Journal of Research and Applications in Mechanical Engineering
ISSN: 2229-2152 (Print); 2697-424x (Online)
(2025) Vol. 13, No. 3, Paper No. JRAME 25-13-038
[DOI: 10.14456/ jrame.2025.38]

Research Article

Derivation of Positional Relation Matrix based on 3D Assembly CAD data: Disassembly Process Inference for Mechanical Products

K. Yamada*
K. Takahashi
K. Hanahara
Faculty of Science and Engineering, Iwate University, Morioka, 0208551, Japan

Received 19 October 2024
Revised 18 December 2024
Accepted 24 December 2024

Abstract:
During machine maintenance, the disassembly task is more challenging than the assembly task because the disassembling process involves more patterns and possibilities than the assembling process, depending on the reasons or objectives of disassembling. In addition, an incorrect disassembly may cause machine damage or human operator injuries. Lists and instruction manuals of parts have been used for machine disassembly; however, these manuals cannot include all parts in general, and the disassembly of some parts is difficult for unskilled operators. Therefore, the disassembly process inference is crucial. In order to infer the disassembly process, a positional relation matrix consisting of the relative positional relationship between any two components is required to clarify the disassembly process. Currently, the matrix is derived manually; however, an automatic generation approach is required to reduce the computation effort. In this study, a method was developed to automatically generate a positional relation matrix based on the geometric information of machine part models, which can be obtained from three-dimensional assembly data of a computer-aided design. A case study was conducted using the proposed method, and it was confirmed that the method could generate positional relation matrix for disassembly.

Keywords: *Disassembling process, Positional Relations Matrix, Separating-Axis Theorem, Maintenance Support, Collision Detection*

1. Introduction

In disassembly tasks associated with the maintenance of complex machinery, the disassembly sequence of all components is rarely provided explicitly. Although manuals are often available, the simplest disassembly sequence is typically the reverse of the assembly process used during production. However, fully disassembling machinery by retracing the assembly process whenever a component fails or specific maintenance is required is inefficient. Therefore, operators must determine a safe and efficient disassembly sequence according to the failure location and maintenance requirements. The more complex the machinery, the more challenging the task, which requires highly skilled operators. Frizziero et al. (1) propose a system that visualizes the generated disassembly sequence in augmented reality (AR) to support inexperienced operators in disassembly tasks.

A method for determining a safe and efficient disassembly sequence using a “positional relations matrix” has been developed (2). In this method, the positional relations matrix represents the spatial relationships between the components that constitute the machinery and is used to determine the extraction direction of the components (3)(4). In previous studies, this matrix was manually derived from the three-dimensional (3D) assembly data of the computer-aided design (CAD) of the product.

* Corresponding author: K. Yamada
E-mail address: yamadak@iwate-u.ac.jp

However, for machinery with many components, in which disassembly sequence inference is required, manual derivation is labor-intensive and challenging. Therefore, a study has been conducted to automatically generate a positional relations matrix from the 3D data of mechanical components using CAD interference recognition (4). Additionally, studies have been conducted to derive the connections between components using the API functions of 3D CAD software (5). Comparisons and analyses of differences among multiple CAD model files are also being carried out (6). However, a limitation exists in that the same API cannot be utilized across different CAD software, resulting in a lack of versatility. Moreover, interference detection demands substantial computational resources.

In order to more versatilely and automatically describe positional relationships of components, independent of software constraints, we attempt to use the geometric data of model files. The aim of this study was to establish an alternative method for automatically generating a positional relations matrix from the geometric information of the 3D data of mechanical components.

2. Positional relations matrix

The positional relations matrix expresses the distances between all pairs of components in a finished product state in six directions corresponding to the ±X, ±Y, and ±Z axes [3][4]. In this study, instead of focusing on distance, we derived a positional matrix by considering the state of contact between two components. For any pair of components, one part is designated as a fixed component, and the other as a moving component. Their relationship can then be expressed in each of the ±X, ±Y, and ±Z directions by categorizing it into two states: "the moving component collides when moved in that direction" or "the moving component does not collide when moved in that direction." When considering the collision relationship between two components, the states of the other components are ignored. A positional relations matrix was constructed by describing the collision state in each direction for every combination of components.

