

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное учреждение высшего
образования**
**«Южно-Уральский государственный университет» (национальный исследовательский
университет)**
Политехнический институт
Кафедра «Электропривод, мехатроника и электромеханика»

Разработка системы технического зрения логистики склада

(наименование темы курсовой работы)

КУРСОВАЯ РАБОТА
по дисциплине «Компьютерное зрение»
ЮУрГУ – 15.03.06. 2023. КР
(код направления/специальности, год)

Нормоконтролер, доц.

_____/ И.Ю. Холодилин /
« ____ » _____ 2023 г.

Руководитель, доц.

_____/ И.Ю. Холодилин /
« ____ » _____ 2023 г.

Авторы

студенты группы П – 356

_____/ А.А. Белопухов /

_____/ Д.В. Федченко /

_____/ А.С. Усынин /

_____/ М.М. Рогов /

« ____ » _____ 2023 г.

**Работа защищена
с оценкой (цифрой, прописью)**

« ____ » _____ 2023 г.

Челябинск 2023

АННОТАЦИЯ

Белопухов А.А., Федченко Д.В., Усынин А.С., Рогов М.М. Разработка системы технического зрения логистики склада, – Челябинск: ЮУрГУ, П-356; 2023, 24 с., 12 ил., библиогр. список – 5 наим.

В ходе работы была смоделирована сцена в Unity, согласно теме курсовой работы, которая симулирует сортировку объектов на складе, созданы модели для сцены, написан код на языке программирования Python для генерации и обработки штрих-кода, установлена связь между двумя средами (Unity и Python).

					<i>ЮУрГУ – 15.03.06. 2023 КР</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>					<i>Разработка системы технического зрения логистики склада</i>			
<i>Провер.</i>		<i>Холодilin И.Ю.</i>						
<i>Н. Контр.</i>								
<i>Утверд.</i>								
					<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>	
							2	24

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1 АНАЛИТИЧЕСКИЙ ОБЗОР ИСПОЛЬЗУЕМЫХ ПРОГРАММ	7
1.1 Python	7
1.2 Unity.....	7
1.3 Связь Unity и Python	8
2 ОПИСАНИЕ МОДЕЛИ UNITY	9
2.1 Компоненты сцены Unity	9
2.2 Конвейер	9
2.3 Появление объектов на конвейерной ленте	11
2.4 Камера	12
2.5 Толкатели.....	13
2.6 Генерация штрих кода.....	16
3 РЕАЛИЗАЦИЯ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ	19
3.1 Запуск python скрипта из Unity	19
3.2 Описание используемого Python скрипта	19
3.3 Обзор используемых в Python библиотек	21
ЗАКЛЮЧЕНИЕ	23
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	24

ВВЕДЕНИЕ

Компьютерное зрение (иначе техническое зрение) — теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов.

Как научная дисциплина, компьютерное зрение относится к теории и технологии создания искусственных систем, которые получают информацию из изображений. Видеоданные могут быть представлены множеством форм, таких как видеопоследовательность, изображения с различных камер или трехмерными данными, например, с устройства Kinect или медицинского сканера.

Как технологическая дисциплина, компьютерное зрение стремится применить теории и модели компьютерного зрения к созданию систем компьютерного зрения. Примерами применения таких систем могут быть:

Системы управления процессами (промышленные роботы, автономные транспортные средства).

Системы видеонаблюдения.

Системы организации информации (например, для индексации баз данных изображений).

Системы моделирования объектов или окружающей среды (анализ медицинских изображений, топографическое моделирование).

Системы взаимодействия (например, устройства ввода для системы человеко-машинного взаимодействия).

Системы дополненной реальности.

Вычислительная фотография, например, для мобильных устройств с камерами.

Компьютерное зрение также может быть описано как дополнение (но не обязательно противоположность) биологическому зрению. В биологии изучается зрительное восприятие человека и различных животных, в результате чего создаются модели работы таких систем в терминах физиологических процессов. Компьютерное зрение, с другой стороны, изучает и описывает системы компьютерного зрения, которые выполнены аппаратно или программно.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.№	ПодписьП	Дата		4

Междисциплинарный обмен между биологическим и компьютерным зрением оказался весьма продуктивным для обеих научных областей.

Подразделы компьютерного зрения включают воспроизведение действий, обнаружение событий, слежение, распознавание образов, восстановление изображений и некоторые другие.

