Информационные процессы, Том 20, № 2, 2020, стр. 150–160

МАТЕМАТИЧЕСКИЕ МОДЕЛИ, ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

Об обучении с подкреплением в последовательных задачах планирования ракурсов съемки априори произвольных 3D-форм[1]

С.Г.Потапова[*], А.В.Артемов[**], С.В.Свиридов[***], Д.А.Мусаткина[**], Д.Н.Зорин[,‡], Е.В.Бурнаев[**,****]**

[*] *Московский физико-технический институт, Долгопрудный, Россия*
[**] *Сколковский институт науки и технологий, Москва, Россия*
[***] *ООО Цифра, Москва, Россия*
[****] *Институт проблем передачи информации, Российская академия наук*
[‡] *Университет Нью-Йорка, Нью-Йорк, США*
e-mail: e.burnaev@skoltech.ru

Поступила в редколлегию 01.06.2020

Аннотация—Построение 3D-моделей реальных объектов по сенсорным данным является фундаментальной задачей компьютерного зрения. Центральный фактор для эффективности решения этой задачи (выражаемой качеством 3D-модели и скоростью ее построения) – выбор ракурсов съемки объекта (3D-поз сенсоров) при получении изображений его поверхности. Последняя проблема остается открытой в области обработки 3D-геометрии и, как правило, решается переборными или жадными алгоритмами. В настоящей работе предложен алгоритм поиска оптимальных 3D-поз сенсоров на основе метода машинного обучения с подкреплением. Показано, что алгоритм обучения с подкреплением DDQN превосходит базовые подходы в рассматриваемой задачи, требуя меньшее количество поз при близком уровне погрешности 3D-реконструкции. Обучение агента и проведение экспериментов реализовано на основе среды для реконструкции 3D-форм, включающей возможность планирования поз камеры, моделирование процесса получения 3D-изображений, реконструкцию 3D-форм в виде треугольных сеток.

КЛЮЧЕВЫЕ СЛОВА: 3D модель, планирование ракурсов съемки, карта глубины, CAD модель, обучение с подкрепление, треугольная сетка.

1. ВВЕДЕНИЕ

Широкий спектр приложений в области цифрового производства и контроля качества, реверс-инжиниринга, сохранения культурного наследия, археологии, медицины, виртуального туризма, развлечений и дополненной реальности опирается на возможности построения цифровых трехмерных моделей (3D-моделей) реальных объектов и сцен [1–3]. Одним из наиболее привлекательных средств такой 3D-реконструкции служит моделирование 3D-геометрии объекта по множеству совмещенных неполных изображений глубины фрагментов его поверхности, получаемых 3D-сканерами (например, времяпролетными камерами или с использованием структурированной подсветки), и предпочтительным результатом в ряде случаев является трехмерная полигональная сетка. Полнота и качество 3D-модели напрямую зависят от набора используемых изображений: например, пропущенные фрагменты поверхности объекта при

[1] Работа выполнена при поддержке Министерства образования и науки Российской Федерации, грант No. 14.615.21.0004, код гранта: RFMEFI61518X0004.

сканировании, вообще говоря, не могут быть реконструированы и приводят к неудовлетворительной 3D-геометрии (ложным отверстиям, размытию формы и т.п.). Таким образом, одной из ключевых задач при 3D-реконструкции геометрии реальных объектов является выбор оптимального множества измерений (сканов) глубины. Если механизм сканирования остается неизменным на протяжении всего эксперимента, последняя задача сводится к выбору оптимального множества позиций и ориентаций (3D-поз) сканера (ракурсов съемки).

В зависимости от априорных предположений о геометрии и топологии реконструируемой 3D-формы проблема выбора множества поз сканера допускает одновременную либо последовательную формулировку. Например, если CAD-модель реконструируемого объекта точно известна заранее (например, при инспекции произведенных деталей), задача сводится к расчету 3D-поз сканера, обеспечивающих достаточное покрытие объекта, и выбору последующей оптимальной последовательности обхода этих поз. В литературе класс задач такого типа, как правило, называется задачами “планирования ракурсов съемки” (view planning problem, VPP-задача) [4], и соответствует NP-трудным задачам об оптимальном покрытии множества (set-covering optimization problem, SCOP-задача) [5].

