2}, and {d3} {direction3} {d1} {direction1}, {d2} {direction2}, and {d3} block{s3} {direction3}

Table 1: First, the basic orientation text is determined. d1, d2, and d3 represent the number of blocks. direction1, direction2, and direction3 are the actual orientations (left/right, up/down, front/behind). s1, s2, and s3 are either empty strings or 's' to control whether the word 'block' is plural or not.

Full Template
{orient_text} the last block you {placed}
{orient_text} the block you just {placed}
{orient_text} the last block
{orient_text} that
{orient_text} the {color} {shape_name}

Table 2: The basic orientation text is plugged into {orient_text} and related to the last shape.

Long Orienation Template
it's gonna be {orient}
it's going to be {orient}
it should be {orient}
this is {orient}
{place} it {orient}

Table 3: If the full orientation text is long, it will be placed in a separate instruction inside {orient}.

A.7 Final Instruction

Dimension	Template
1	{intro}{make} {article} {color} {shape_name} {dim_text} blocks {long_tall} {intro}{make} a {dim_text} block {long_tall} {color} {shape_name} {intro}{make} {article} {color} {shape_name} that is {dim_text} blocks {long_tall} {intro}{make} a {shape_name} of {dim_text} {color} blocks {intro}{place} {dim_text} {color} blocks in a {shape_name}
2	{intro}{make} {article} {dim_text} {color} {shape_name} {intro}{make} {article} {color} {shape_name} that is {dim_text} {intro}{make} a {dim_text} {shape_name} with {color} blocks {intro}{make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall}
3	{intro}{make} {article} {dim_text} {color} {shape_name} {intro}{make} {article} {color} {shape_name} that is {dim_text} {intro}{make} a {dim_text} {shape_name} with {color} blocks
diagonal	{intro}{make} {article} {color} {shape_name} {intro}{make} a {shape_name} that is {color} {intro}{make} a {shape_name} with {color} blocks {intro}{make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall}

Table 4: Final templates for any shape (suitable for the first shape or non-first shapes). "Dimension" refers to whether a shape is 1D, 2D, or 3D.

Dimension	Template
1	{intro}{orient} {make} {article} {color} {shape_name} {dim_text} blocks {long_tall} {intro}{make} a {dim_text} block {long_tall} {color} {shape_name} {orient} {intro}{orient} {make} a {dim_text} block {long_tall} {color} {shape_name} {intro}{orient} {make} {article} {color} {shape_name} that is {dim_text} blocks {long_tall} {intro}{make} {article} {color} {shape_name} {orient} that is {dim_text} blocks {long_tall} {intro}{orient} {make} a {shape_name} of {dim_text} {color} blocks {intro}{make} a {shape_name} of {dim_text} {color} blocks {orient} {intro}{place} {dim_text} {color} blocks in a {shape_name} {orient} {intro}{orient} {place} {dim_text} {color} blocks in a {shape_name}
2	{intro}{orient} {make} a {dim_text} {color} {shape_name} {intro}{make} a {dim_text} {color} {shape_name} {orient} {intro}{make} a {dim_text} {color} {shape_name} that is {orient} {intro}{orient} {make} {article} {color} {shape_name} that is {dim_text} {intro}{orient} {make} a {dim_text} {shape_name} with {color} blocks {intro}{make} a {dim_text} {shape_name} {orient} using {color} blocks {intro}{orient} {make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall} {intro}{make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall} {orient}
3	{intro}{orient} {make} a {dim_text} {color} {shape_name} {intro}{make} a {dim_text} {color} {shape_name} {orient} {intro}{make} a {dim_text} {color} {shape_name} that is {orient}

	{intro}{orient} {make} {article} {color} {shape_name} that is {dim_text} {intro}{orient} {make} a {dim_text} {shape_name} with {color} blocks {intro}{make} a {dim_text} {shape_name} {orient} with {color} blocks
diagonal	{intro}{orient} {make} {article} {color} {shape_name} {intro}{make} {article} {color} {shape_name} {orient} {intro}{make} {article} {color} {shape_name} that is {orient} {intro}{orient} {make} a {shape_name} that is {color} {intro}{orient} {make} a {shape_name} with {color} blocks {intro}{make} a {shape_name} {orient} using {color} blocks {intro}{orient} {make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall} {intro}{make} {article} {color} {shape_name} that is {len1} blocks wide and {len2} {long_tall} {orient}

Table 5: Final templates for non-first shapes. These include the orientation phrase.

