

Адаптивный алгоритм управления манипуляционным роботом для построения модели внешней среды

А.Д. Воронков

МИРЭА - Российский технологический университет, г. Москва

Аннотация: Для успешной работы нейросетевых алгоритмов при обработке 3D-облаков точек необходимо обеспечение детального облака точек внешней среды. Подобная задача возникает при функционировании манипуляционного робота в новой среде, где перед обработкой облака точек сцены необходимо получить детальное представление внешней среды с помощью RGB-D камеры, закрепленной на конечном звене робота. Для решения этой проблемы в данном исследовании предлагается алгоритм для адаптивного управления манипуляционным роботом для построения модели внешней среды. За счет применения адаптивного подхода, в процессе исследования внешней среды манипуляционный робот перемещает RGB-D камеру с учетом изменений в текущей модели среды, внесенных предыдущим полученным RGB-D изображением. Полученные результаты позволяют судить об эффективности предложенного подхода, показывая, что за счет адаптивности он позволяет достичь больших показателей покрытия сцены.

Ключевые слова: модель среды, манипуляционный робот, адаптивный алгоритм управления, реконструкция поверхности, RGB-D камера, обработка визуальной информации.

Введение

Развитие манипуляционной робототехники открывает широкие перспективы для автоматизации процесса захвата и манипулирования априорно неизвестными объектами. Подобные задачи возникают при устранении последствий природных и техногенных катастроф, захвате неизвестных объектов в складских помещениях, в задачах сортировки твердых бытовых отходов [1]. Для новых объектов известны различные стратегии синтеза захватов, такие как вывод положения и ориентации захватного устройства [2, 3], полной конфигурации захватного устройства [4, 5] и выборка контактных точек на поверхности объекта [6, 7]. В отличие от первых двух групп подходов, для планирования захвата на уровне контактных точек требуется подробное облако точек объекта манипулирования, что подразумевает знание модели внешней среды. При построении модели внешней среды важен баланс между числом

интегрируемых RGB-D кадров и величиной покрытия рабочей сцены, при этом время каждой итерации алгоритма должно быть минимальным. Вместе с тем при планировании обновленной конфигурации робота требуется учет его кинематических ограничений и ограничений, вызванных присутствием препятствий в рабочей зоне. Вышеописанные факторы определяют проблематику и актуальность решаемой задачи.

В данной работе предложен адаптивный алгоритм управления манипуляционным роботом для построения облака точек внешней среды с использованием визуальной информации с RGB-D камеры, закрепленной на захватном устройстве манипулятора (ЗУМ). Адаптивность подхода заключается в том, что информация, получаемая на каждой новой итерации алгоритма, используется при принятии решения о перемещении робота в следующую конфигурацию, что позволяет адаптироваться к сложной форме априорно неизвестных сред.

Обзор известных работ предметной области

Облака точек являются трехмерным представлением окружающей среды с отражением ее пространственных характеристик, что позволяет определять состав внешней среды и наличие препятствий в ней [8]. Сложность при обработке облаков точек вызывают высокая размерность и объем данных, поэтому с развитием машинного обучения появились нейросетевые архитектуры [9-11], ориентированные на непосредственную работу с облаками точек в задачах сегментации, классификации и детектирования за счет специализированных операций извлечения признаков.

Современные RGB-D камеры, такие как Intel RealSense D435 работают на основе проекции структурированного света. В общем случае, частичного облака точек с одного кадра RGB-D камеры недостаточно для построения

детального представления внешней среды, поэтому требуется разработка алгоритмов управления движением робота, обеспечивающих детализированное представление внешней среды [12].

Рассмотрим алгоритмы, позволяющие осуществлять построение облака точек за счет вычисления нового положения камеры, максимизирующего прирост информации. Данные подходы могут полагаться на известную модель объектов, что позволяет планировать операцию реконструкции офлайн, или предполагают априорную неизвестность среды, позволяя осуществлять построение облака модели среды в онлайн режиме [13-15].

Методы, основанные на использовании воксельной сетки работают по принципу формирования набора потенциальных положений камеры и оценки вклада каждого положения в прирост покрытия облака точек с помощью метода бросания лучей [16]. В [16] описана разработка программно-алгоритмического обеспечения для экспериментальных исследований воксельного подхода и различных методов выбора положения камеры. В [17] предложен подход, при котором управление манипуляционным роботом осуществляется таким образом, чтобы исследовать граничные области текущего представления сцены. Для представления сцены использовалась воксельная сетка, каждому вокселю которой сопоставлено значение усеченной функции расстояния со знаком (Truncated Signed Distance Function — TSDF), результатом чего является модель среды в виде TSDF-объема.

Главными преимуществами подходов, основанных на машинном обучении, являются способность осуществлять обработку данных в реальном времени и производить сложную обработку информации на разных масштабах. В [13] представлен подход для определения нового положения камеры, основанный на самообучении. В данном подходе полносвязная нейронная сеть обрабатывает топологические признаки облака точек для вывода нового положения камеры. В [14] нейронная сеть для выбора

следующего лучшего вида по облаку точек (Point Cloud Next Best View — PC-NBV) выделяет глобальный вектор признаков имеющегося частичного облака точек и определяет положение камеры, дающее максимальный прирост покрытия поверхности объекта. Предложенная нейронная сеть показала устойчивость к шуму в облаке точек, высокую скорость как обработки информации, так и сходимости к полному покрытию сцены. В [18] предложен подход к управлению манипуляционным роботом для перемещения камеры, закрепленной на конечном звене, с помощью оценки энтропии Шеннона. Используется разбиение плоскости стола с предметами на 2D сетку. В данном подходе новое положение камеры определяется областью с большей энтропией. Смещение на основе энтропии производится в плоскости XOY при постоянной величине шага. В [19] предложен подход к выбору положения камеры на основе доступности объекта для захвата манипуляционным роботом с помощью параллельного захватного устройства, для чего задача захвата рассматривается совместно с задачей выбора следующего положения камеры. Для осуществления захвата определенного целевого объекта, подход работает итеративно, используя TSDF-представление сцены и определяя положение камеры, при котором вероятность успеха выданного нейронной сетью захвата достигнет наибольшего значения.