Противоположностью такой постановки является задача “последовательного планирования ракурсов съемки” (next best view planning, NBV-задача) [6–11], в которой предполагается, что реконструируемый объект может иметь априори произвольную геометрию; как следствие, выбор 3D-позы сканера требуется осуществлять в режиме реального времени, интегрируя доступную на каждый момент времени информацию. Среди подходов к решению NBV-задач предлагались методы синтеза на основе анализа информации о поверхности [6–8], методы на основе поиска 3D-поз с помощью оптимизации функции полезности [9–11], и, в последние годы, методы на основе обучения [12, 13].

Общими требованиями при решении задач планирования съемки являются минимизация количества сканирований либо перемещений сенсора, так как они снижают вычислительную эффективность реконструкции и являются дорогостоящими операциями для робота.

Трудности, свойственные задачам последовательного планирования ракурсов съемки, являются общими для ряда проблем в контексте компьютерных наук, робототехники и искусственного интеллекта. Так, последовательное принятие решений в условиях неопределенности в частично наблюдаемых среде и с необходимостью выбора из комбинаторного количества вариантов характерно для задач построения игроков в настольные [14] и компьютерные [15] игры, задач исследования операций [16], задач построения рекомендательных систем [17] и других. В последние годы, особенно с развитием глубокого обучения, многообещающие результаты в описанных задачах такого рода демонстрируют алгоритмы обучения с подкреплением (RL-алгоритмы); кроме того, последние малоизучены в контексте планирования ракурсов съемки и автоматической 3D-реконструкции. Авторам известна единственная работа [4], в которой ряд RL-алгоритмов используется для решения VPP-задачи.

В настоящей работе исследована проблема последовательного планирования ракурсов съемки методом обучения с подкреплением. Для этого NBV-задача сформулирована как процесс последовательного принятия решений, в котором агент последовательно сканирует реконструируемый объект из разных точек пространства с целью получения максимального качества реконструкции, при этом минимизируя число сканирований. Предложен RL-алгоритм решения задачи на основе глубинного обучения с подкреплением, создана его эффективная программная реализация. Количественное исследование эффективности созданного подхода проведено в ходе вычислительных экспериментов в симуляционной среде.

Математическое описание процесса последовательного принятия решений приведено в разделе 2.1, описание вычислительного метода решения задачи – в разделе 2.2, а созданная экспериментальная среда – в разделе 2.3. Раздел 3 описывает постановку и результаты вычис-

лительных экспериментов на модельных 3D-данных с привлечением реализованного метода и базовых подходов.

2. ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ В NBV-ЗАДАЧАХ

2.1. NBV-задача как марковский процесс принятия решений

В данной работе NBV-задача формулируется в виде *марковского процесса принятия решений* (Markov decision process, MDP), в котором агент взаимодействует со средой. При этом, т. к. агент не имеет доступа к состоянию среды (истинной 3D-модели), для формализуем задачу в виде эпизодического *частично наблюдаемого марковского процесса принятия решений* с дискретным временем (partially observable MDP, POMDP). Формально POMDP можно описать следующим кортежем $M = \langle S, A, R, T, O, E, \gamma, H\rangle$, где

- S — *пространство состояний* $\mathbf{s}_t \in S$, которое может быть как дискретным, так и непрерывным.
- A — *пространство действий* $\mathbf{a}_t \in A$, которое также может быть как дискретным, так и непрерывным.
- R — *функция вознаграждения*, представляющая собой отображение из пространств состояний и действий во множество вещественных чисел $R : S \times A \to \mathbb{R}$.
- T — представляет собой отображение $T(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ и описывает динамику POMDP.
- O — пространство наблюдений $\mathbf{o}_t \in O$.
- E — оператор эмиссии, представляющий в POMDP отображение вида $E(\mathbf{o}_{t+1}|\mathbf{s}_{t+1}, \mathbf{a}_t)$.
- γ — фактор дисконтирования, $0 < \gamma \leq 1$.
- H — горизонт, определяющий длительность эпизода в эпизодическом POMDP, при этом эпизод представляет собой последовательность $\{s_t, a_t, s_{t+1}, a_{t+1}, \ldots\}_{t=0}^{H}$.

