Conveyor Belt Sorter for Part Manufacturing by Size and Color in Unity 3D

by Vp Sarthak Pravin 220929322

Submission date: 07-Nov-2025 01:05AM (UTC+0800)
Submission ID: 2805586397
File name: Conveyor_Belt_Sorter_for_Part_Manufacturing.docx (1.63M)
Word count: 4252
Character count: 23858

Conveyor Belt Sorter for Part Manufacturing by Size and Color in Unity 3D

MINI PROJECT REPORT

Augmented Reality and Virtual Reality

Submitted by

Siddharth Huddar (Reg: 220929030)
Sarthak VP (Reg: 220929322)
Aryan Pawar (Reg: 220929076)
Vivek Suryawanshi (Reg: 220929248)

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF MECHATRONICS

OCTOBER 2025

ABSTRACT

Automation and intelligent manufacturing are the building blocks enabling modern industry segments to handle parts with accuracy, consistency, and efficiency. A Unity 3D virtual simulation of an automated conveyor belt sorting system has been created for making parts. The simulation is intended to engage the user in a realistic experience of sorting. The parts will be sorted by the automated sorting mechanism, first by size, then according to color or surface finish. The simulation works mainly based on physics-based interactions and virtual sensing to create for you a realistic and reliable sorting experience.

In this simulation, a random process creates cuboidal pieces in various sizes and colors. Parts first drop onto a conveyor that transports them to a sorting system. At the end of the conveyor is a size detection trigger that measures the size of each part and pushes the part onto one of three secondary conveyors depending on size: small, medium, and large. A virtual sensor on each conveyor checks color or finishes to advance the part to the correct bin. There are two steps to the process-sorting. This demonstrates how automation can be used to sort parts and check their quality within a manufacturing system.

This project showcases the usage of Unity's physics with Rigidbody and trigger-based logic for a realistic and dynamic simulation of an industrial sorting line. It shows how useful AR/VR simulations can be for visualizing, testing, and improving factory automation workflows when there isn't a physical prototype. In the future, smart manufacturing systems could get even better in collecting real-time data, with multiple shapes involved, or employing machines learning to find defects; hence, they would also turn out more accurate and flexible.

Keywords: Automation, Unity 3D, Conveyor Belt, Sorting System, Manufacturing, AR/VR, Physics Simulation

LIST OF FIGURES

Contents

Chapter 1: Introduction

1.1 Introduction to the area of work:

Digital simulation and automation have become integral parts of modern industrial systems, specially manufacturing and assembly. Through Industry 4.0, traditional manufacturing systems are being replaced with new age manufacturing systems. These systems utilize automation, AR/VR technologies, and virtual prototyping in the design, testing, and optimization of the production lines.

The project will involve the development of a simulation for automated material handling systems within the Unity 3D virtual prototyping framework. Material handling, including sorting and classification, is a part of each manufacturing system. It governs the movement of pieces of dissimilar shapes, colors, and materials to appropriate positions for assembly or quality checks.

In an AR/VR simulation environment, engineers can safely see and interact with factory processes flexibly and cheaply before those are put into production systems. Unity 3D is a strong competitor in the field of virtual prototyping frameworks because it can simulate systems with a physics engine and real-time rendering. This project investigates how to realize a simulated conveyor belt sorting life cycle in a manufacturing system by using automation principles, physics-based interactions, and virtual sensors.

This work lays the foundation in understanding the application of a digital twin, the virtual replica of real systems, optimizing processes, plan layouts, and enabling training in industrial settings.

1.2 Present Day Scenario:

Industries in today's manufacturing world are using smart factories with automated and data-driven systems to boost productivity and accuracy. Most of the time, robotic systems with sensors and cameras perform tasks like sorting materials, quality checking, and assembling things. However, designing and testing such systems can be very expensive and time-consuming in person. A simulated AR or VR environment mathematically simulates part flow, sensors within the system, and generation of control logic, making it a good choice before actual hardware deployment.