Поскольку при построении модели неизвестной среды манипуляционный робот оперирует в неопределенной заранее обстановке, требуется применения алгоритмов планирования пути. Алгоритмы планирования пути решают задачу поиска пути робота в операционном или конфигурационном пространстве с избеганием препятствий. В манипуляционной робототехнике широко используется семейство алгоритмов быстрорастущего случайного дерева (Rapidly-Exploring Random Tree — RRT) [20].

Существующие решения не рассматривают плотность точек в окрестности точек интереса и их удаленность от центра сцены, что важно для построения детального облака точек рабочей среды для задачи захвата объектов. Предложенный адаптивный алгоритм управления манипуляционным роботом лишен этих недостатков и позволяет эффективно осуществлять построение модели внешней среды априорно неизвестной формы, учитывая также кинематические ограничения робота и обработку препятствий, в том числе обновляющейся модели среды, в рабочей области.

Формальная постановка задачи

Решаемая задача заключается в разработке адаптивного алгоритма управления манипуляционным роботом для построения модели неизвестной внешней среды. Разрабатываемый алгоритм осуществляет управление манипуляционным роботом по последовательности конфигураций $Q = \{Q_1, \dots, Q_k\}$, обеспечивающих покрытие рабочей сцены RGB-D изображениями с камеры глубины. Определение новой целевой конфигурации робота осуществляется за счет фильтрации и сортировки множества потенциально возможных новых положений камеры T_{cam}^w , учитывающего имеющуюся на i -ом шаге модель среды M_i , за счет чего обеспечивается адаптация к неизвестной среде. Управление на стратегическом уровне осуществляется путем определения новой целевой конфигурации, в которую необходимо переместить робота для обновления модели среды. Управление на тактическом уровне осуществляется за счет поиска пути для робота в среде с препятствиями.

При разработке алгоритма были использованы следующие допущения:

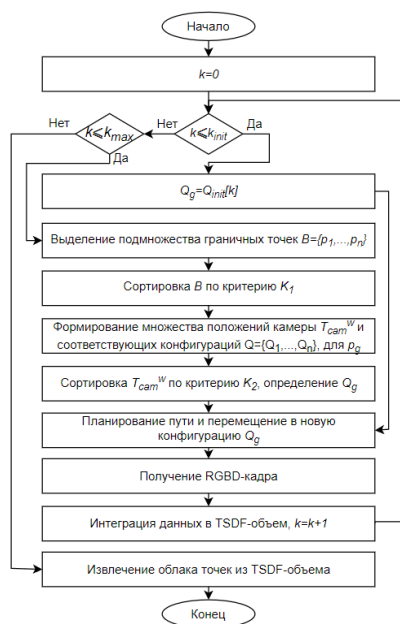
1. Координаты точки C центра сцены (x_c, y_c, z_c) и параметры ограничивающего параллелепипеда $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ для

исследуемой сцены в глобальной системе координат $\{W\}$ известны, причем базовая система координат (СК) робота $\{B\}$ и глобальная СК $\{W\}$ совпадают;

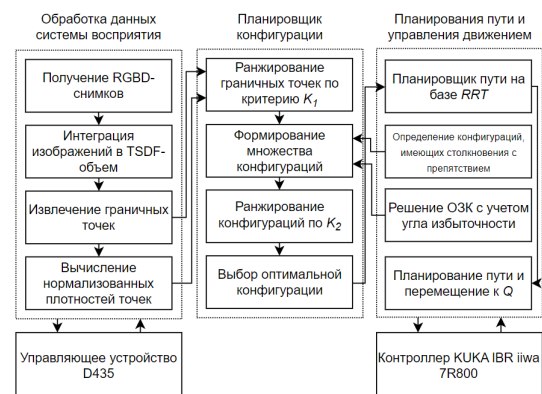
2. Известны расстояния до ближней и дальней плоскостей отсечения RGB-D камеры;
3. Сцена является стационарной.

Описание предлагаемого подхода

Блок-схема предложенного алгоритма приведена на рис. 1а. Структура разработанного программного обеспечения представлена на рис. 1б. Схема системы управления приведена на рис. 2.



а)



б)

Рис. 1. – Предложенный алгоритм для построения модели неизвестной среды: а) блок-схема адаптивного алгоритма управления манипуляционным роботом; б) структура разработанного программного обеспечения

Построение модели среды манипуляционным роботом является комплексной задачей, требующей учета ряда факторов, таких как кинематические ограничения манипуляционного робота, наличие

препятствий и искажения, присутствующие в изображении глубины. Управление манипуляционным роботом должно осуществляться на основе определения новой, наилучшей по какому-либо критерию конфигурации манипулятора.

Как видно из рис. 1а, в начале работы алгоритма управление манипуляционным роботом выполняется по $k_{init} = 2$ предопределенным конфигурациям, поскольку форма рабочей сцены предполагается не известной заранее. Таким образом осуществляется построение первичной модели среды, которая используется для анализа текущих условий и принятия решений о следующей целевой конфигурации робота. Полученные данные интегрируются в текущую модель среды.