В контексте решения NBV-задачи основные элементы POMDP задаются следующим образом.

- Пространство состояний S состоит из одного элемента, является непрерывным и представляет собой истинную 3D-модель объекта.
- Пространство действий A задается дискретным множеством размера N равномерно распределенных на сфере точек. Каждая точка представляет собой положение виртуального 3D-сканера, ориентированного в центр масс объекта, вокруг которого описана сфера. В момент времени t действие $\mathbf{a}_t$ размещает виртуальный 3D-сенсор в одной из точек на сфере и производит сканирование объекта из этой точки, получая наблюдение $\mathbf{o}_t$. При этом запрещается совершать уже совершенные действия, т.е. производить сканирование объекта из уже посещенных точек.
- Функция вознаграждения r в момент времени t вычисляется согласно

$$r_t = w_1 * r_t^{\mathrm{Dat}} - w_2 * r_t^{\mathrm{Rec}} - w_3 * r_t^{\mathrm{UR}} + w_4 * r_t^{\mathrm{Act}}, \tag{1}$$

  в котором слагаемые выражают, соответственно:
  - r_t^{Dat} — приращение доли площади поверхности объекта, покрытой измерениями,
  - r_t^{Rec} — погрешность реконструкции объекта,
  - r_t^{UR} — доля зоны неопределенности,
  - r_t^{Act} — штраф за совершение действия,
  - $w_1, w_2, w_3, w_4 \in \mathbb{R}^+$ — веса соответствующих компонентов.

  Функция вознаграждения задается таким образом, чтобы среда давала большие награды за большее покрытие 3D-модели и лучшее качество 3D-реконструкции, а также штрафовала агента за количество сканирований. Детали расчета данных компонентов функции вознаграждения приведены в п. 2.3.

– Динамика POMDP T является детерминированной и определяется средой, устройство которой описано в п. 2.3.
– Пространство наблюдений O дискретно и представляет собой объемную 3D-сетку, каждая ячейка которой может принимать одно из следующих значений: “поверхность”, “пустое”, “неопределенность”. Объемная сетка строится в результате сканирования объекта, детали которой приведены в п. 2.3.
– Оператор эмиссии E определяет наблюдение $\mathbf{o}_{t+1}$, которое агент получает при совершении действия $\mathbf{a}_t$, т. е. при операции сканирования объекта из определенной точки на сфере, описанной вокруг объекта.
– Фактор дисконтирования γ является определяется условиями эксперимента, его значение задано в п. 3.
– Длительность H каждого эпизода определяется числом шагов, необходимых для первого достижения доли покрытия α_H граней объекта и уровня ошибки реконструкции β_H. Описание процедур расчета покрытия и реконструкции даны в п. 2.3.

Таким образом, формально определив POMDP, мы можем описать процесс реконструкции объекта следующим образом. В каждый момент времени t среда находится в состоянии $\mathbf{s}_t$, а агент имеет наблюдение о состоянии $\mathbf{o}_t$; агент выбирает действие $\mathbf{a}_t$, соответствующее сканированию объекта с выбранной точки на сфере, описанной вокруг объекта. После этого среда переходит в состояние $\mathbf{s}_{t+1}$ и, используя оператор эмиссии E, возвращает агенту следующее наблюдение $\mathbf{o}_{t+1}$ в виде объемной сетки, полученной в результате сканирования, а также вознаграждение $\mathbf{r}_t$. Агент продолжает осуществлять действия, получать наблюдения и вознаграждения до конца горизонта H, т. е. до достижения уровня покрытия поверхности α_H и ошибки реконструкции β_H. Стоит заметить, что в представленном POMDP состояние среды не зависит от момента времени t и представляется собой истинную 3D-модель объекта.

Целью агента в данном POMDP является поиск стохастической политики $\pi(\mathbf{a}_t|\mathbf{o}_t)$, которая оптимизирует следующий критерий:

$$\max \mathop{\mathbb{E}}_{a\sim\pi} \big[\sum_{t=0}^{H} \gamma^t r_t\big]. \tag{2}$$

Такая политика является оптимальной и обозначается π^*. Нахождение такой оптимальной политики в описанном POMDP эквивалентно решению задачи реконструкции 3D-объекта с наименьшей погрешностью 3D-формы за наименьшее количество операций сканирования.