The positional relations matrix derived in this manner is represented as M [Fix(A), Tfr(A)], where A represents the entire set of components P that constitute the product. In this study, the positional relations matrix represents a state in which two components collide during movement as "1," and a state in which no collision occurs as "0." Thus, if "0" is indicated in the matrix, it signifies that no collision occurs between the two components in that direction, and the component can be extracted. Table 1 presents an example of a structure comprising five components, as shown in Figure 1. For example, the first row of Table 1 indicates the collision state between components 1 and 2 in six directions when component 1 is fixed and component 2 is moving. In the X+, X−, Y+, and Z+ directions, component 2 can move without colliding with component 1; hence, "0" is recorded. However, the two components collide during movement in the Y− and Z− directions; thus, "1" is recorded. Using this type of positional relations matrix, we propose a method for determining possible extraction directions and determining the disassembly sequence [1].

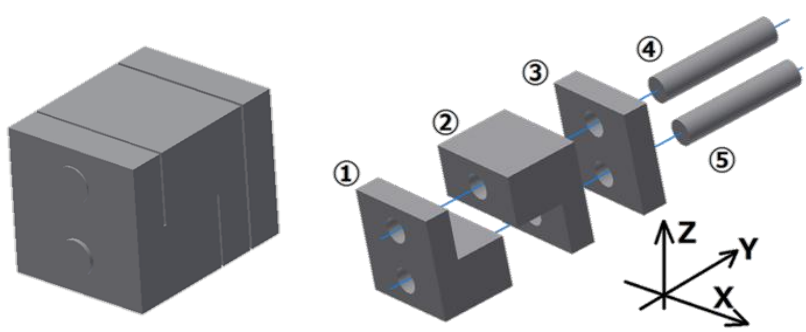

Fig. 1. Schematic of structure comprising five components.

Table 1: Positional relations matrix of Fig. 1

Fixing Part	Transferring Part	X+	X−	Y+	Y−	Z+	Z−
1	2	0	0	0	1	0	1
1	3	0	0	0	1	0	0
1	4	1	1	0	0	1	1
1	5	1	1	0	0	1	1
…	...						

3. Positional relations matrix

The assembly CAD data of the product converted into the OBJ format was used in this study to derive the positional relations matrix. The OBJ format divides the model into triangular polygons and contains information on the vertex coordinates of the triangles, normal vectors oriented outward from the model, and texture coordinates. The positional relations matrix is derived based on the vertex coordinates and normal vectors of the polygons in the OBJ file.

Any polygon within the moving or fixed component is considered. If a polygon in the moving component collides with a polygon in the fixed component when the moving polygon moves in a specific direction, the moving component collides with the fixed component when moving in that direction. A positional relations matrix can be derived by performing collision detection for all combinations of polygons in the fixed and moving components. Figure 2 illustrates the flow of the collision detection process between polygons.

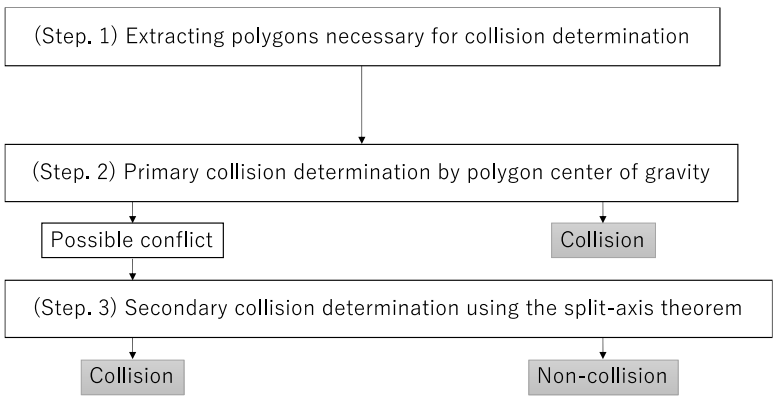

Fig. 2. Flowchart of the process

3.1 Extracting polygons for collision detection

For each combination of fixed and moving components, collision detection between polygons is performed to determine whether the fixed component obstructs the movement of the moving component.