Классическая задача в компьютерном зрении, обработке изображений и машинном зрении это определение содержат ли видеоданные некоторый характерный объект, особенность или активность. Эта задача может быть достоверно и легко решена человеком, но до сих пор не решена удовлетворительно в компьютерном зрении в общем случае: случайные объекты в случайных ситуациях.

Существующие методы решения этой задачи эффективны только для отдельных объектов, таких как простые геометрические объекты (например, многогранники), человеческие лица, печатные или рукописные символы, автомобили и только в определенных условиях, обычно это определённое освещение, фон и положение объекта относительно камеры.

В литературе описано различное множество проблем распознавания:

Распознавание: один или несколько предварительно заданных или изученных объектов, или классов объектов могут быть распознаны, обычно вместе с их двумерным положением на изображении или трехмерным положением в сцене.

Идентификация: распознается индивидуальный экземпляр объекта. Примеры: идентификация определенного человеческого лица или отпечатка пальцев, или автомобиля.

Обнаружение: видеоданные проверяются на наличие определенного условия. Например, обнаружение возможных неправильных клеток или тканей в медицинских изображениях. Обнаружение, основанное на относительно простых и быстрых вычислениях иногда используется для нахождения небольших участков в анализируемом изображении, которые затем анализируются с помощью приемов, более требовательных к ресурсам, для получения правильной интерпретации.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
						5
Изм.	Лист	№ докум.№	ПодписьП	Дата		

Существует несколько специализированных задач, основанных на распознавании, например:

Поиск изображений по содержанию: нахождение всех изображений в большом наборе изображений, которые имеют определенное содержание. Содержание может быть определено различными путями, например, в терминах схожести с конкретным изображением (найдите мне все изображения похожие на данное изображение), или в терминах высокоуровневых критериев поиска, вводимых как текстовые данные (найдите мне все изображения, на которых изображено много домов, которые сделаны зимой и на которых нет машин).

Оценка положения: определение положения или ориентации определённого объекта относительно камеры. Примером применения этой техники может быть содействие руке робота в извлечении объектов с ленты конвейера на линии сборки.

Оптическое распознавание знаков: распознавание символов на изображениях печатного или рукописного текста, обычно для перевода в текстовый формат, наиболее удобный для редактирования или индексации (например, ASCII).

Оптическое распознавание дорожных знаков [1]

В данной работе будет произведен аналитический обзор используемых программ и языков программирования, а также анализ. Также будет рассмотрен процесс реализации системы логистики склада в Unity.

Целью курсовой работы является разработка системы технического зрения логистики склада.

Для этого нам необходимо:

- создать виртуальную модель склада с системой распределения объектов;
- осуществить распознавание штрих-кода каждого объекта;
- установить связь между средами для моделирования логистики;
- проверить работоспособность симулятора;

1 АНАЛИТИЧЕСКИЙ ОБЗОР ИСПОЛЬЗУЕМЫХ ПРОГРАММ

1.1 Python

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++. [2]

1.2 Unity

Unity — кроссплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Выпуск Unity состоялся в 2005 году и с того времени идёт постоянное развитие.

Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
						7
Изм.	Лист	№ докум.№	ПодписьП	Дата		

На Unity написаны тысячи игр, приложений, визуализации математических моделей, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями. [3]

1.3 Связь Unity и Python

Сценарии на Python облегчают взаимодействие Unity с различными приложениями для медиа и индустрии развлечений и гарантируют, что вы сможете легко интегрировать Unity в более широкий производственный конвейер. Потенциальные преимущества использования Python в вашем проекте Unity включают:

- Автоматизация сборки сцен и последовательностей в контексте использования Unity в качестве средства визуализации в реальном времени.
- Автоматизация трудоемких задач, таких как дублирование или модификация большого количества ресурсов.
- Разработка пользовательских инструментов редактирования, которые могли бы использовать сложные элементы пользовательского интерфейса и взаимодействовать изначально с другими наборами DCC.

Сценарии на Python обеспечивают:

- Окно редактора сценариев на Python, в основном предназначенное для технических специалистов, которые хотят запускать короткие сценарии и легко создавать новые пункты меню, которые запускают код на Python.
- API, предназначенный для технических директоров Pipeline, которые хотят вызывать скрипты studio, реализованные на Python, из C # и наоборот.
- Настройки проекта.
- Установка и загрузка Pip-пакетов. [4]

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.№	ПодписьП	Дата		8

2 ОПИСАНИЕ МОДЕЛИ UNITY

2.1 Компоненты сцены Unity

Для реализации проекта компьютерного зрения в программе Unity, мы создали 3D-модель конвейера, который содержит общую направляющую, 3 сортировочных пути объектов, 3 направляющих толкателя, датчик, камеру для считывания штрих-кода на каждом объекте сортировки.