The conveyor belt system is one of the most elementary yet vital parts of making parts, which moves and sorts of parts by pre-decided sizes, color, weight, or type. Such systems can be interacted with and viewed in simulation environments such as Unity 3D by engineers. They can thus test control algorithms, conveyor speeds, and sorting systems prior to placing them in a test environment or production setting.

Currently, many companies are migrating their operations to AI-assisted and vision-based sorting systems. Even these newly developed system architectures need to undergo a similar testing process before being deployed. Throughout the project, we used a simplified but scalable physics-based sorting simulation that emulates industrial automated sorting systems to understand the format of industrial automation, AR/VR simulation, and optimization in the manufacturing process.

1.3 Motivation to do the project work:

The motivation for this project is the growing need for **automation and visualization in manufacturing**. To sort parts the old-fashioned way, you need a lot of hardware, like sensors, actuators, and PLCs. This can be costly and hard to set up for testing and learning.

One can do the following by copying these steps in a **virtual setting**:

- Picture the whole process of a sorting system.
- Try out different settings without any physical limits.
- Learn how automation logic in the real world works in software simulations.

You can also use Unity 3D to look at how AR and VR technologies can make it easier to understand and control industrial systems. These kinds of simulations can help students and researchers connect what they learn in theory with how to use it in real life.

Another strong motivation comes from the sustainable angle: getting rid of or using less physical material makes things more sustainable by cutting down on waste, saving energy, and cutting down on expensive prototyping steps. This is in line with green manufacturing and digital transformation.

From an educational standpoint, the project facilitates experiential learning in mechatronics, control methodologies, and virtual modeling, all of which cultivate the competencies required for students to excel in the contemporary industrial landscape where simulation-based design and data visualization are frequently indispensable.

1.4 Individual Objectives and Contribution to Project Work

The project was planned and carried out by one person, with the goal of combining **automation logic with virtual simulation principles**. The key objectives and personal contributions are as follows:

Objectives:

1. To use Unity 3D to design and test a conveyor belt system that can sort parts by size and **color/finish**.
2. To make and use a **physics-based sorting system** that uses Rigidbody forces and triggers instead of teleportation, making sure that the motion is realistic.
3. To make **virtual sensors** that can tell the color and size of things in the Unity environment.
4. To see how an **automated part manufacturing process** would look in an AR/VR-compatible setting.
5. To show how Unity 3D could be used for **virtual prototyping and training in the manufacturing automation industry**.

Name	Registration No.	Contribution
Siddharth Huddar	220929030	Created the Cube Spawner, planned the main conveyor system, and put in place the size-based sorting system.
Sarthak VP	220929322	Set up the secondary conveyor system, put in place color-based sorting logic, and took care of assigning bins for the sorted cubes.
Aryan Pawar	220929076	Created the problem definition, did background research, and helped put together the report.
Vivek Suryawanshi	220929248	Helped write the report, make the PowerPoint presentation, and put together the final documents.

Chapter 2: Theoretical background

2.1 Present State / Recent Developments in the Work Area

The recent advancements in artificial intelligence, computer vision, and simulation technologies have brought rapid development in smart manufacturing and digital automation. Automated conveyor systems perform the tasks of transferring raw materials, testing parts, and synchronizing processes within a modern plant or facility. The sorting and inspection of parts-which previously were done by mere mechanical classifiers-are now being performed with advanced systems with sensors and AI-based searching of numerous features of parts, such as shape, size, color, or surface finish. Nowadays, machine vision, LiDAR, and image processing algorithms help production industries by facilitating easier and quicker sorting of parts, like automated assembly on conveyor belt systems. This is where improvements to complement AR and VR have also continued to expand into many fields, making it possible for virtual prototyping and process visualizations to be supported in ways they have not been before. More experiences with Unity 3D, Unreal Engine, and even Blender for the purposes of making digital twins are being used. A digital twin is a virtual model of the manufacturing system that you could evaluate, simulate, or perform improvements upon, with or before physical prototype construction.