It is unnecessary to consider all possible combinations of polygons between the fixed and moving components, and collision detection can be sufficiently performed using a limited set of polygon combinations. Consider the case in which fixed component F and moving component T have a positional relationship (Figure 3). In this case, the collision detection when component T moves in the Z+ direction is analyzed. Let S_F and S_T represent the surfaces of F and T, respectively, where the dot product of the unit vector in the disassembly direction d is (0, 0, 1), and the outward-facing normal vector of the surface is positive. Whether T collides with F when moving in the Z+ direction can be determined by considering whether S_T collides with S_F when moving in the Z+ direction. Therefore, primary collision detection between polygons, as described in Section 3.2, is performed only when the dot products of the normal vectors of the polygons in the fixed and moving components with ***d*** are positive.

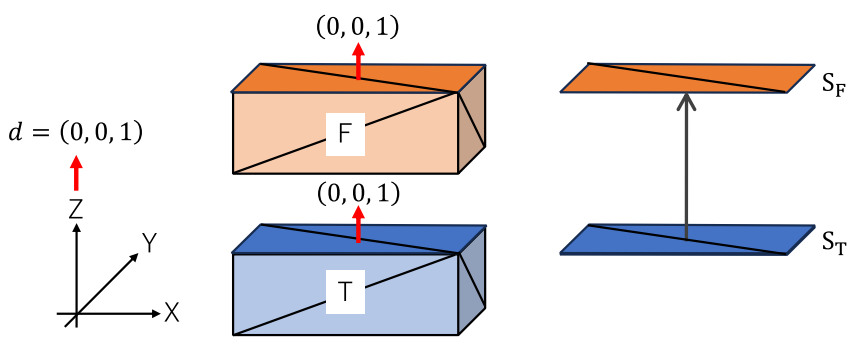

Fig. 3. Positional relationship between F and T

3.2 Primary collision detection based on the center of gravity of polygons

Primary collision detection is performed to determine whether the moving component may collide with or avoid the fixed component by focusing on the centroid positions of the two polygons.

Let *f* and *t* represent the polygons of fixed component F and moving component T, respectively, for which the dot product with the unit direction vector ***d*** in the disassembly direction is positive. If the positional relationship between *f* and *t* is such that the z-coordinate of the centroid of *f* is less than that of *t*, as illustrated in Figure 4, then moving *t* in the Z+ direction will not cause a collision with *f*. In contrast, if the positional relationship is such that the z-coordinate of the centroid of *f* exceeds or is equal to that of *t*, as shown in Figure 5, then a collision might occur when *t* moves in the Z+ direction. This implies a potential collision between moving component T and fixed component F when T moves in the Z+ direction, necessitating secondary collision detection, as described in Section 3.3.

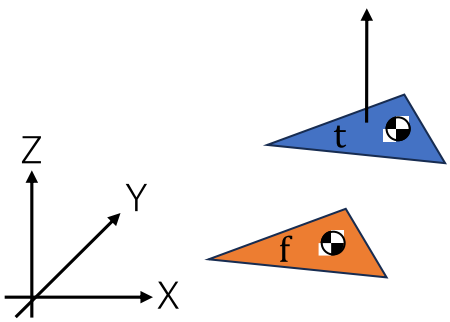

Fig. 4. *f* and *t* do not collide

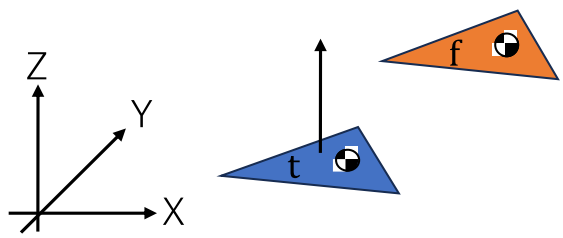

Fig. 5. *f* and *t* might collide

3.3 Secondary collision detection using split-axis theorem

For polygons *f* and *t* that are identified as potentially colliding during the primary collision detection, a secondary collision detection is conducted. Let *f′* and *t′* represent the projections of *f* and *t*, respectively, onto a plane P_z perpendicular to the disassembly direction, Z+. The intersection and containment between triangles *f′* and *t′* are then evaluated. As shown in Figure 6, if *f′* and *t′* intersect or one contains the other, moving *t* in the Z+ direction will lead to a collision with *f*. In contrast, as illustrated in Figure 7, *t*' can be moved in the Z+ direction without colliding with *f* if *f′* and *t′* do not intersect or contain each other. The separating-axis theorem is employed to determine the intersection and containment of the triangles.