Объектами сортировки являются боксы со штрих-кодами, созданные в программе Blender (рис. 1). [5]

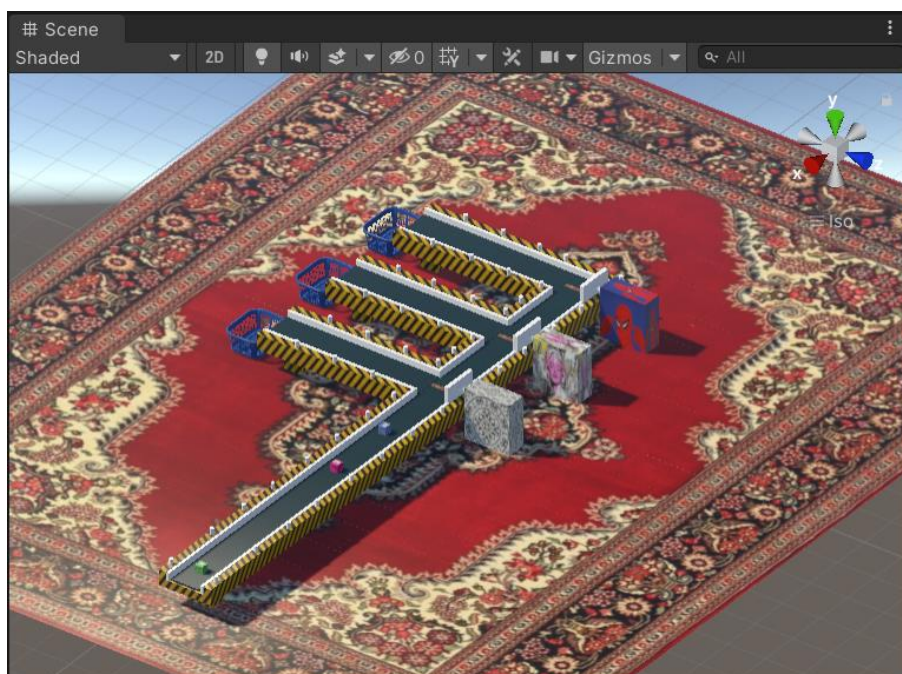


Рисунок 1 – Сцена в Unity.

2.2 Конвейер

Так как в нашем курсовом проекте по логистике склада используется сортировка объектов по штрих коду, нам необходимо реализовать движение объектов по конвейеру. Для того чтобы решить проблему движения кубиков по конвейерной ленте необходимо создать скрипт #С. На рисунке 2 представлен данный скрипт.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Conveir1 : MonoBehaviour
{
    public Rigidbody rb;
    public float speed;
    public Material mt;
    public float direction;
    // Start is called before the first frame update
    void FixedUpdate()
    {
        mt.mainTextureOffset = new Vector2(Time.time * 10 * Time.deltaTime, 0f);
        Vector3 pos = rb.position;
        rb.position += Vector3.right * speed * direction * Time.fixedDeltaTime;
        rb.MovePosition(pos);
    }
}

```

Рисунок 2 – Программный код для реализации движения конвейерной ленты.

Данный скрипт помещаем в GameObject и присваиваем четырем конвейерным линиям. После этого мы вводим несколько переменных: Rigidbody, float speed, Material mt, float direction. Вся функция выполняется с помощью FixedUpdate, так как используется физика. Сначала мы обращаемся к материалу и такому свойству как к смещению текстуры. Смещение мы найдем с помощью Vector2, где прописываем время, умноженное на скорость и умноженное на дельту времени. Потом мы создаем Vector3 и присваиваем ему позицию компонента Rigidbody. Далее к позиции Rigidbody мы прибавляем прямую вектор умноженную на скорость умноженная на фиксированную дельту времени и умноженную на угол поворота. В конце мы обращаемся к компоненту Rigidbody, выбираем метод MovePosition, и снова обращаемся к новому созданному Vector3. Тем самым задавая переменные скорости и угла, мы можем регулировать движение объектов по конвейеру.

2.3 Появление объектов на конвейерной ленте

Для реализации поступления объектов на конвейере необходимо написать функцию их случайного появления, так как именно случайное появление в большей мере отражает реальную систему логистики склада.

Для начала в Unity создадим папку Prefabs, в которую вложим все имеющиеся объекты сортировки. Далее создадим на сцене объект GameObject и расположим его в начале конвейерной ленты. Затем необходимо создать скрипт #C. На рисунке 3. представлен данный скрипт.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Randombox : MonoBehaviour
6  {
7      public GameObject[] boxes;
8      // Start is called before the first frame update
9      void Start()
10     {
11
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17
18         if (Input.GetMouseButtonDown(0))
19         {
20             int randomnumber = Random.Range(0, boxes.Length);
21             Instantiate(boxes[randomnumber], transform.position, Quaternion.identity);
22         }
23     }
24 }
```