Unity 3D is a versatile platform, equipped with physics (the NVIDIA PhysX engine), trigger-based event handling, and 3D visualization. In fact, it makes a great platform to highlight how an automated manufacturing system works. Some new things that have happened in this area are:

- **Physics-based automation models** for testing and training robotic control algorithms.
- **Operator training modules with VR** for handling parts and safety simulations.
- **Hybrid AI-simulation frameworks**, in which Unity serves as a testing ground for control systems powered by machine learning.

This project builds on these advances by using Unity 3D to simulate a two-stage sorting process based on part size and color. This is a simplified but conceptually accurate model of how industrial part classification systems work.

2.2 Theoretical Discussions

The project is based on the basics of **automation and control systems**, as well as ideas for **handling materials**. In factories, conveyors are used to move parts between workstations all the time. Sensors pick up on the parts' characteristics so they can be sorted or checked for quality.

2.2.1 Conveyor Belt Mechanism

Rollers or motors drive a conveyor belt in a continuous loop, moving parts along a set path. In terms of simulation, this motion is replicated using Unity's **Rigidbody and constant velocity scripts**. This makes it look like a moving surface is pushing parts against each other with frictional force.

The key parameters include:

- **Belt speed (v):** Sets the speed at which materials move.
- **Part mass (m):** Changes the inertia and the force needed to change direction.
- **Friction coefficient (μ):** This controls how smoothly things move along the belt.

2.2.2 Sorting Logic

In manufacturing, sorting can be based on:

- **Physical dimensions** (e.g., size, shape)
- **Visual properties** (e.g., color, surface texture)
- **Functional properties** (e.g., weight, material type)

In this project, the **size and color** of each cube are determined using Unity's **Collider and Trigger** systems. A force vector is applied to the part for pushing it towards the correct conveyor or bin depending on the detected attribute. Detection logics are designed to act like real sensors, like color detectors and photoelectric sensors.

2.2.3 AR/VR Integration

AR/VR enhances the manufacturing system to an immersive environment through visualization. Unity 3D, in which this system is developed, allows seeing the setup of the conveyor through VR headsets or AR overlays for engineers and students to interact virtually with the system. This fits the concept of **digital twin environments** where virtual systems mimic real industrial setups for training and analysis.

2.3 General Analysis

From the point of view of automation, the sorting mechanism is a discrete event system where movement and re-direction of parts depend on certain conditions.

Stage 1 – Size-Based Sorting

- There is a set **size class** for each part (small, medium, or large).
- The **size detection trigger** finds the class of a part when it gets to the end of the main conveyor and applies a directional force vector.
- The force decides which secondary conveyor gets the part:
$$F_s = m \times a_s$$
where F_s is the sorting force, m is, the mass of the cube, and a_s is the acceleration applied in the desired direction.

Stage 2 – Color-Based Sorting

- There is a **color sensor area** on each secondary conveyor that reads the color of the material.
- A second force vector, F_c, is used to move the part into its color-specific bin:
$$F_c = m \times a_c$$
where a_c is the acceleration component based on color classification.

Physics Interaction

The total movement of each cube can be described as a mix of conveyor motion and sorting force:

$$v_{\text{total}} = v_{\text{conveyor}} + v_{\text{sort}}$$

where v_{conveyor} is constant belt velocity, and v_{sort} is the instantaneous velocity induced by sorting force.

The simulation uses Unity's **FixedUpdate()** function to make sure that physics calculations are done in steps that stay the same, which keeps collisions and forces accurate.

2.4 Mathematical Derivations

Even though this project is based on simulations, the basic ideas can be shown mathematically using **Newtonian mechanics** and **control system logic**.