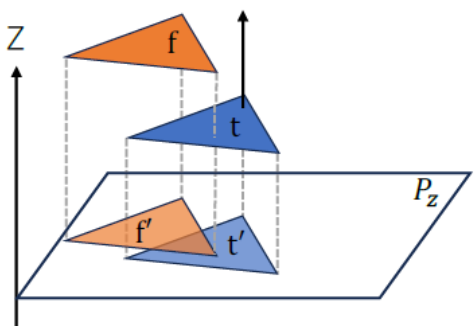

Fig. 6. *f*' and *t*' intersect and entail

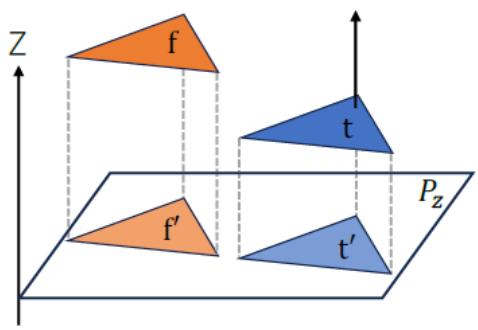

Fig.7. *f*' and *t*' do not intersect or entail

The separating-axis theorem states that, if two triangles do not intersect, a line that separates them can be drawn. The edges of the triangles function as candidates for separating the axes. One edge of the triangle is selected to determine the existence of a separating axis (Figure 8), and a line perpendicular to it is considered. Triangles A and B are then projected onto this line, resulting in line segments *a*' and *b*'.

If line segments a' and b' overlap, then one of the following conditions is satisfied.

- $b \leq a \cap a \leq b'$ (a lies between b and b′)
- $b \leq a' \cap a' \leq b'$ (a′ lies between b and b′)
- $a \leq b \cap b \leq a'$ (b lies between a and a′)
- $a \leq b' \cap b' \leq a'$ (b′ lies between a and a′)

Thus, no separating axis exists between triangles A and B. In this case, the same test is performed using another edge. If none of the six possible tests identify a separating axis, then triangles A and B intersect, or one contains the other. In contrast, if a separating axis is identified, triangles A and B neither intersect nor contain one another.

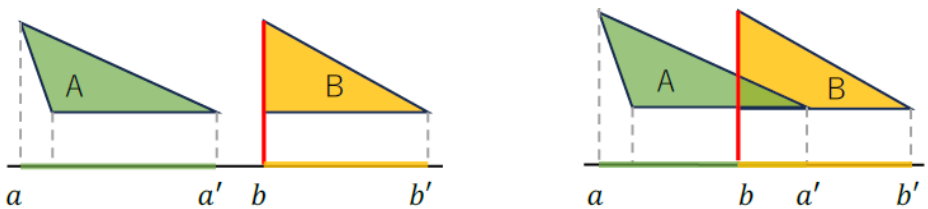

Fig. 8. Existence conditions of a separating axis

In this study, we focused on whether these two components collide with each other. Therefore, in cases where triangles A and B are merely touching each other (Figure 9), "the triangles do not collide," indicating that a separating axis exists. Consequently, no separating axis exists between triangles A and B if any of the following conditions are satisfied.

- $b \leq a \cap a < b'$
- $b < a' \cap a' \leq b'$
- $a \leq b \cap b < a'$
- $a < b' \cap b' \leq a'$

If no separating axis is identified between f' and t', then f' and t' intersect, or one contains the other. This suggests that moving t in the Z+ direction will cause a collision with f. In other words, "the moving component, T, will collide with the fixed component, F, when moved in the Z+ direction." If a separating axis exists, f' and t' do not intersect or contain each other, and moving t in the Z+ direction will not lead to a collision with f.

This process is applied to all combinations of polygons from the fixed and moving components in the six directions (±X, ±Y, ±Z) to derive the positional relations matrix between fixed component ***F*** and moving component ***T***.