Рисунок 3 – Программный код для случайного появления объекта.

Построчное описание кода:

1. Импортируем необходимые библиотеки:
using System.Collections; using System.Collections.Generic; using UnityEngine;
2. Создаем класс Randombox, который наследует MonoBehaviour.
3. Создаем публичный массив GameObject'ов с именем boxes.
4. В методе Start() ничего не делаем.

5. В методе Update() проверяем, была ли нажата левая кнопка мыши (Input.GetMouseButtonDown(0)).

6. Если кнопка была нажата, создаем переменную randomnumber и присваиваем ей случайное число в диапазоне от 0 до длины массива boxes

(int randomnumber = Random.Range(0, boxes.Length);).

7. Используя метод Instantiate, создаем копию объекта из массива boxes с индексом, равным случайному числу randomnumber, в текущей позиции transform.position и с поворотом Quaternion.identity

(Instantiate(boxes[randomnumber],transform.position,Quaternion.identity);).

Данный скрипт помещаем в GameObject. Так как функция Randombox имеет класс публичный массив, то мы можем добавить имеющиеся коробки из папки Prefabs в GameObject поэлементно. Теперь функция работает.

2.4 Камера

Вся работа компьютерного зрения заключена именно в камере, по области считывания (рис.4) которой происходит обработка изображения (рис.5) скриптом Python.



Рисунок 4 - Расположение и область считывания камеры.

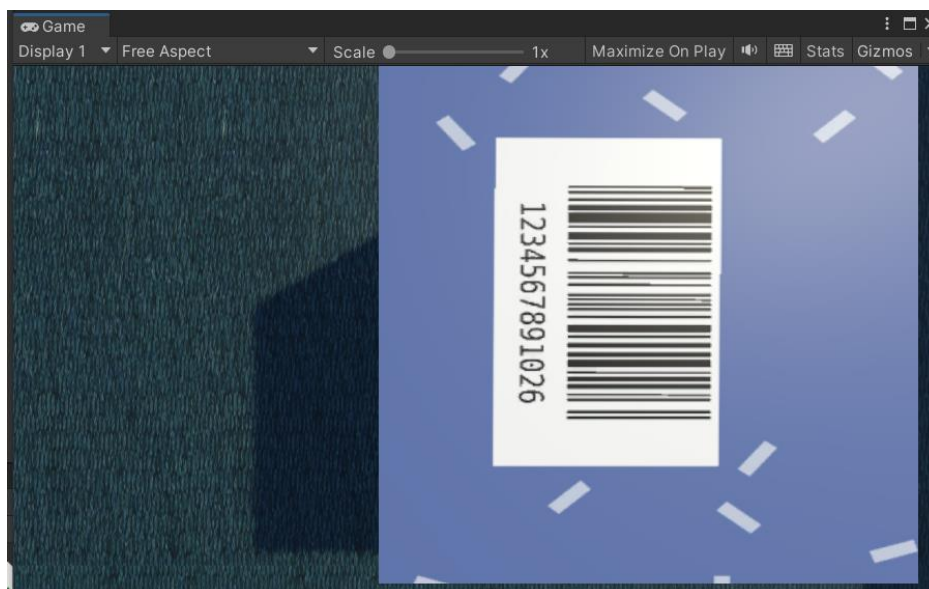


Рисунок 5 - Момент считывания штрих-кода.