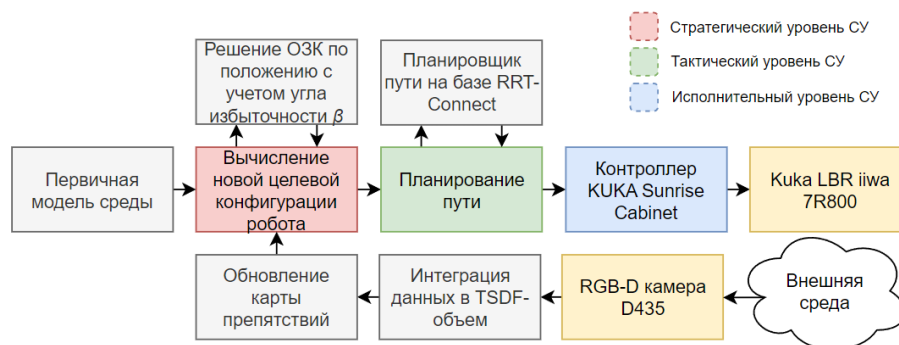


Рис. 2. – Структура системы управления для построения модели внешней среды

Рассмотрим алгоритм определения новой конфигурации манипуляционного робота. Облако точек текущей модели среды M_i кадрируется по заданным параметрам ограничивающего прямоугольного параллелепипеда $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$. С помощью встроенного метода библиотеки Open3D, на основе нормалей точек осуществляется извлечение подмножества граничных точек B , имеющее m элементов. Граничные точки служат индикаторами областей, для которых

требуется получить новые данные. Для каждой граничной точки $p_i \in B$ оценивается нормализованная плотность d_{in} (1) и нормализованное расстояние l_{in} между точками p_i и C . Значение d_{in} и l_{in} лежат в интервале $[0; 1]$.

$$d_{in} = \frac{d_i - d_{min}}{d_{max} - d_{min}}, \quad (1)$$

где d_i – число точек, лежащих в радиусе r_d от p_i ; $d_{min} = \min(\{d_1, \dots, d_m\})$; $d_{max} = \max(\{d_1, \dots, d_m\})$.

Наиболее перспективным с точки зрения построения модели среды являются граничные точки с наименьшей нормализованной плотностью, при этом точки, находящиеся ближе к C , имеют больший приоритет. Множество граничных точек сортируется по критерию (2), выбирается точка с наименьшим значением критерия. При увеличении коэффициента w_d при одинаковой нормализованной плотности больший приоритет будут иметь граничные точки, имеющие меньшее расстояние до C .

$$K_1(p_i) = d_{ni} + w_d l_{ni}, \quad (2)$$

где w_d – весовой коэффициент; d_{ni} – нормализованная плотность для локальной окрестности точки p_i ; l_{ni} – нормализованное расстояние между точками C и p_i .

Для отобранной граничной точки $p \in B$ формируется множество T_{cam}^w потенциальных положений СК камеры, расположенных на сфере радиуса l , причем p лежит на $O_c Z_c$, таких, что для любого $t_{cam}^w \in T_{cam}^w$ существует хотя бы одно решение обратной задачи кинематики (ОЗК) с углом избыточности $\beta \in [-\frac{\pi}{2}; \frac{\pi}{2}]$. Данный способ формирования потенциальных положений камеры связан с тем, что ориентацию камеры требуется привязать к

положению интересующей точки, что позволяет захватывать окрестность данной точке на RGB-D кадре. Множество T_{cam}^w сортируется в соответствии с критерием (3).

$$K_2(t_{cam}^w) = n_{bvi} + n_{hi}, \quad (3)$$

Для реализации данного критерия используется представленный в [21] подход, позволяющий осуществить выделение множества видимых точек $M_{t_{cam}^w}$ в исходном облаке точек для заданного положения оптического сенсора t_{cam}^w . В случае, если RGB-D камера полагается не идеальной, из $M_{t_{cam}^w}$ также удаляются точки, находящиеся за пределами ближней и дальней плоскостей отсечения камеры. Для каждого t_{cam}^w в T_{cam}^w вычисляется число граничных точек n_{bv} , для которых выполняется условие $\widehat{\vec{n}_v, \vec{Z}_c O_c} < \frac{\pi}{2}$, и число n_h точек в $M_{t_{cam}^w}$, для которых выполняется условие $\widehat{\vec{n}_v, \vec{Z}_c O_c} > \frac{\pi}{2}$. Таким образом, критерием для сортировки конфигурации Q_i является сумма количества граничных точек в $M_{t_{cam}^w}$, нормали которых ориентированы по направлению к камере и всех точек в $M_{t_{cam}^w}$, нормали которых направлены от камеры. Определяется новая конфигурация робота $Q_g = IK(t_{cam}^w T_g^c), t_{cam}^w = \operatorname{argmax}_{t \in T_{cam}^w} (K_2(t))$. Визуализация точек, участвующих в оценке конфигурации робота, представлена на рис. 3.

Приведенный подход к выбору новой конфигурации робота учитывает несколько факторов: для прироста информации о внешней среде требуется обеспечивать реконструкцию облака точек в областях с низкой нормализованной плотностью точек; точки, удаленные от заданного центра сцены, представляют меньший интерес, чем точки вблизи центра сцены; новая конфигурация должна быть кинематически осуществимой и не иметь столкновений с имеющейся моделью среды и другими препятствиями.