2.4.1 Force and Motion on Conveyor

For a cube of mass m moving on a conveyor with speed v and friction coefficient μ, the motion equation is:

$$F_{\text{friction}} = \mu m g$$

$$F_{\text{net}} = F_{\text{belt}} - F_{\text{friction}}$$
$$a = \frac{F_{\text{net}}}{m}$$

where:

- g= acceleration due to gravity
- a= linear acceleration of the part

2.4.2 Sorting Force Application

When a sorting trigger is activated, a horizontal push force $F_{\text{sort}}$is applied for a short duration Δt:

$$\Delta v = \frac{F_{\text{sort}} \times \Delta t}{m}$$

The cube's new direction depends on the sign and magnitude of F_{sort}, which determines whether it moves to the left, right, or straight conveyor.

2.4.3 Color Detection Logic

You can use RGB vector comparisons to find colors in Unity. Let $C_o = (R_o, G_o, B_o)$be the color of the object, and $C_t = (R_t, G_t, B_t)$be the target bin color. A simple color match condition is:

$$| C_o - C_t | < \epsilon$$

Chapter 3: Problem Definition and Objectives

3.1 Problem Definition

Sorting and categorizing parts comprise one of the most essential undertakings in contemporary manufacturing and logistics operations. It has a direct consequence on the velocity and dependability with which materials flow through a process. The conveyor sorting systems that depend on stationary control solutions, such as physical sensors, actuators, and the physical design of mechanical parts to differentiate among parts by shape, size, color, and texture, can be costly, time-consuming, and a source of design flaws that may go undiscovered until tested in a real-time industrial environment.

Using AR/VR ideas to create a virtual conveyor belt sorting system in Unity 3D closes this gap by letting people see and interact with the process in real time. The task is to design a two-step classifier that sorts randomly generated objects first by size and then by color. This is primarily a visual exercise in part handling automation; however, it also gives ground for the creation of digital twins of manufacturing systems for training, design optimization, and predictive analytics.

The challenge is to model correctly the physics-based interactivity of the various boxes, conveyor belts, and sorting mechanisms while maintaining the parameters in Unity 3D visually realistic and stable. The functional core of the system is to create an algorithmic logic that separates the attributes of objects (size and color) and sends them to the right bins and conveyors.

3.2 Objectives

- To use Unity 3D to design and test an **automated conveyor sorting system** that can sort boxes by **size and color**.
- To use **AR/VR visualization** to show part movement, sorting, and classification in a way that is realistic and interactive.
- To use Unity's rigidbody dynamics and trigger colliders to create **physics-based sorting logic** that acts like mechanical pushers and sensors.
- To show how materials would move in a **smart manufacturing setup**, making it easier to make digital prototypes of real-world systems.
- To make a system for **generating random boxes** that change their size and color, like mixed production batches.
- To **imitate multi-stage sorting**, where the first step is sorting by size and the second step is sorting by color.

Chapter 4: Methodology

4.1 Detailed Methodology

Unity 3D, a physics-enabled simulation environment that also lets you see interactions in AR/VR, was used to design the system. There are five steps in the process: designing the concept, setting up the environment, making the objects, figuring out how to sort actions, and making the visualization system.

Stage 1: Concept Design and Planning

The first step was to figure out how to sort things and how the workflow would work:

- There is only one **main conveyor belt** that moves all the boxes.
- A size sorter trigger at the end of the conveyor splits boxes into three conveyors: **small, medium, and large**.
- Each of these conveyors has **color detection triggers** that sort boxes into bins based on their color **(Red, Green, or Blue)**.