2.5 Толкатели

Для реализации работы толкателя, мы создали 3 одинаковых скрипта и каждому толкателю присвоили по скрипту. Рассмотрим работу скрипта более подробно (Move1ForwardAndBack.cs):

1. Объявление переменных:

- score (целое число) - инициализируется значением 4. Эта переменная используется в условии движения вперед.
- distance (вещественное число) - инициализируется значением 5f (5 в формате с плавающей запятой). Это расстояние между начальной позицией объекта и конечной позицией.
- speed (вещественное число) - инициализируется значением 1f. Определяет скорость движения объекта.
- startPosition (векторное значение) - хранит начальную позицию объекта.
- moving Back (логическое значение) - инициализируется значением true. Указывает, движется ли объект назад.
- movement Count (целое число) - инициализируется значением 0. Эта переменная используется для отслеживания количества циклов движения.

2. Метод Start():

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.	№	Подпись П		13

- Вызывается при запуске объекта.
- Присваивает переменной startPosition текущую позицию объекта.

3. Метод Update():

- Вызывается каждый кадр игры.
 - Если movingBack равно true, выполняется следующее:
 - Проверяется условие `score == 1`. Если оно истинно:
 - Изменяется позиция объекта, перемещая его вперед (`transform.position -= transform.forward * speed * Time.deltaTime`).
 - Проверяется расстояние между текущей позицией объекта и начальной позицией (`Vector3.Distance(transform.position, startPosition)`).
 - Если расстояние больше или равно distance, то объект достиг конечной позиции и переменная moving Back устанавливается в false.
 - movement Count увеличивается на 1, чтобы отслеживать количество выполненных циклов движения.
 - Если movingBack равно false, выполняется следующее:
 - Изменяется позиция объекта, перемещая его назад (`transform.position += transform.forward * speed * Time.deltaTime`).
 - Проверяется расстояние между текущей позицией объекта и начальной позицией.
 - Если расстояние меньше или равно 0.1, то объект вернулся в начальную позицию. В этом случае переменная movingBack устанавливается в true, movementCount увеличивается на 1, а score возвращается к исходному значению 4.
- ### 4. Код после блока условий движения:
- Задается переменная maxMovements (целое число), которое определяет желаемое количество циклов движения.
 - Проверяется условие `movement Count >= max Movements`. Если оно истинно, то компонент Move 1 Forward And Back отключается, устанавливая enabled в

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.№	ПодписьП	Дата		14

false. Это означает, что компонент перестает обновляться и прекращает движение объекта.

Также проанализируем код из файла BarcodeScanner, связанный с толкателями.

После обработки штрих кода, переменная bk становится равна тому значению, на какую линию необходимо отправить объект. Работа данного участка кода показана на рисунке 6.

```
179         if (result == "1234567891026")
180         {
181             bk = 1;
182             print("1234567891026");
183         }
184         else if (result == "2345678910212")
185         {
186             bk = 2;
187             print("2345678910212");
188         }
```

Рисунок 6 – Часть кода, отвечающая за распределение объектов после обработки штрих кода.

Когда объект доходит до датчика под нужным толкателем, в данном примере до первого, то происходит вызов метода Scr1().

Рассмотрим данный метод более подробно:

- С помощью цикла foreach мы производим изменения в коде для всех объектов, которым присвоен скрипт MoveForwardAndBack.
- $bk = 0$; - сбрасываем значение, отвечающее за определение линии, на которую будет направлен объект
- `moveComponent.enabled = true;` - включается компонент MoveForwardAndBack, устанавливая значение свойства `enabled` в `true`. Это позволяет компоненту продолжать обновляться и выполнять свою функциональность.
- Обнуляется значение переменной `movementCount` объекта MoveForwardAndBack. Это используется для отслеживания количества циклов движения объекта.

Изм.	Лист	№ докум.№	ПодписьП	Дата

ЮУрГУ - 15.03.06. 2023. КР

- Устанавливается значение переменной score объекта Move1ForwardAndBack равным 1. Это изменяет значение переменной score, которая влияет на условие движения вперед в компоненте Move1ForwardAndBack.

```

85     void Scr1()
86     {
87         // Изменяем score во всех объектах Move1ForwardAndBack
88         foreach (Move1ForwardAndBack moveComponent in moveComponents)
89         {
90             bk = 0;
91             moveComponent.enabled = true;
92             moveComponent.movementCount = 0;
93             moveComponent.score = 1;
94         }
95     }

```

Рисунок 7 – код метода Scr1().