При решении задачи построения модели статической сцены используется статический планировщик пути, использующий определенные ранее модели робота и статических препятствий [20]. Формируемая модель среды также представляет собой препятствие, добавляемое в планировщик пути путем вычисления выпуклой оболочки для облака точек текущей модели среды. Данный планировщик пути осуществляет планирование траектории в конфигурационном пространстве, состоящий из точек. Была использована реализация однозапросного планировщика RRT-Connect открытой библиотеки планирования движения (The Open Motion Planning Library — OMPL) [22, 23]. Траектория, полученная при планировании пути, далее оптимизируется и сглаживается встроенными средствами OMPL.

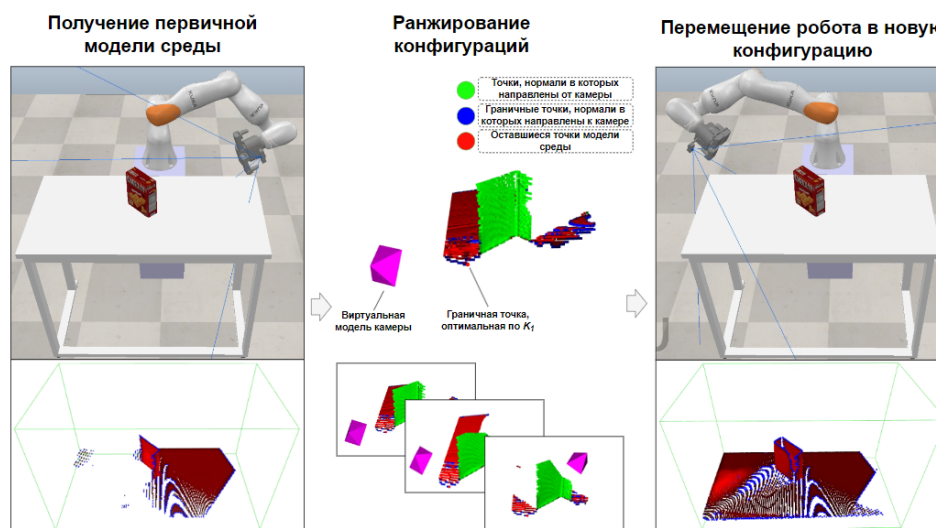


Рис. 3. – Процесс поиска новой конфигурации манипуляционного робота с учетом имеющейся модели среды

Для интеграции в TSDF-объем каждый RGB-D кадр должен иметь соответствующие атрибуты, характеризующие пространственное положение камеры и ее внутренние параметры. Для этого используется модель камеры, которая описывается матрицами внутренних и внешних параметров. Матрица внутренних параметров T_i содержит информацию, необходимую

для прямой и обратной проекций точки в трехмерном пространстве камеры в координаты пикселя на двумерном изображении: фокусные расстояния f_x и f_y по осям OX и OY изображения и координаты оптического центра c_x и c_y . Матрица внешних параметров T_e является однородной матрицей преобразований, описывающей положение и ориентацию глобальной СК $\{W\}$, относительно СК, закрепленной на камере: $T_e = \begin{bmatrix} R_w^c & T_w^c \\ 0 & 1 \end{bmatrix}$. Для работы с камерой глубины, закрепленной на конечном звене робота, в задаче построения модели внешней среды, необходимо использовать обе описанных матрицы. Калибровка положения камеры, закрепленной на ЗУМ, относительно его СК осуществлялась с помощью метода, представленного в [24]. Координаты точки препятствия в СК камеры, полученные в результате обратной проекции с матрицей T_i , могут быть преобразованы в координаты в СК $\{W\}$ путем применения преобразования $T_g^w T_c^g$ (4).

$$p_w = T_g^w T_c^g p_c, \quad (4)$$

где p_c – координаты точки, выраженные в СК камеры; p_w – координаты точки, выраженные в СК $\{W\}$.

Полученное в результате перемещения робота в новую конфигурацию RGB-D изображение требуется интегрировать в текущую модель среды. Для этого использовано TSDF-представление. TSDF-представление – способ представления трехмерных данных в виде воксельной сетки, каждый воксель которой содержит значение усеченной функции расстояния со знаком и число наблюдений.

Основными параметрами алгоритма, влияющими на процесс построения облака точек, являются: целевое расстояние между камерой и поверхностью регистрируемого облака точек l ; весовой коэффициент w_d ,

влияющий на процесс сортировки граничных точек; алгоритм завершает работу по достижении максимального числа итераций k_{max} .

Экспериментальное исследование разработанного алгоритма

Для исследования разработанного адаптивного алгоритма управления манипуляционным роботом при построении модели внешней среды был проведен ряд экспериментов, охватывающих различные сценарии применения разработанного решения. Первая серия экспериментов проводилась в среде виртуального моделирования CoppeliaSim 4.7.0. Были определены 4 тестовых модели (рис. 4), целью экспериментов было определить способность разработанного алгоритма реконструировать поверхность априорно неизвестного объекта. Усредненные по 10 экспериментам результаты приведены в табл. 1.

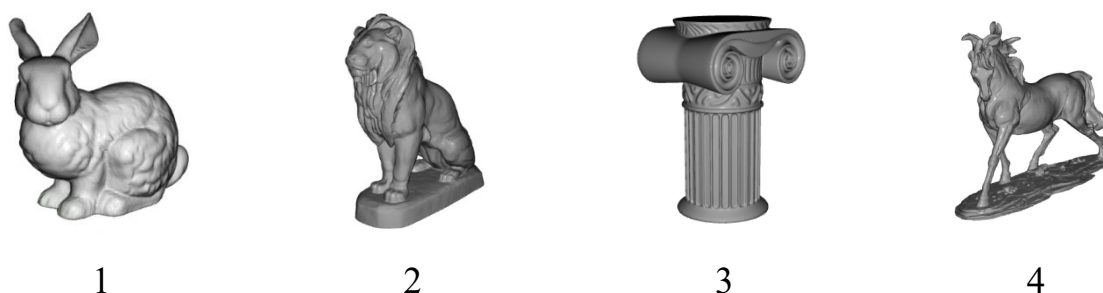


Рис. 4. – Тестовые 3D-модели, использованные в модельном эксперименте

В ходе данной серии экспериментов построение модели среды производилось с помощью 3 стратегий:

1. Управление роботом по равновероятной выборке k_{max} конфигураций, при которых RGB-D камера расположена на поверхности сферы, центр которой размещен в C ;
 2. Управление роботом по k_{max} конфигурациям робота, полученным с помощью алгоритма выборки дальней точки (Farthest Point Sampling — FPS), при которых RGB-D камера расположена на поверхности сферы, центр которой размещен в C ;
-

3. Предлагаемый подход.

