

1. ФИНАЛЬНЫЙ ЭТАП

Задача командного тура

1.1. Легенда

С каждым днем в нашем мире задействуется все больше современных технологий. Еще 150 лет назад не существовало машин, а сегодня мы встречаем беспилотные автомобили и роботов доставщиков. Такие технологии позволяют передать роботам некоторые задачи выполняемые людьми, которые требуют большого внимания или могут быть опасны для здоровья. Таким образом, будет минимизироваться человеческий фактор, а вслед за этим повышаться скорость и точность работы.

Уже сейчас существуют роботы -работники в разных сферах, например, робот хирург, спасающий жизни людей, робот - доставщик, робот - помощник и другие. В промышленной среде широко распространены роботы-манипуляторы. Манипулятор — устройство для перемещения объектов без прямого воздействия человека. Но для каждого такого манипулятора нужны разработчики, способные "оживить"его и дать ему указания для работы. В основе задач профиля «Интеллектуальные робототехнические системы» лежит работа с манипулятором.

Для решения задачи участникам потребуются знания линейной алгебры и аналитической геометрии, умение работать с компьютерным зрением, а также навыки взаимодействия с манипулятором.

Задача финала заключается в следующем: На Заводе №6 занимаются изготовлением различных металлических деталей путем литья. Процесс устроен таким образом, что за одну итерацию изготавливаются несколько различных типов изделий, которые перемешаны между собой и хранятся все вместе. Сортировка деталей вручную возможна, но этот процесс занимает слишком много времени и Завод рискует не успеть сделать планируемое количество в срок, а другие методы слишком дороги или неэффективны. По стечению обстоятельств, на складе Завода лежит подозрительно непригодный к работе манипулятор, и он достался вам вместе с задачей по сортировке деталей. К счастью для вас, конфигурация манипулятора известна, а детали, которые вам поручили сортировать, оказались маленькими блоками двух разных цветов. Однако, места на складах вам выделили немного, а среди блоков периодически проскакивает брак.

Ваша задача — при помощи известного манипулятора распределить кучу блоков двух разных цветов в заранее известные места. При сортировке блоки желательно складывать в виде башен, чтобы оптимизировать пространство.

1.2. Набор заданий

Задача: В наличии робота в начальной зоне которого располагается навал из 8 блоков двух разных цветов. В навале помимо блоков присутствует 4 бракованных объекта, но они всё ещё отличаются от обычных блоков формой. Однако теперь сортировать данные объекты можно. Также, теперь для экономии места блоки необходимо сложить так, чтобы как можно меньше из них соприкасалось с изначальной зоной (например в виде башни). При этом складывать блоки всё ещё необходимо по цветам (т.е. так, чтобы объекты разных цветов не соприкасались.) Для такого складывания объектов всё ещё будут отведены 2 зоны двух цветов, совпадающих с цветами блоков.

Включая содержательные задачи:

- Реализация алгоритмов для определения формы объектов
- Реализация алгоритмов определения отдельных объектов при их наложении.
- Реализация алгоритмов планирования определения последовательности захвата объектов на основе их пространственного расположения.
- Реализация алгоритмов захвата объектов роботом при усложненном пространственном расположении объектов.
- Реализация высокоточного и аккуратного перемещения робота
- Реализация алгоритмов составления объектов

1.3. Описание системы

Робот - манипулятор, способный выполнять вращательные и поступательные движения. Манипулятор оснащен камерой, для распознавания объектов расположенных на столе.

Рабочая / достижимая область — пространство стола, в котором манипулятор способен взаимодействовать с объектами. Непосредственно в этой зоне производятся все действия манипулятора, и ограничиваются ей. В рабочей области расположены начальная зона и места сортировки.

Блок — объект, который располагается внутри рабочей области робота и имеет форму прямоугольного параллелепипеда $50 \times 25 \times 20$ мм. Существует два вида блоков: красные блоки и синие блоки.

Начальная зона — прямоугольная зона 40×30 см, расположенная внутри рабочей области. В данной зоне изначально располагаются все блоки с которыми манипулятор взаимодействует. Внутри стартовой зоны блоки могут располагаться навалом: блоки имеют случайную ориентацию и местоположение внутри данной области. Также допускается их расположение друг на друге.

Место хранения — квадратная зона 10×10 см, расположенная внутри рабочей области. Всего на столе расположено два таких места. Места хранения имеют постоянные координаты.