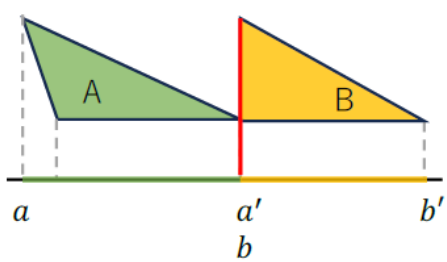

Fig. 9. Cases in which a separating axis exists

3.4 Incorrect detection

In this system, if a part contains long and narrow polygons, cases exist in which the collision detection is incorrect. Consider a collision detection in the Z+ direction, with a parallelepiped part being a fixed part, F, and a rectangular parallelepiped part being a moving part, T (Figure 10). When an arbitrary polygon in F is f and an arbitrary polygon

in T is *t*, the Z– coordinate of the center of gravity of *f* in the primary collision determination exceeds that of *t*. Next, second-order collision determination is performed using the split-axis theorem for *f*', *t*', which is the projection of *f*, *t* onto P_Z. Because the triangles intersect and include each other for *f*' and *t*', no dividing axis exists between *f*' and *t*'; that is, if moving part T moves in the Z+ direction, it will collide with fixed part F. However, in reality, moving part T does not collide with fixed part F even if it moves in the Z+ direction. There were cases in which incorrect judgments like these were made.

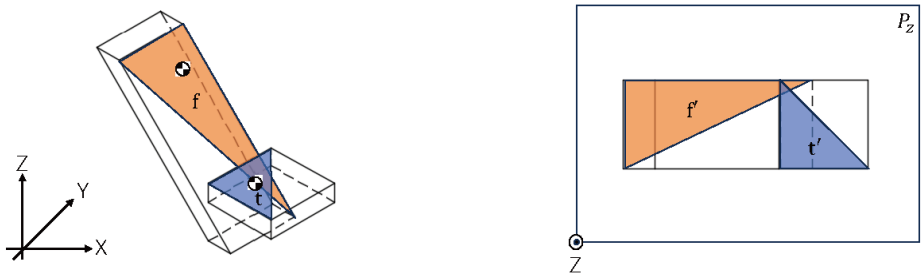

Fig. 10. Cases of misjudgement

4. System for deriving the positional relations matrix

The following describes the process of automatically generating the positional relations matrix from the 3D assembly CAD data of the mechanical component output in the OBJ format. In this system, the state in which two components collide owing to movement is represented by "1," and the state where they do not collide is represented by "0." The positional relations matrix between fixed component F and moving component T is denoted $\boldsymbol{M}_T^F$ and expressed as $\boldsymbol{M}_T^F = [M_{T(0)}^F, M_{T(1)}^F, \ldots, M_{T(5)}^F]$. Here, $M_{T(0)}^F$, $M_{T(1)}^F$, …, $M_{T(5)}^F$ represent the results of collision detection for T against F in the X+ direction, X− direction, and so on, up to the Z− direction.

Step 1: Determine fixed component F and moving component T.

Step 2: Define the positional relations matrix $\boldsymbol{M}_T^F$, with the initial values set to $\boldsymbol{M}_{\mathrm{T}}^F = [0,\ 0,\ 0,\ 0,\ 0,\ 0]$ (indicating no collision with F regardless of the movement direction of T.

Step 3: Select one polygon from fixed component F, which is represented as F.

Step 4: Select one polygon from moving component T, which is represented as t.

Step 5: Perform collision detection for F and t in all six directions (i = 0–5).

Step 5.1: When i = 0, perform collision detection in the X+ direction.

Step 5.1.1: If the dot product of the normal vectors F and t with the unit direction vector d = (1,0,0) is less than or equal to zero, skip detection and return to Step 5.

Step 5.1.2: Determine the centroid coordinates of F and t. If the x-coordinate of the centroid of F is less than that of t, no potential collision occurs; therefore, return to Step 5.

Step 5.1.3: Project F and t onto plane P_x perpendicular to the X+ direction, and perform the separating-axis test. If no separating axis is identified, no potential collision will occur; therefore, return to Step 5. If a separating axis is found, record the collision as $\boldsymbol{M}_{T(i)}^F = 1$, and return to Step 5.

Similarly, perform collision detection for i = 1 in the X− direction, i = 2 in the Y+ direction, i = 3 in the Y− direction, i = 4 in the Z+ direction, and i = 5 in the Z− direction.

Step 6: Repeat Steps 3–5 for all combinations of polygons from fixed component F and moving component T.

Step 7: The positional relations matrix $\boldsymbol{M}_T^F$ is completed.

Step 8: Repeat Steps 1–7 for all combinations of fixed and moving components.