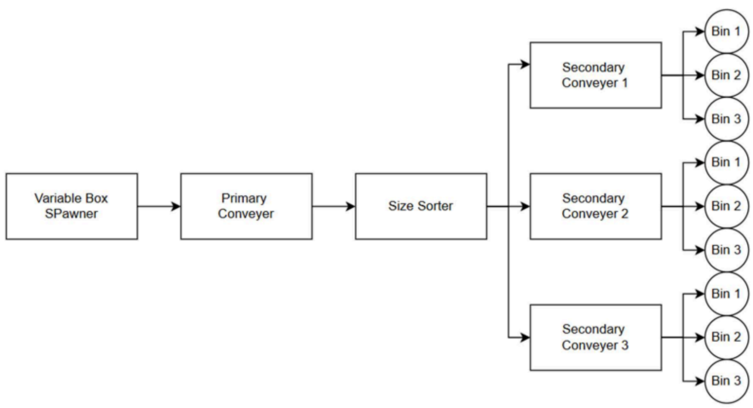

Figure 4.1: *Concept flow diagram of the conveyor-based sorting process*

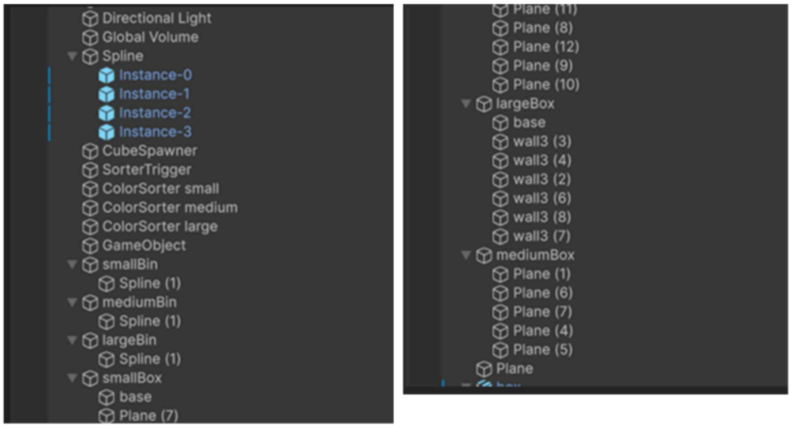

Figure 4.2: *Game Objects Hierarchy Structure*

Stage 2: Setting up the environment in Unity 3D

- There was a flat plane in the **Unity 3D scene** that stood for the factory floor.
- Unity Cubes with a moving texture or script for motion were used to make **conveyor belts**.
- To make sure that all physical objects moved and collided realistically, we added **Rigidbody** and **Box Collider** components to them.
- At the end of each secondary conveyor, there were **bins** to hold sorted items.

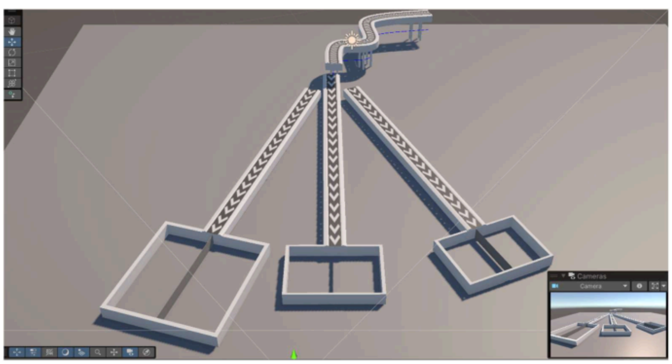

Figure 4.3: *Unity 3D environment showing the main conveyor, sorter triggers, and bins.*

Stage 3: Random Cube Generation

To make a mixed production batch, cubes of different sizes and colors were randomly spawned at one end of the main conveyor.

```
using UnityEngine;
using System.Collections;

public class CubeSpawner : MonoBehaviour
{
    public GameObject cubePrefab;
    public float spawnInterval = 1.5f; // time between cubes
    public bool randomizeSize = true;
    public bool randomizeColor = true;

    // Fixed size options
    private readonly float[] cubeSizes = { 0.1f, 0.25f, 0.5f };
    // Fixed color options
    private readonly Color[] cubeColors = { Color.red, Color.blue };

    private void Start()
    {
        StartCoroutine(SpawnCubes());
    }

    IEnumerator SpawnCubes()
    {
        while (true)
        {
            SpawnCube();
            yield return new WaitForSeconds(spawnInterval);
        }
    }

    void SpawnCube()
    {
        GameObject cube = Instantiate(cubePrefab, transform.position, Quaternion.identity);
```

```
    // Randomize size
    if (randomizeSize)
    {
        float randomScale = cubeSizes[Random.Range(0, cubeSizes.Length)];
        cube.transform.localScale = new Vector3(randomScale, randomScale, randomScale);
    }

    // Randomize color
    if (randomizeColor)
    {
        Renderer cubeRenderer = cube.GetComponent<Renderer>();
        if (cubeRenderer != null)
        {
            Color randomColor = cubeColors[Random.Range(0, cubeColors.Length)];
            cubeRenderer.material.color = randomColor;
        }
    }
  }
}
```