Таблица 1

Результаты работы различных стратегий управления манипуляционным роботом при построении модели среды (Cov_{pcd}/AUC)

| Искажения в изображении глубины | Подход | 3D модель | | | | Среднее |
|---------------------------------|---------------------------------|-------------|-------------|-------------|-------------|-------------|
| | | 1 | 2 | 3 | 4 | |
| - | Равновероятная выборка на сфере | 0.852/0.713 | 0.812/0.631 | 0.743/0.627 | 0.775/0.648 | 0.795/0.654 |
| | FPS-выборка на сфере | 0.951/0.864 | 0.911/0.782 | 0.908/0.786 | 0.947/0.862 | 0.929/0.823 |
| | Предлагаемый подход | 0.966/0.877 | 0.945/0.820 | 0.937/0.840 | 0.970/0.879 | 0.955/0.854 |
| + | Равновероятная выборка на сфере | 0.750/0.557 | 0.692/0.530 | 0.734/0.625 | 0.627/0.489 | 0.701/0.550 |
| | FPS-выборка на сфере | 0.885/0.754 | 0.861/0.701 | 0.860/0.726 | 0.833/0.724 | 0.859/0.726 |
| | Предлагаемый подход | 0.867/0.730 | 0.878/0.706 | 0.901/0.779 | 0.873/0.741 | 0.879/0.739 |

Поскольку полная 3D-модель исследуемого объекта известна, для оценки эффективности подхода была использована величина покрытия поверхности, вычисляемая как отношение числа точек в текущей модели среды к числу точек в полном облаке точек (5) [14, 16] Также в качестве метрики для оценки скорости покрытия сцены была использована площадь под графиком (Area under curve – AUC) изменения величины покрытия $AUC = \int_1^{k_{max}} Cov_{pcd}(P_i) [14, 16]$.

$$Cov_{pcd}(P) = \frac{1}{N_{P_c}} \sum_{p_c \in P_c} U(\varepsilon - \min_{p \in P} \|p_c - p\|_2), \quad (5)$$

где P – рассматриваемое облако точек; P_c – полное облако точек; N_{P_c} – число точек в полном облаке точек; ε – пороговое значение для учета точки p в вычисление процента покрытия (использовано значение $\varepsilon = 4.33 \cdot 10^{-3}$ м при размере стороны вокселя $5 \cdot 10^{-3}$); U – функция Хевисайда.

Как видно из табл. 1, при отсутствии искажений в изображении глубины для всех моделей наибольшую эффективность по критерию

величины покрытия и площади под кривой имеет предложенный алгоритм, достигая в среднем Cov_{psd} на 2.8% выше и AUC на 3.7% выше, чем у подхода FPS-выборки. При присутствии искажений, в большинстве случаев предложенный алгоритм опережает подход FPS-выборки, достигая в среднем на 2.3% большего значения Cov_{psd} и на 1.8% большего значения AUC .

Для модельного и натурального экспериментов были сформированы 3 тестовых сцены с различной степенью загроможденности, включающих объекты различной формы. Общий вид реконструированных сцен приведен на рис. 5.