2.6 Генерация штрих кода

Рассмотрим код, с помощью которого мы сгенерировали штрих коды для данного проекта. Данный код продемонстрирован на рисунке 8.

```

1  #генерация штрих кода
2  import barcode
3  from barcode.writer import ImageWriter
4
5  ean = barcode.get('ean13', '345678910212')
6  ean.get_fullcode()
7  filename = ean.save('img/ean13_svg3')
8  filename = ean.save('ean13', options)

```

Рисунок 8 – код для генерации штрих кодов.

Данный код использует библиотеку barcode для создания и сохранения штрих-кодов различных типов.

В первых двух строках импортируются необходимые модули: barcode для создания штрих-кодов и ImageWriter для сохранения штрих-кодов в виде изображений.

Далее определяется переменная ean, которая создает объект штрих-кода типа EAN-13 (European Article Number 13). В данном случае, переданный аргумент '345678910212' является строкой, которая будет преобразована в штрих-код.

Затем вызывается метод get_fullcode(), который возвращает полный код штрих-кода (в данном случае '345678910212').

Далее, вызывается метод save() для сохранения штрих-кода в виде файла. В первом случае, штрих-код сохраняется в папку 'img' с именем файла 'ean13_svg3' в формате SVG. Во втором случае, используются параметры 'ean13' и 'options' для сохранения штрих-кода.

Затем создается новый объект штрих-кода типа EAN-13 с использованием ImageWriter(). Здесь указан аргумент writer=ImageWriter(), чтобы задать тип сохраняемого файла как изображение.

Наконец, вызывается метод save() для сохранения штрих-кода в виде файла изображения. В данном случае, штрих-код сохраняется в папку 'img' с именем файла 'ean13_png3' в формате PNG.

Рассмотрим используемый тип штрих кодов EAN-13. Изображение данного штрих кода представлено на рисунке 9.



Рисунок 9 – пример штрих кода типа EAN-13.

Штрих-код типа EAN-13 (European Article Number 13) используется для кодирования и идентификации товаров и продуктов. Он состоит из 13 цифр и является одним из наиболее распространенных типов штрих-кодов.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
						17
Изм.	Лист	№ докум.№	ПодписьП	Дата		

Код начинается с трех цифрового префикса, который обычно указывает на страну, в которой был выпущен товар. Затем следуют девять цифр, которые кодируют информацию о конкретном товаре. Последняя цифра является контрольной суммой, используемой для проверки правильности чтения штрих-кода. Именно по этой причине при генерации мы назначили 12 цифр. Программа сама выбрала и назначила контрольную сумму для штрих кода.

Штрих-код типа EAN-13 широко применяется в розничной торговле, логистике и управлении запасами. Он позволяет эффективно и точно идентифицировать товары и автоматизировать процессы продажи, отслеживания и учета товаров.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.	№	Подпись	Дата	18

3 РЕАЛИЗАЦИЯ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Рассмотрим поэтапно, как реализуется система логистики склада в Unity.

3.1 Запуск python скрипта из Unity

Когда объект дошёл до первого датчика и сработала камера, которая сохранила изображение объекта в папке img.

Затем запускается Python скрипт для обработки изображения. Код в unity для запуска python скрипта представлен на рисунке 10.

```
131 // Запускаем Python-скрипт для обработки фотографии
132 Process process = new Process();
133 process.StartInfo.FileName = pythonPath;
134 process.StartInfo.Arguments = scriptPath;
135 process.StartInfo.UseShellExecute = false;
136 process.Start();
```

Рисунок 10 – Код в unity для запуска python скрипта.

Данный код создает объект Process, указывает путь к исполняемому файлу Python, а затем запускает внешний процесс, который будет выполнен с использованием интерпретатора Python. Давайте более подробно рассмотрим код python.