2.2. Метод глубокого обучения с подкреплением для решения NBV-задачи

Для оптимизации функционала (2) в настоящей работе используется глубокая Q-сеть [18], представляющая из себя сверточную нейронную сеть, принимающую на вход тензор-наблюдение из O, и возвращающую q-значения для всех возможных 3D-поз сенсора. Оптимальная 3D-поза выбирается согласно ϵ-жадной стратегии среди еще не посещенных с максимальным q-значением.

Q-сеть с весами θ может быть обучена путем минимизации функции потерь $L_t(\theta_t)$, которая изменяется при каждой итерации t,

$$L_t(\theta_t) = \mathbb{E}_{s\sim S, a\sim A}[(y_t - Q(s_t, a_t; \theta_t))^2]$$

где $y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta_{t-1})$ — целевое значение для итерации t. Дифференцируя функцию потерь по весам, мы получаем следующий градиент

$$\nabla_{\theta_t} L_t(\theta_t) = \mathbb{E}_{s\sim S, a\sim A}\left[(y_t - Q(s, a; \theta_t))\nabla_{\theta_t} Q(s, a; \theta_t)\right]. \tag{3}$$

Параметры предыдущей итерации $t-1$ сохраняются фиксированными при оптимизации функции потерь $L_t(\theta_t)$. Проблемой такого подхода является то, что целевые значения зависят от веса сети и меняются с каждой итерацией обучения; это противоречит принципам обучения с учителем, где целевые значения устанавливаются до начала обучения. При таких условиях сложно обеспечить хорошую сходимость при обучении нейронной сети. Для решения этой проблемы часто используют дополнительную "целевую" нейронную сеть, веса которой обновляются (копируются из основной сети) раз в C итераций. В то же время эта сеть используется для генерации целевых значений $y_t = r_t + \gamma Q(s_{t+1}, \arg\max_{a'} Q(s_{t+1}, a', \theta_t), \widehat{\theta})$, где $\widehat{\theta}$ — параметры целевой сети. Данный алгоритм называется Double Deep Q-network [19].

Распространенной практикой обучения RL-агентов является использование буфера памяти (experience-replay buffer), хранящего в себе кортежи вида (наблюдение, действие, награда, следующее действие). Последовательности наблюдений-действий часто взаимосвязаны во времени, и случайное сэмплирование из буфера памяти помогает решать проблему корреляции обучающих данных. Для добавления временного контекста для лучшей сходимости обучения мы склеиваем n_{fsc} последних подряд идущих наблюдений в одно (аналогично [18]).

Алгоритм процедуры обучения DDQN с использованием ϵ-жадной стратегии и Experience-Replay Buffer:

Инициализировать буфер D размера K;
Инициализировать Q-функцию случайными значениями;
for episode $= 1, M$ **do**
  Инициализировать случайное a_1 и получить в среде для него s_1;
  for $t = 1, T$ **do**
    if $t \bmod C = 0$ **then**:
      $\hat{\theta} \leftarrow \theta_t$
    end if
    С вероятностью ϵ случайное значение a_t;
    иначе выбрать $a_t = \arg\max_a Q(s_t, a; \theta)$;
    В среде совершить действие a_t и получить вознаграждение r_t и воксель-грид s_{t+1};
    Принять $s_{t+1} = s_t$;
    Сохранить кортеж (s_t, a_t, r_t, s_{t+1}) в D;
    $(s_j, a_j, r_j, s_{j+1}) \sim D$;
    Принять $y_j = \begin{cases} r_j & \text{для конечного } s_{j+1} \\ r_j + \gamma \max_{a'} Q(s_{j+1}, \arg\max_{a'} Q(s_{j+1}, a', \widehat{\theta}); \theta_j) & \text{иначе} \end{cases}$
    Выполнить шаг градиентного спуска для ошибки $(y_j - Q(s_j, a_j; \theta_j))^2$ согласно уравнению (3);
  end for
end for

2.3. Вычислительная среда для обучения RL-агента

Для обучение агента и проведения экспериментов реализована программная среда для реконструкции 3D-форм, совместимая с программным интерфейсом OpenAI Gym [20] и включает определения пространства действий A, состояний S и наблюдений O; инициализацию и обновление среды; логику, совершаемая на каждом шаге итерации до выполнения критерия останова H; вычисление функция вознаграждения r; иллюстрацию наблюдений.