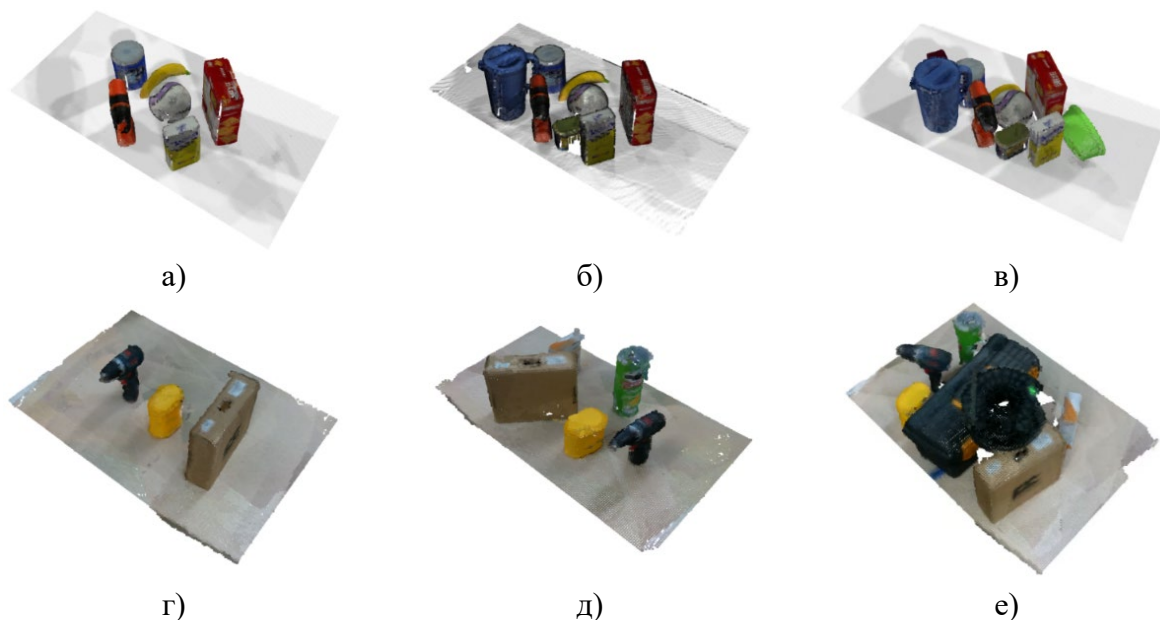


Рис. 5. Результаты работы предложенного алгоритма на тестовых настольных сценах: а)-в) в среде виртуального моделирования CoppeliaSim; г)-е) в ходе натурального эксперимента

Вторая серия экспериментов была проведена в среде CoppeliaSim и с использованием реального робототехнического комплекса (РТК). В реальном РТК используется манипуляционный робот KUKA LBR iiwa 7R800 и RGB-D камера D435. В ходе данной серии экспериментов также производилась оценка управления роботом по выборке из k_{max} конфигураций, при которых

RGB-D камера расположена в плоскости, параллельной плоскости XOY , с равной вероятностью и с помощью алгоритма FPS. Усредненные по 10 экспериментам результаты сведены в табл. 2.

Таблица 2

Результаты работы различных стратегий управления манипуляционным роботом при построении модели среды для настольных сцен

| Эксперимент | Сцена | k | Стратегия | Итерация | | | | |
|-------------|-------|-----|-------------------------------------|----------|-------|-------|-------|-------|
| | | | | 0.2k | 0.4k | 0.6k | 0.8k | k |
| Модельный | 1 | 10 | Равновероятная выборка на плоскости | 0.507 | 0.613 | 0.642 | 0.674 | 0.689 |
| | | | Равновероятная выборка на сфере | 0.537 | 0.609 | 0.634 | 0.663 | 0.680 |
| | | | FPS-выборка на плоскости | 0.457 | 0.614 | 0.713 | 0.723 | 0.730 |
| | | | FPS-выборка на сфере | 0.369 | 0.679 | 0.687 | 0.709 | 0.723 |
| | | | Предлагаемый подход | 0.565 | 0.691 | 0.730 | 0.740 | 0.741 |
| | 2 | 10 | Равновероятная выборка на плоскости | 0.470 | 0.606 | 0.694 | 0.748 | 0.771 |
| | | | Равновероятная выборка на сфере | 0.589 | 0.673 | 0.723 | 0.751 | 0.781 |
| | | | FPS-выборка на плоскости | 0.455 | 0.605 | 0.759 | 0.797 | 0.822 |
| | | | FPS-выборка на сфере | 0.322 | 0.725 | 0.762 | 0.803 | 0.835 |
| | | | Предлагаемый подход | 0.572 | 0.743 | 0.801 | 0.839 | 0.861 |
| | 3 | 20 | Равновероятная выборка на плоскости | 0.684 | 0.773 | 0.821 | 0.847 | 0.872 |
| | | | Равновероятная выборка на сфере | 0.685 | 0.776 | 0.837 | 0.857 | 0.865 |
| | | | FPS-выборка на плоскости | 0.631 | 0.839 | 0.887 | 0.906 | 0.915 |
| | | | FPS-выборка на сфере | 0.709 | 0.834 | 0.886 | 0.898 | 0.913 |
| | | | Предлагаемый подход | 0.712 | 0.864 | 0.915 | 0.937 | 0.948 |
| Натурный | 1 | 10 | Равновероятная выборка на плоскости | 0.240 | 0.339 | 0.391 | 0.450 | 0.475 |
| | | | Равновероятная выборка на сфере | 0.328 | 0.411 | 0.497 | 0.531 | 0.546 |
| | | | FPS-выборка на плоскости | 0.156 | 0.318 | 0.393 | 0.483 | 0.514 |
| | | | FPS-выборка на сфере | 0.330 | 0.532 | 0.585 | 0.588 | 0.589 |
| | | | Предлагаемый подход | 0.278 | 0.447 | 0.540 | 0.589 | 0.601 |
| | 2 | 10 | Равновероятная выборка на плоскости | 0.146 | 0.408 | 0.448 | 0.498 | 0.508 |
| | | | Равновероятная выборка на сфере | 0.319 | 0.514 | 0.570 | 0.583 | 0.601 |
| | | | FPS-выборка на плоскости | 0.158 | 0.316 | 0.416 | 0.486 | 0.514 |
| | | | FPS-выборка на сфере | 0.353 | 0.541 | 0.576 | 0.642 | 0.651 |
| | | | Предлагаемый подход | 0.279 | 0.472 | 0.632 | 0.643 | 0.659 |
| | 3 | 20 | Равновероятная выборка на плоскости | 0.299 | 0.435 | 0.527 | 0.558 | 0.587 |
| | | | Равновероятная выборка на сфере | 0.362 | 0.467 | 0.536 | 0.591 | 0.662 |
| | | | FPS-выборка на плоскости | 0.283 | 0.461 | 0.546 | 0.583 | 0.621 |
| | | | FPS-выборка на сфере | 0.474 | 0.608 | 0.665 | 0.740 | 0.758 |
| | | | Предлагаемый подход | 0.399 | 0.632 | 0.720 | 0.764 | 0.791 |

Для оценки эффективности алгоритма при работе с настольными сценами была использована оценка площади реконструированной поверхности. Реконструкция поверхности по имеющемуся облаку точек осуществлялась с помощью алгоритма поворотных шаров [25].