3.2 Описание используемого Python скрипта

Для анализа штрих кода, мы использовали язык python. Для обработки штрих кода мы создали файл read.py и написали в нём такой код (рис. 11)

```

13  #чтение штрих кода
14  from pyzbar.pyzbar import decode
15  from PIL import Image
16
17  image_barcode = Image.open("C:/Users/LexaD/Documents/Kursovay/img/image.png")
18  decoded = decode(image_barcode)
19  f = open('C:/Users/LexaD/Documents/Kursovay/result.txt','w') # открытие в режиме записи
20  f.write(decoded[0].data.decode("utf-8"))
21  f.close()

```

Рисунок 11 – Код на python для обработки штрих кодов

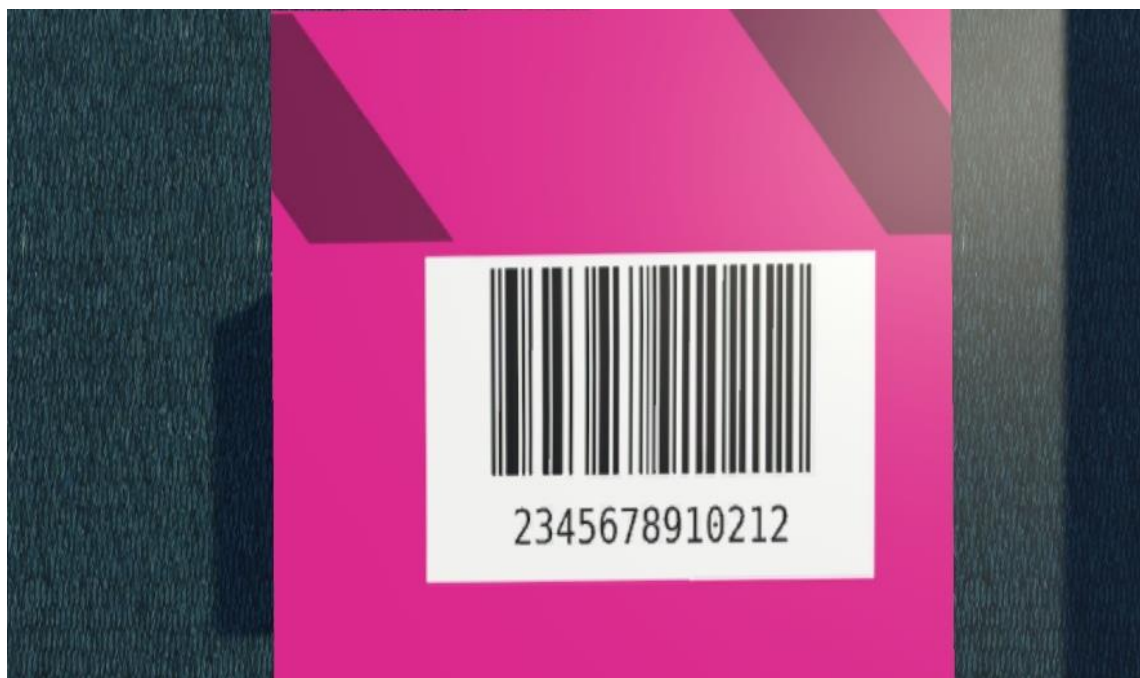


Рисунок 12 – Изображение штрих-кода, поступающего на обработку в Python.

Построчно рассмотрим данный код:

- Строка `from pyzbar.pyzbar import decode` импортирует функцию `decode` из модуля `pyzbar` для распознавания штрих-кодов.
- Строка `from PIL import Image` импортирует класс `Image` из модуля `PIL` (Python Imaging Library) для работы с изображениями.
- Строка `image_barcode = Image.open("C:/Users/LexaD/Documents/Kursovay/img/image.png")` открывает изображение с путем `"C:/Users/LexaD/Documents/Kursovay/img/image.png"` с помощью метода `open` из класса `Image`. Открываемое изображение присваивается переменной `image_barcode`.

- Строка `decoded = decode(image_barcode)` вызывает функцию `decode` из модуля `pyzbar`, передавая ей переменную `image_barcode`, чтобы распознать штрих-коды на изображении. Результат распознавания сохраняется в переменной `decoded`.
- Строка `f = open('C:/Users/LexaD/Documents/Kursovay/result.txt','w')` открывает файл с путем `"C:/Users/LexaD/Documents/Kursovay/result.txt"` в режиме записи. Файл, который будет использоваться для сохранения распознанного штрих-кода, присваивается переменной `f`.
- Строка `f.write(decoded[0].data.decode("utf-8"))` записывает распознанные данные штрих-кода в файл, используя метод `write` объекта файла `f`. `decoded[0].data` представляет собой данные первого распознанного штрих-кода, а `decode("utf-8")` преобразует байтовую строку в строку Unicode.
- Строка `f.close()` закрывает файл, освобождая ресурсы и сохраняя записанные данные.

Рассмотрим более подробно используемые библиотеки.