Основная среда включает в себя следующую логику: 1) подгрузка случайной 3D-формы и ее нормализация; 2) равномерная генерация 3D-поз сенсора на сфере с центром в центре масс объекта; 3) случайная генерация первой 3D-позы сенсора; 4) сканирование объекта с заданной позы; 5) определение площади покрытия объекта; 6) реконструкция объекта из облака

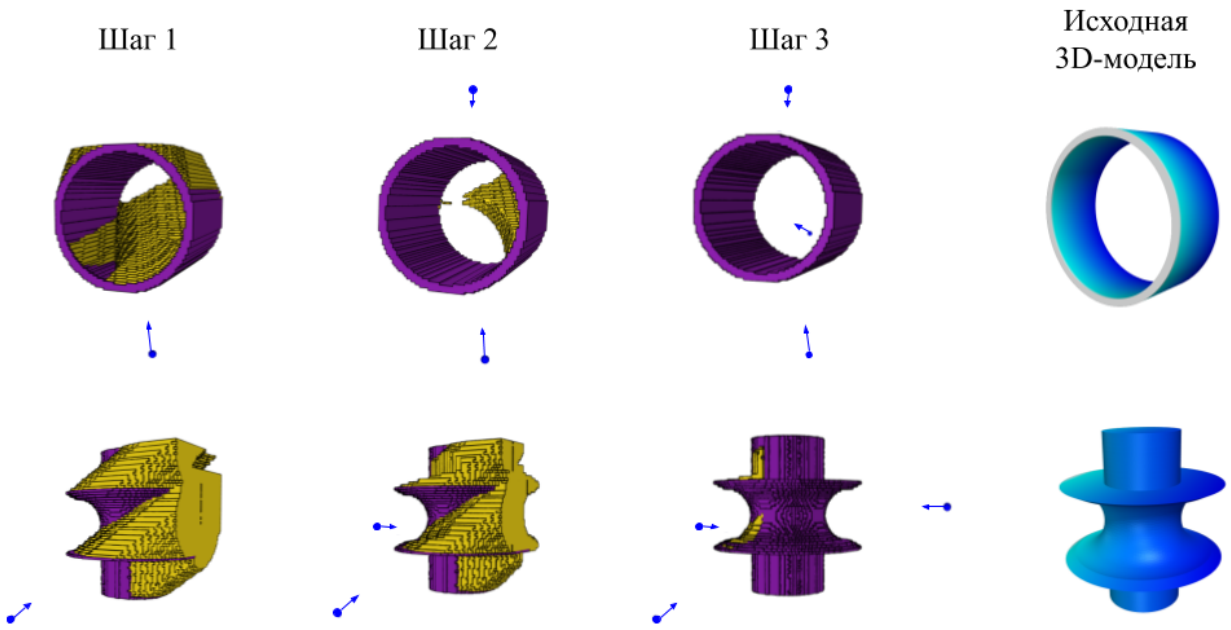

Рис. 1. Примеры уточнения объемных сеток для двух 3D-объектов и подряд совершенных сканирований с разных позиций сенсора. Слева направо: показана последовательность действий RL-агента и исходные сетки.

точек; 7) сравнение реконструированного объекта с исходным; 8) иллюстрации исходного и реконструированного объекта, сгенерированных 3D-поз сенсора и наблюдений. Под 3D-позой сенсора подразумевается центр виртуального сенсора, ориентированного в центр масс объекта, с двумя степенями свободы — углами Эйлера ϕ и θ, задающими его положение на сфере. Моделью формирования изображений глубины служит рейкастинг, общепринятый в области компьютерной графики метод рендеринга поверхностей [21]. Реконструкция полигональной 3D-сетки по облаку точек производится методом Пуассона [3]. Сравнение 3D-сеток — вычислением расстояния Хаусдорфа между ними. Равномерное сэмплирование точек на сферу — методом решетки Фибоначчи [22]. Площадь покрытия объекта определяется числом полигонов 3D-сетки, каждый из которых был пересечен хотя бы одним лучем при рейкастинге.