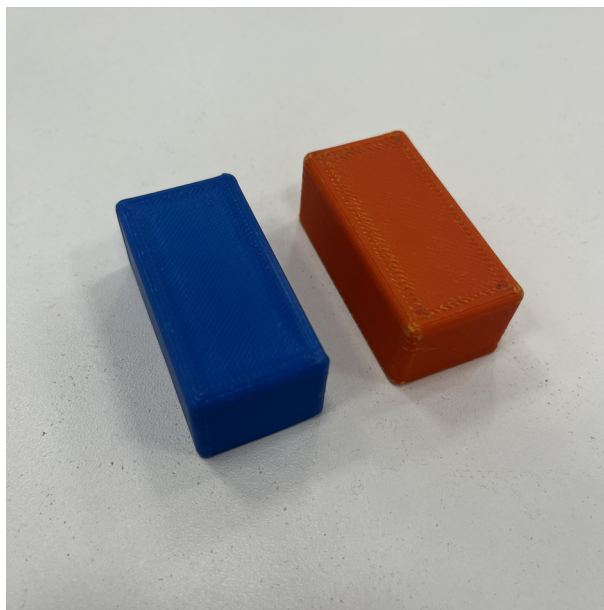


Рис. 1.1: Примеры блоков

1.4. Описание робота

На финальном туре команды используют реального робота. Робот представляет собой манипулятор UR10e, оснащенный камерой realsense.

Манипулятор UR10e имеет 6 степеней свободы. Полную официальную документацию манипулятора можно найти здесь (источник на английском): https://s3-eu-west-1.amazonaws.com/ur-support-site/27484/UR10_User_Manual_en_E670N_Global-3.4.5.pdf

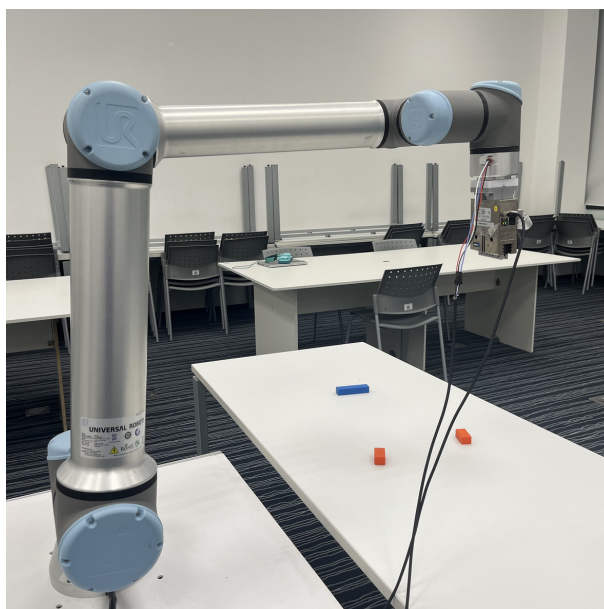


Рис. 1.2: Манипулятор UR10e

На манипулятор установлен двухпальный мягкий захват, который имеет два состояния — открыт и закрыт. Гарантируется, что при помощи данного захвата можно захватить любые представленные в задачах объекты тем или иным образом.

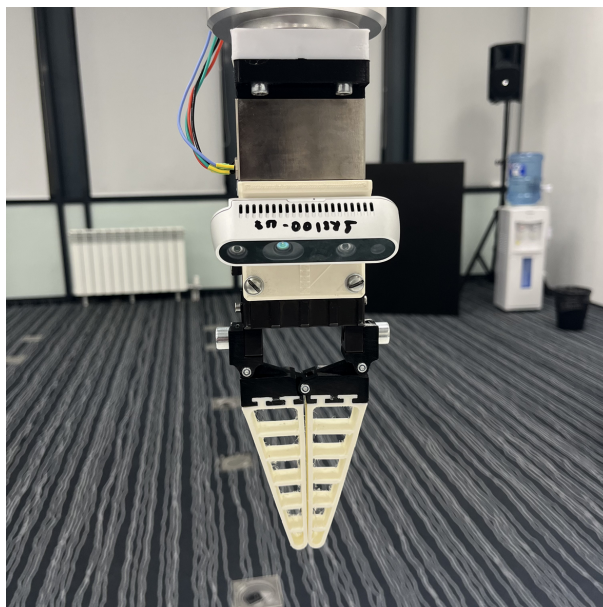


Рис. 1.3: Захват

Робот управляется при помощи python - библиотеки `python-urx` (<https://github.com/jkur/python-urx>), а для работы с камерой помогает python - библиотека `open3d` (http://www.open3d.org/docs/0.9.0/tutorial/Basic/python_interface.html). Однако для удобства участников существуют специальные скрипты - прослойки, упрощающие работу.

Участникам предоставляется библиотека **“OperateRobot.py”**, в которой лежит класс `OperateRobot`, содержащий следующие методы:

- `move1` — перемещает робота в точку с координатой относительно основания робота и заданными углами.
- `get1` — получает координаты робота относительно основания робота
- `getj` — получает координаты робота в пространстве звеньев манипулятора (joint space)
- `close` — закрывает соединение с роботом
- `open_gripper` — открывает захват
- `close_gripper` — закрывает захват

Также участникам предоставляется библиотека **“OperateCamera.py”**, в которой лежит класс `OperateCamera`, содержащий следующие методы:

- `catch_frame` — получает один снимок с камеры
- `save` — сохраняет последний снимок полученный `catch_frame` в файл с указанным именем
- `stop` — завершает работу с камерой

Файл **“tutorial.py”** позволяет познакомиться с основными методами и способами работы с роботом и камерой.