5. Case study

For the structure comprising five components 1–5 (Figure 1), the proposed system was used to automatically derive the positional relations matrix. Modeling was performed using Fusion 360 (7), and the system was constructed using Unity (8). The derived positional relations matrices are listed in Table 2. This confirms that the positional relations matrix derived using the system is consistent with the matrix manually developed while reviewing the CAD assembly data. Hence, the proposed system can correctly generate the matrix.

Table 2: Positional relations matrix of Fig.1 derived using the proposed system.

Fixing Part	Transferring Part	X+	X−	Y+	Y−	Z+	Z−
1	2	0	0	0	1	0	1
1	3	0	0	0	1	0	0
1	4	1	1	0	0	1	1
1	5	1	1	0	0	1	1
2	1	0	0	1	0	1	0
2	3	0	0	0	1	0	0
2	4	1	1	0	0	1	1
2	5	1	1	0	0	1	1
3	1	0	0	1	0	0	0
3	2	0	0	1	0	0	0
3	4	1	1	0	0	1	1
3	5	1	1	0	0	1	1
4	1	1	1	0	0	1	1
4	2	1	1	0	0	1	1
4	3	1	1	0	0	1	1
4	5	0	0	0	0	1	0
5	1	1	1	0	0	1	1
5	2	1	1	0	0	1	1
5	3	1	1	0	0	1	1
5	4	0	0	0	0	0	1

6. Conclusion

In this study, a method was developed to automatically generate a positional relations matrix from the 3D assembly CAD data of mechanical components. The results of the case study demonstrate the effectiveness of the proposed system for deriving positional relations matrixes. This confirms that it is possible to automatically derive positional relations matrixes from CAD assembly data of mechanical components created with various 3D CAD software. By automating the derivation of positional relations matrixes, which was previously described manually, we can significantly reduce the workload. Furthermore, by combining the proposed system for automatic derivation of positional relations matrixess with existing disassembly process inference, it becomes possible to infer the disassembly process of various mechanical products. This allows non-experts to operate disassembly work smoothly while being guided through the optimal disassembly process.

However, errors may occur when the parts are divided into triangular polygons, particularly if the polygons are elongated (mentioned in section 3.4). To address this issue, we plan to incorporate collision detection using depth information of the positional relationships of components. Determining whether one triangular polygon is contained within another triangular polygon when viewed from a specific line of sight is planned. If they are contained, collision or non-collision is determined based on which one appears in front. If they are not contained, it is determined that they do not overlap. If they do not contain each other from any direction, it is determined that there is no collision. We are currently engaged in implementing this approach into our system. In addition, the system does not currently handle cases in which parts must be extracted using twisting motions. Regarding these issues, it is anticipated that by using AI to identify the threaded sections from the triangular polygon mesh data, it will be possible to describe the positional relations matrix.

Acknowledgments

This work was supported by JSPS KAKENHI (Grant Number: JP23K11719).

References

[1] Frizziero L, Liverani A, Caligiana G, Donnici G, Chinaglia L. Design for Disassembly (DfD) and Augmented Reality (AR): Case Study Applied to Gearbox. Machines. 2019;7(2):29.
[2] Yamada K, Monma T, Hanahara K. Disassembling process inference using positional relations matrix for complicated machines. Mechanical Engineering Research. 2023;11(1):1-8.
[3] Shinoda S, Shimozawa K, Niwa A, Kawase T, Matsumoto T, Mizumachi T. A proposal for prototype-free production preparation processes utilizing 3DCG animations. Industrial Engineering and Management Systems. 2009;8(2):109-120.
[4] Hashimoto Y, Ichizaki O, Shinoda S. A fundamental study on an exhaustive assembly sequences generation method using the positional relations matrix. Journal of Japan Industrial Management Association. 2016;67(2):83-91.
[5] Hasan BA, Wikander J, Onori M. Assembly Design Semantic Recognition Using SolidWorks-API. International Journal of Mechanical Engineering and Robotics Research. 2016;5(4):280-287.
[6] Zou L, Guo D, Gao H. A method to analyze the difference of 3-D CAD model files based on feature extraction. Journal Mechanical Science Technology. 2011;25(4):971-976.
[7] AUTODESK. Fusion360 [Internet]. [cited 2024 Aug 24]. Available from: https://www.autodesk.co.jp/products/fusion-360
[8] Unity Technologies. Unity [Internet]. [cited 2024 Aug 24]. Available from: https://unity.com