Как видно из табл. 2, в натурном эксперименте предложенный позволяет увеличить площадь построенной модели среды в среднем на

13.07% и на 2.54% по сравнению с управлением роботом по конфигурациям, полученным при равновероятной выборке на сфере и с помощью FPS-выборки на сфере соответственно. Временные затраты алгоритма, полученные при натурных экспериментальных исследованиях, приведены в табл. 3.

Таблица 3

Время выполнения различных этапов работы алгоритма

| № | Этап работы алгоритма | Усредненное время выполнения, с |
|---|--|---------------------------------|
| 1 | Формирование новой конфигурации робота | 0.816 |
| 2 | Планирование пути | 0.135 |
| 3 | Перемещение робота | 3.877 |
| 4 | Интеграция в TSDF | 0.083 |

Заключение

Разработанный подход позволяет осуществлять построение модели среды, имеющей сложную форму, за счет адаптивного алгоритма управления манипуляционным роботом. Для сценария реконструкции поверхности неизвестного объекта с использованием неидеальной RGB-D камеры предложенный подход достигает на 2.3% большего значения покрытия и на 1.8% большего значения площади под графиком покрытия по сравнению с управлением роботом по конфигурациям, полученным при FPS-выборке на сфере. По результатам натурального эксперимента, при построении модели настольных сцен предложенный позволяет увеличить площадь построенной модели среды в среднем на 13.07% и на 2.54%, по сравнению с управлением роботом по конфигурациям, полученным при равновероятной выборке на сфере и с помощью FPS-выборки на сфере соответственно.

Литература

1. Карелин А. Е., Кожемяченко А. В., Лемешко М. А. Обзор систем сортировки твердых коммунальных отходов // Инженерный вестник Дона, 2021, №. 7. URL: ivdon.ru/ru/magazine/archive/n7y2021/7064;
2. Wang S., Zhangli Z., Kan Z. When Transformer Meets Robotic Grasping: Exploits Context for Efficient Grasp Detection // IEEE Robotics and Automation Letters, 2023, vol. 7, № 3, pp. 8170-8177;
3. Breyer M., Chung J., Ott L., Siegwart R., Nieto J. Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter // 4th Conference on Robot Learning (CoRL 2020), 2020, pp. 1602-1611;
4. Lundell J., Corona E., Nguyen T., Verdoja F., Weinzaepfel P., Rogez G., Moreno-Noguer F., Kyrki V. Multi-FinGAN: Generative Coarse-To-Fine Sampling of Multi-Finger Grasps // 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 4495-4501;
5. Mayer V., Feng Q., Deng J., Shi Y., Chen Z., Knoll A. FFHNet: Generating Multi-Fingered Robotic Grasps for Unknown Objects in Real-time // 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 762-769;
6. Lin S., Ferreira F., Jorda M., Nambiar V., Luo J., Solowjow E., Aparicio J., Khatib O., Bohg J. UniGrasp: Learning a Unified Model to Grasp with N-Fingered Robotic Hands // IEEE Robotics and Automation Letters, 2020, vol. 5, №2, pp. 2286-2293;
7. Воронков А.Д., Диане С. А. К. Планирование захвата неизвестных объектов многопалым хватным устройством на основе нейросетевого механизма внимания // Нейрокомпьютеры: разработка, применение, 2024, т. 26, №5, с. 80-95;

8. Нгуен Т. З. Алгоритмы ускоренной обработки изображений препятствий в системе технического зрения робота // Инженерный вестник Дона, 2015, №. 1. URL: ivdon.ru/ru/magazine/archive/n1p2y2015/2855;

9. Charles R. Qi., Hao S., Kaichun M., Leonidas J. G. PointNet: deep learning on point sets for 3D classification and segmentation // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 652-660;

10. Charles R. Qi., Hao S., Kaichun M., Leonidas J. G. Pointnet++: Deep hierarchical feature learning on point sets in a metric space // Neural Information Processing Systems (4-9 December 2017, Long Beach, CA, USA), 2017, pp. 5099-5108;

11. Li Y., Rui B., Mungchao S., Wei W., Xinhan D., Baoquan C. PointCNN: Convolution on x-transformed points // Advances in Neural Information Processing Systems, 2018, pp. 820-830;

12. Mineo C., Cerniglia D., Ricotta V., Reiting. B. Autonomous 3D geometry reconstruction through robot-manipulated optical sensors. The International Journal of Advanced Manufacturing Technology, 2021, vol. 116, pp. 1895–1911;

13. Collander C., Beksi W., Huber M. Learning the Next Best View for 3D Point Clouds via Topological Features // Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 12207-12213;

14. Zeng R., Zhao W., Liu Y. PC-NBV: A Point Cloud Based Deep Network for Efficient Next Best View Planning // 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas, NV, USA), 2020, pp. 7050-7057;

15. Zaki A., Carnevale M., Schlette C., Giberti H. On the use of low-cost 3D stereo depth camera to drive robot trajectories in contact-based applications. // The International Journal of Advanced Manufacturing Technology, 2023, vol. 128, pp. 3745–3759;