3.3 Обзор используемых в Python библиотек

Модуль PIL (Python Imaging Library) представляет собой библиотеку для работы с изображениями в Python. Он предоставляет различные функции и классы, которые позволяют открывать, изменять, сохранять и обрабатывать изображения различных форматов.

Модуль `pyzbar` использует алгоритмы библиотеки ZBar для распознавания штрих кодов на изображении. ZBar является популярной и широко используемой библиотекой для сканирования и распознавания штрихкодов.

ZBar поддерживает распознавание различных типов штрихкодов, включая одномерные (например, EAN-13, UPC-A, Code 39) и двухмерные (например, QR-коды, Data Matrix, PDF417) штрихкоды.

Алгоритмы распознавания штрихкодов в ZBar обычно основываются на анализе яркости и контраста изображения. Они пытаются найти и расшифровать

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.№	ПодписьП	Дата		21

штрихкоды, обнаруживая и интерпретируя черные и белые полосы или узоры, составляющие штрихкод.

Алгоритмы ZBar могут включать в себя операции обнаружения контуров, выравнивания штрихкода, поиска синхронизационных маркеров и интерпретации данных штрихкода на основе его уникальной структуры и формата.

В нашем проекте, мы использовали тип штрих кода EAN-13. Рассмотрим общий алгоритм распознавания данного типа штрих кодов.

В общем случае, распознавание штрихкодов EAN-13 включает следующие шаги:

Обнаружение штрихкода: Алгоритмы обнаружения штрихкода сканируют изображение, чтобы найти потенциальные области, где может находиться штрихкод EAN-13. Это может включать операции поиска вертикальных и горизонтальных черных полос, которые образуют штрихкод.

Выравнивание штрихкода: После обнаружения штрих кода алгоритмы выравнивают его, чтобы обеспечить правильное положение и ориентацию штрихкода для дальнейшего распознавания.

Извлечение данных: Алгоритмы извлекают информацию из черных и белых полос штрихкода. В случае EAN-13, данные включают код страны, производителя и продукта.

Проверка контрольной суммы: Для EAN-13 важным шагом является проверка контрольной суммы, чтобы убедиться в правильности распознанных данных штрихкода. Контрольная сумма используется для проверки корректности остальных цифр в штрих коде.

Декодирование данных: На основе извлеченных и проверенных данных, алгоритмы выполняют декодирование штрихкода EAN-13 для получения фактического числового значения, представляющего штрихкод.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.№	ПодписьП	Дата		22

ЗАКЛЮЧЕНИЕ

Одним из недостатков симулятора является то, что расстояние между объектами (коробками) должно быть равно или больше длины толкающей поверхности, иначе он будет захватывать и соседние коробки, то есть должно быть фиксированное время срабатывания толкателя, что сказывается на общем времени сортировки.

По ходу курсового проекта разработан симулятор логистики склада:

- Выполнены 3D – модели объектов в Blender.
- Смоделирована сцена в Unity.
- Сгенерированы штрих-коды для считывания камеры и распознавания кодом в Python.
- Отлажена работа конвейера и толкателей с помощью #C и связки Unity-Python.

Достоинством является то, что обработка штрих-кода происходит мгновенно, без остановки конвейера, что увеличивает производительность склада.

В заключение курсовой работы можно отметить, что использование автоматизированной системы сортировки товаров позволяет существенно ускорить процесс и уменьшить количество ошибок, связанных с человеческим фактором. Однако, для успешной реализации такой системы необходимо учитывать множество факторов, таких как выбор подходящего оборудования, разработка оптимального алгоритма работы и обучение персонала. В целом, автоматизация процесса сортировки товаров является важным шагом в повышении эффективности логистики склада и улучшении качества обслуживания клиентов.

					ЮУрГУ - 15.03.06. 2023. КР	Лист
Изм.	Лист	№ докум.	№	Подпись П		23

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [1] https://ru.wikipedia.org/wiki/Компьютерное_зрение#Распознавание
- [2] <https://ru.wikipedia.org/wiki/Python>
- [3] [https://ru.wikipedia.org/wiki/Unity_\(игровой_движок\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок))
- [4] <https://docs.unity3d.com/Packages/com.unity.scripting.python@6.0/manual/index.html>
- [5] <https://www.blender.org/>

					<i>ЮУрГУ - 15.03.06. 2023. КР</i>	Лист
						24
Изм.	Лист	№ докум.	Подпись П	Дата		