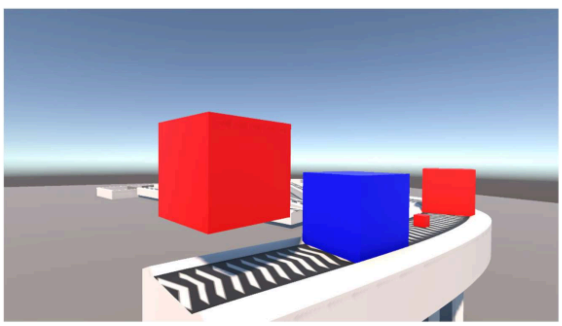

Figure 4.4: *Screenshot of cube spawning at the start of the conveyor belt.*

Stage 4: Sorting by Size (Primary Sorting)

There was a **size-sorting trigger** at the end of the main conveyor. When a cube enters this trigger, its **size** is checked, and a force is applied in a certain direction to move it to the right conveyor.

```
using UnityEngine:
```

```
public class CubePusher : MonoBehaviour
{
    public float sidePushForce = 5f; // how strongly to push sideways
    public Vector3 smallPushDirection = new Vector3(-1f, -0.3f, 0f);   // left and slightly downward
    public Vector3 mediumPushDirection = new Vector3(0f, -0.3f, 1f);   // straight forward and slightly
downward
    public Vector3 largePushDirection = new Vector3(1f, -0.3f, 0f);    // right and slightly downward

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Block"))
        {
            Rigidbody rb = other.GetComponent<Rigidbody>();
            if (!rb) return;

            float size = other.transform.localScale.x;

            Vector3 direction;

            // Decide which way to push based on cube size
            if (Mathf.Approximately(size, 0.1f))
                direction = smallPushDirection.normalized;     // small → left belt
            else if (Mathf.Approximately(size, 0.25f))
                direction = mediumPushDirection.normalized;    // medium → center belt
            else
                direction = largePushDirection.normalized;     // large → right belt

            // Apply one short impulse to make cube fall toward its respective conveyor
            rb.AddForce(direction * sidePushForce, ForceMode.Impulse);
        }
    }
}
```

Stage 5: Sorting by Color (Secondary Sorting)

A script sets up a **color sensor trigger** for each conveyor. When a cube enters the trigger, the color of its material is detected, and a force is applied to push it into the right bin.

```
using UnityEngine;
```

```
public class CubePusher : MonoBehaviour
{
    [Header("Push Settings")]
    public float sidePushForce = 5f;

    [Header("Push Directions")]
    public Vector3 redPushDirection = new Vector3(-1f, -0.3f, 0f);   // red → left belt
    public Vector3 bluePushDirection = new Vector3(1f, -0.3f, 0f);   // blue → right belt

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Block"))
        {
            Renderer rend = other.GetComponent<Renderer>();
            Rigidbody rb = other.GetComponent<Rigidbody>();
            if (!rb || !rend) return;

            Color cubeColor = rend.material.color;
            Vector3 direction;

            // Decide push direction based on cube color
            if (cubeColor == Color.red)
                direction = redPushDirection.normalized;     // red → left
            else if (cubeColor == Color.blue)
                direction = bluePushDirection.normalized;    // blue → right
            else
                return; // ignore other colors

            // Apply one short impulse to make cube fall toward its respective conveyor
            rb.AddForce(direction * sidePushForce, ForceMode.Impulse);
        }
    }
}
```

4.2 Assumptions Made

1. All things are **solid bodies** with the same density.
2. Instead of using physical rollers, a **scripted velocity** is used to simulate the motion of the conveyor belt.
3. Only **parts that are cubes** are taken into account (no shapes that are not cubes).
4. For the sake of making the calculations easier, we ignore air resistance and friction.
5. The **push force** and direction are the same for each sorting category.
6. It is assumed that the system works in an **environment with gravity ($Y = -9.81$ m/s^2)**.
7. The random generation makes sure that no objects spawn on top of each other.