16. Delmerico J., Isler S., Sabzevari R., Scaramuzza D. A comparison of volumetric information gain metrics for active 3D object reconstruction // *Autonomous Robots*, 2018, vol. 42, №2, pp. 197–208;
17. Riccardo M., Jacopo A. Contour-based next-best view planning from point cloud segmentation of unknown objects // *Autonomous Robots*, 2017, vol. 42, pp. 443-458;
18. Morrison D., Corke P., Leitner J. Multi-View Picking: Next-best-view Reaching for Improved Grasping in Clutter // *2019 International Conference on Robotics and Automation (Montreal, QC, Canada)*, 2019, pp. 8762-8768;
19. Zhang X., Wang D., Han S., Li W., Zhao B., Wang Z., Duan X., Fang C., Li X., He J. Affordance-Driven Next-Best-View Planning for Robotic Grasping // *The Conference on Robot Learning (Atlanta, GA, USA)*, 2023, vol. 229, pp. 2849-2862;
20. Luo S., Zhang M., Zhuang Y., Ma C., Li Q. A survey of path planning of industrial robots based on rapidly exploring random trees // *Frontiers in Neurorobotics*, 2023, vol. 17, pp. 1-15;
21. Katz S., Tal A., Basri R. Direct Visibility of Point Sets // *ACM Transactions on Graphics*, 2007, vol. 26, №3, pp. 1-11;
22. Şucan I., Moll M., Kavraki L. The Open Motion Planning Library // *IEEE Robotics & Automation Magazine*, 2012, vol. 19, №4, pp. 72–82;
23. Kuffner J., LaValle S. RRT-Connect: An Efficient Approach to Single-Query Path Planning // *Proceedings - IEEE International Conference on Robotics and Automation*, 2000, vol. 2, pp. 995-1001;
24. Tsai R., Lenz R. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration // *IEEE Transactions on Robotics and Automation*, 1989, vol. 5, №3, pp. 345–358;

25. Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G. The ball-pivoting algorithm for surface reconstruction // IEEE Transactions on Visualization and Computer Graphics, 1999, vol. 5, №4, pp. 349–359.

References

1. Karelin A. E., Kozhemyachenko A. V., Lemeshko M. A. Inzhenernyj vestnik Dona, 2021, №. 7. URL: ivdon.ru/ru/magazine/archive/n7y2021/7064;
2. Wang S., Zhangli Z., Kan Z. IEEE Robotics and Automation Letters, 2023, vol, 7, №3, pp. 8170-8177;
3. Breyer M., Chung J., Ott L., Siegwart R., Nieto J. 4th Conference on Robot Learning (CoRL 2020), 2020, pp. 1602-1611;
4. Lundell J., Corona E., Nguyen T., Verdoja F., Weinzaepfel P., Rogez G., Moreno-Noguer F., Kyrki V. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 4495-4501;
5. Mayer V., Feng Q., Deng J., Shi Y., Chen Z., Knoll A. 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 762-769;
6. Lin S., Ferreira F., Jorda M., Nambiar V., Luo J., Solowjow E., Aparicio J., Khatib O., Bohg J. IEEE Robotics and Automation Letters, 2020, vol. 5, №2, pp. 2286-2293;
7. Voronkov A.D., Diane S. A. K. Nejrokomп'yutery: razrabotka, primeneniye, 2024, vol. 26, № 5, pp. 80-95;
8. Nguen T. Z. Inzhenernyj vestnik Dona, 2015, №. 1. URL: ivdon.ru/ru/magazine/archive/n1p2y2015/2855;
9. Charles R. Qi., Hao S., Kaichun M., Leonidas J. G. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 652-660;
10. Charles R. Qi., Hao S., Kaichun M., Leonidas J. G. Neural Information Processing Systems (4-9 December 2017, Long Beach, CA, USA), 2017, pp. 5099-5108;



11. Li Y., Rui B., Mungchao S., Wei W., Xinhan D., Baoquan C. Advances in Neural Information Processing Systems, 2018, pp. 820-830;
 12. Mineo C., Cerniglia D., Ricotta V., Reitingen. B. The International Journal of Advanced Manufacturing Technology, 2021, vol. 116, pp. 1895–1911;
 13. Collander C., Beksi W., Huber M. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 12207-12213;
 14. Zeng R., Zhao W., Liu Y. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas, NV, USA), 2020, pp. 7050-7057;
 15. Zaki A., Carnevale M., Schlette C., Giberti H. The International Journal of Advanced Manufacturing Technology, 2023, vol. 128, pp. 3745–3759;
 16. Delmerico J., Isler S., Sabzevari R., Scaramuzza D. Autonomous Robots, 2018, vol. 42, №2, pp. 197–208;
 17. Riccardo M., Jacopo A. Autonomous Robots, 2017, vol. 42, pp. 443-458;
 18. Morrison D., Corke P., Leitner J. 2019 International Conference on Robotics and Automation (Montreal, QC, Canada), 2019, pp. 8762-8768;
 19. Zhang X., Wang D., Han S., Li W., Zhao B., Wang Z., Duan X., Fang C., Li X., He J. The Conference on Robot Learning (Atlanta, GA, USA), 2023, vol. 229, pp. 2849-2862;
 20. Luo S., Zhang M., Zhuang Y., Ma C., Li Q. Frontiers in Neurorobotics, 2023, vol. 17, pp. 1-15;
 21. Katz S., Tal A., Basri R. ACM Transactions on Graphics, 2007, vol. 26, №3, pp. 1-11;
 22. Sucas I., Moll M., Kavraki L. IEEE Robotics & Automation Magazine, 2012, vol. 19, №4, pp. 72–82;
 23. Kuffner J., LaValle S. IEEE International Conference on Robotics and Automation, 2000, vol. 2, pp. 995-1001;
-



24. Tsai R., Lenz R. IEEE Transactions on Robotics and Automation, 1989, vol. 5, №3, pp. 345–358;

25. Bernardini F., Mittleman J., Rushmeier H., Silva C., Taubin G. IEEE Transactions on Visualization and Computer Graphics, 1999, vol. 5, №4, pp. 349–359.

Дата поступления: 12.02.2025

Дата публикации: 25.04 2025