Все описанные выше вспомогательные файлы доступны по ссылке <https://bit.ly/3JvurbW>.

1.5. Условия проведения

1. Участники во время командного этапа финального тура могут использовать интернет и заранее подготовленные библиотеки для решения задачи.
2. В качестве языка программирования робота используется Python 3.9.
3. Участникам рекомендуется начать с библиотек, заданных в файле `requirements.txt`.
4. Не допускается использование скорости робота больше $0.2 \frac{m}{c}$ и ускорения больше $0.2 \frac{m}{c^2}$. В случае если участники будут использовать или попытаются использовать большую скорость они будут дисквалифицированы.
5. Участники обязаны бережно относиться к оборудованию, в частности роботу, блокам и т.д. В случае подозрений на преднамеренную порчу или опасное управление оборудованием организаторы оставляют за собой право штрафовать команду нарушителей на баллы или вовсе снять их с соревнований.
6. В случае возникновения опасности повреждения оборудования организаторы имеют право экстренно прервать исполнение программы участников. Выполнение попытки будет остановлено в тот же момент, на этом она завершится. Баллы за такую попытку присваиваются в соответствии с правилами, но на усмотрение судей.
7. Участники не могут использовать помощь тренера, сопровождающего лица или привлекать третьих лиц для решения задачи.
8. Может быть предусмотрено до двух попыток сдачи решения одной и той же подзадачи в симуляторе. Конкретное количество попыток определяется в конкретных подзадачах.
9. После прохождения приемочных запусков, баллы, набранные командой, заносятся судьями в протокол.
10. Робот должен выполнять задание полностью автономно. Удаленное управление не допускается.
11. Если во время приемочных запусков у судьи возникает ситуация, когда он не может однозначно решить выполняются ли критерии решения подзадачи, он вправе принять решение не в пользу команды.
12. В случае, если возникает техническая проблема, независящая от участников, то по решению судей может быть предоставлена возможность перезапуска.

1.6. Процедура проведения приемочных запусков и критерии оценки

1. Командам необходимо подготовить следующую задачу для робота:
 - 1.1. Робот с известной конфигурацией начинает свое движение. В начальной зоне, достижимой роботом, располагается 12 объектов двух цветов, из них 4 объекта отличаются по форме. Данные объекты располагаются навалом, что означает что у них может быть случайная ориентация и местоположение в данной зоне. Также допускается их расположение друг на друге. Необходимо распределить данные объекты в два заранее известных места. Распределение необходимо выполнить на основании цвета объектов, включая новые объекты. Необходимо сложить сортируемые объекты в виде башни.

- 1.2. Имя файла с управляющей программой для проверки решения на роботе: `main.py`.
- 1.3. Максимальное время выполнения одной попытки для задачи — 5 минут.
- 1.4. На демонстрацию решения предоставляется 2 попытки.
- 1.5. Баллы за решение подзадач этапа, связанных с роботом:
 - 1.5.1. Робот смог переместить один объект в любое из заранее известных мест — 2 балла за объект. Максимальное количество баллов за подзадачу — 24 баллов.
 - 1.5.2. В данном месте располагаются объекты одного цвета — 4 балла за место. Максимальное количество баллов за подзадачу — 8 баллов.
 - 1.5.3. Робот смог переместить один объект в любое из заранее известных мест. Данный объект не касается основания, иначе говоря опирается только на другие объекты, своего цвета — 2 балла. Максимальное количество баллов за подзадачу — 20 баллов.
 - 1.5.4. Робот смог убрать все объекты из начальной зоны — 2 балла.
- 1.6. Баллы за все попытки в каждой подзадаче суммируются.
- 1.7. Выполнение всех критериев (не обязательно на максимум баллов) в каждой из двух попыток задачи на реальном роботе дает дополнительные 2 балла.
2. Все попытки осуществляются 24 марта.
3. Максимальное количество баллов — 110.

1.7. Критерии определения команды-победителя командного тура

1. Сумма баллов, набранных за решения подзадач всех этапов командной части финального этапа,
2. Также командам необходимо подготовить презентацию, демонстрирующую свои результаты. Информация о требованиях к презентации будет позднее.
3. Сумма баллов, набранных за демонстрацию решения, и баллов, полученных за презентацию, определяет итоговую результативность команды (измеряемую в баллах).
4. Команды ранжируются по результативности.
5. Команда-победитель определяется как команда с максимальной результативностью.