B Adjustable Parameters

Dialogues can be generated with additional options. The following parameters can be adjusted: whether **B** can ask questions, the chance **B** will speak, the chance **B** will ask a question after receiving an instruction with missing information, the chance **A** will give an instruction with missing information, the chance that the first letter of an utterance will be capitalized, and the seed or range of seeds. If a range of seeds is specified, a set of dialogues for the same target structures will be generated for each seed, resulting in multiple paraphrased dialogues for each target structure.

C Additional Figures

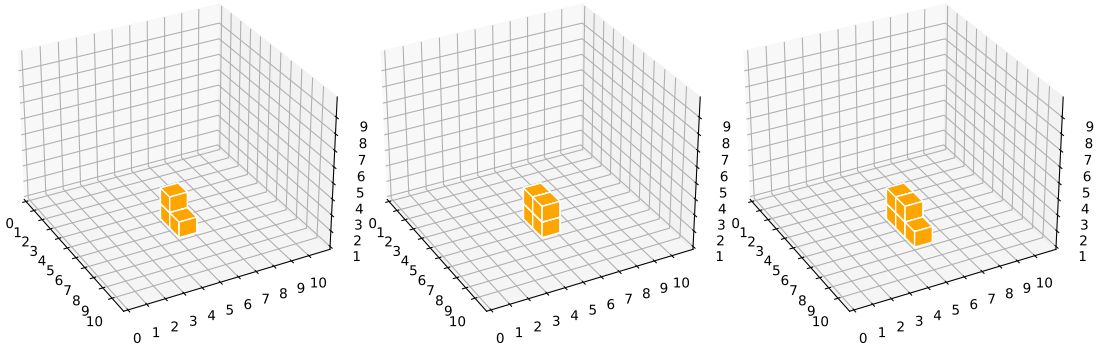

Figure 5: Building a plane by column

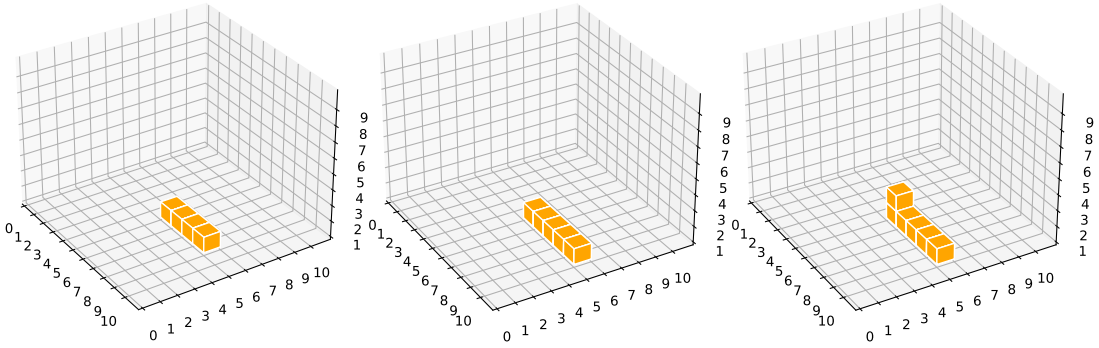

Figure 6: Building a plane by row

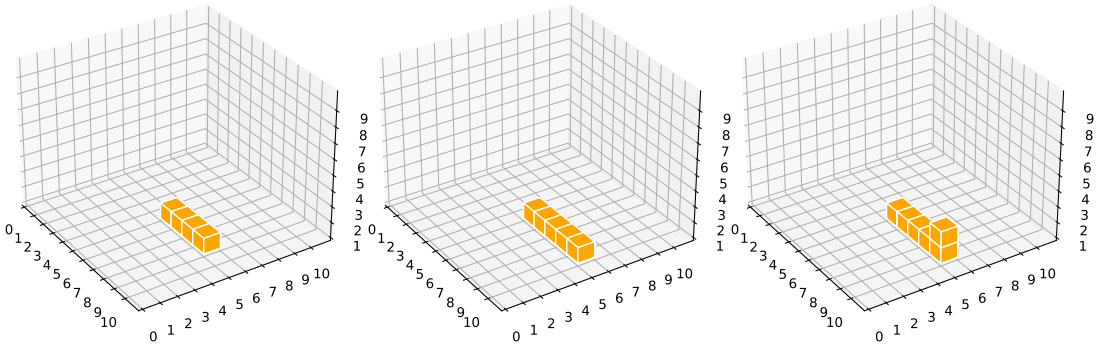

Figure 7: Building a plane by row (zigzag)