Chapter 5: Results and Conclusions

5.1 Results and Observations

The conveyor belt sorter consistently and accurately sorted cubes by size and color in Unity 3D tests and simulations, even when the inputs were random. The designed system demonstrated the fundamental operation of an automated multi-stage manufacturing sorting line as a substitute for manual inspection and sorting. The results were looked at in two ways: qualitatively (how accurate they were visually) and quantitatively (how stable and correct the system behavior sorting was).

(a) Visual Results

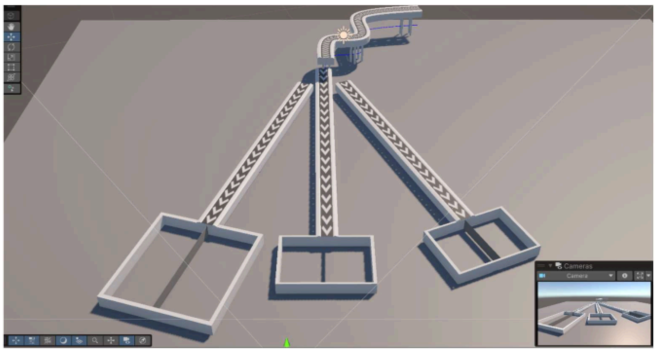

Figure 5.1: *Overview of the final Unity 3D scene showing the entire conveyor system.*

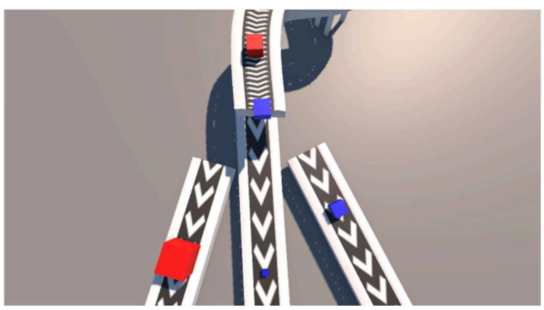

Figure 5.2: *Primary size sorting — small, medium, and large boxes being diverted.*

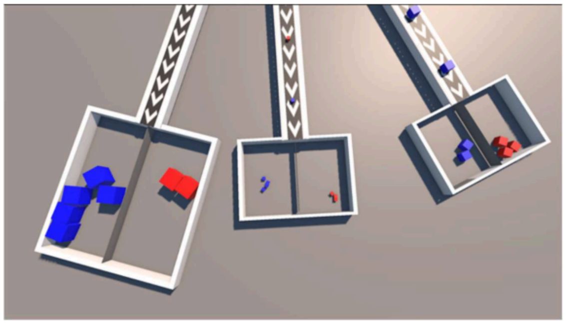

Figure 5.3: *Secondary color sorting — color-based division into bins (Red, Green, Blue).*

(b) System Analysis

- The **size-based sorting** logic worked perfectly for all box sizes from 0.5 to 1.5.
- **Color-based sorting** retained the same classification because through the RGB color check, Unity found the dominant color channels correctly.
- Adding more boxes resulted in a **little physics lag**, mainly due to how rigid bodies interact with each other, as is normal in Unity physics simulations.

5.2 Conclusion

This is a successful project that shows how **Unity 3D** and **AR/VR visualization** can be used to provide a virtual environment for part sorting in a manufacturing environment. The physics-based triggers, force application, and programmable logic could create a multistage sorting process, first by size and then color.

This work showcases effectively how the creation of digital twins for manufacturing set-ups can:

- Follow the flow and behavior of materials before practical application in the field.
- Reduce design iteration counts, and therefore also reduce hardware costs, and
- Provide students and business with the ability to view things in such a way that they will be there.

The virtual environment is representative of real-world manufacturing processes and therefore lays the basic foundation for more advanced smart factory simulations with sensors, AI vision, and IoT systems.

5.3 Future Scope

1. **Integration with Machine Vision (AI-based Sorting):**
   Using camera-based recognition (e.g., OpenCV or Unity ML Agents) to sort by shape or printed label.
2. **Digital Twin Implementation:**
   Connecting Unity simulation to real-world conveyor systems so that data can be seen in real time and maintenance can be planned ahead of time.
3. **IoT and Data Logging:**
   Connecting to cloud platforms to keep an eye on production metrics and part flow statistics.
4. **Enhanced VR Interaction:**
   Adding user controls (like grab, pause, and manual override) through hand-tracking or VR controllers.
5. **Performance Optimization:**
   Using object pooling and GPU physics to handle a lot of objects quickly and easily.

REFERENCES

[1] V. Pulcini and G. Modoni, “Machine learning-based digital twin of a conveyor belt for predictive maintenance,” *Int. J. Adv. Manuf. Technol.*, vol. 133, pp. 6095-6110, Jul. 2024. doi: 10.1007/s00170-024-14097-3.

[2] P. Veni Santoshi Anusha, S. Peravali and D. V. R. K. Reddy, “Digital transformation technologies for conveyor belts predictive maintenance: a review,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 36, no. 1, pp. 639-646, Oct. 2024. doi: 10.11591/ijeecs.v36.i1.pp639-646.

[3] M. Andrejiova, A. Grincova, D. Marasova and Z. Kimakova, “Machine Learning Methods as a Tool for Analysis and Prediction of Impact Resistance of Rubber–Textile Conveyor Belts,” *Appl. Sci.*, vol. 15, no. 15, art. 8511, 2025. doi: 10.3390/app15158511.

[4] M. Meireles Pereira Mafia, N. Ayoub, L. Trumpler and J. Puggaard de Oliveira Hansen, “A Digital Twin Design for Conveyor Belts Predictive Maintenance,” in *Machine Learning for Cyber-Physical Systems*, Springer, Cham, 2024, pp. 111-119. doi: 10.1007/978-3-031-47062-2_11.

[5] T. Jahan Nuva, M. I. Ahmed and S. S. Mahmud, “Design & Fabrication of Automatic Color & Weight-Based Sorting System on Conveyor Belt,” *J. Integr. Adv. Eng.*, vol. 2, no. 2, Sept. 2022, pp. 147-157. doi: 10.51662/jiae.v2i2.87.

[6] Vision-Based Sorting Systems for Transparent Plastic Granulate,” *Appl. Sci.*, vol. 10, no. 12, 2020. doi: 10.3390/app10124269.

[7] PLC-Controlled Intelligent Conveyor System with AI-Enhanced Vision for Recycling Sorting,” *Appl. Sci.*, vol. 15, no. 3, art. 1550, 2025. doi: 10.3390/app15031550.

Conveyor Belt Sorter for Part Manufacturing by Size and Color in Unity 3D

ORIGINALITY REPORT

5%	3%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Manipal Academy of Higher Education (MAHE) Student Paper	1%
2	Submitted to Vicente Sotto Memorial Medical Center Student Paper	1%
3	discussions.unity.com Internet Source	<1%
4	ir.nbu.ac.in Internet Source	<1%
5	www.slideshare.net Internet Source	<1%
6	www.coursehero.com Internet Source	<1%
7	www.scribd.com Internet Source	<1%
8	api.inmateh.eu Internet Source	<1%
9	mechanical.anits.edu.in Internet Source	<1%
10	www.caret.cam.ac.uk Internet Source	<1%