Наблюдения, получаемые после операции рейкастинга, имеют вид неструктурированных облаков точек, которые далее преобразуются в объемную сетку фиксированного разрешения, в которой каждому объему соответствует одно из значений “поверхность”, “пустое пространство”, “неопределенность”. Последовательность наблюдений из одного эпизода преобразуется в комбинированную объемную сетку объединением объемов поверхности и пересечением объемов неопределенности (см. рис. 1). Таким образом, на каждой итерации t агент видит накопленные наблюдения с уже посещенных позиций сенсора.

В программной среде дополнительно реализован расчет компонентов функции вознаграждения. На каждом шаге t алгоритма выполняется упрощенная реконструкция полигональной 3D-сетки из равномерно фильтрованного облака точек скомбинированных наблюдений, вычисляется r_t^{Rec}; вычисляется r_t^{Dat} как приращение доли покрытой площади поверхности объекта; вычисляется r_t^{UR} как доля объемов, соответствующих “неопределенности”, относительно всего размера объемной сетки.

3. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

3.1. Постановка экспериментов

Для обучения и валидации RL-агента были выбраны 617 и 43 3D CAD-модели из коллекции ABC [23] соответственно, для которых число сплайн-поверхностей не выше 20, обладающие достаточно простой топологией (без узких длинных впадин, отверстий и т.п.). Примеры 3D-моделей из обучающей выборки изображены на Рис. 2.

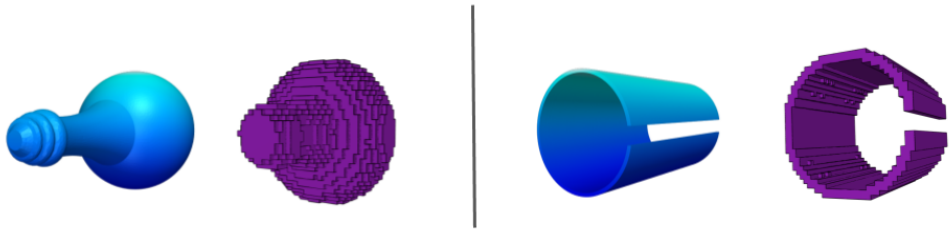

Рис. 2. Примеры CAD-моделей и соответствующих им объемных сеток из коллекции ABC, используемых для обучения и тестирования RL-агента.

Все RL-агенты обучались в течение 250 000 эпох. В качестве DDQN была использована нейронная сеть с 5 сверточными слоями и нелинейностью ReLU. В качестве параметров среды были использованы: $n_{\mathrm{C}} = 100$ равномерно распределенных на сфере 3D-позиций сенсора; размер моделируемого изображения глубины $w = h = 1024$; разрешение объемной сетки 64^3. Заданы следующие гиперпараметры алгоритма обучения: размер памяти предыдущих состояний $n_{\mathrm{fsc}} = 4$; фактор дисконтирования $\gamma = 0.99$; размер буфера памяти $n_{\mathrm{erb}} = 10^5$. Рассматривался ряд конфигураций функции вознаграждения (1), отличающихся заданием весов w_1, w_2, w_3, w_4. Горизонт H определялся достижением доли покрытия поверхности объекта $\alpha_H = 0.95$ и погрешности восстановления сетки $\beta_H = 0.1$.

Сравнительное исследование проводилось против ряда базовых алгоритмов планирования ракурсов съемки: “Полный (n_{I})”, использующий весь доступный бюджет из n_{I} снимков; “Случайный”, выбирающий ракурс съемки равновероятно среди доступных; “Наиболее удаленная поза”, где следующая 3D-поза сенсора выбирается как наиболее удаленная от текущей без повторений; “Наивный жадный”, на каждом шаге сканирующий объект со всех доступных ракурсов, выбирая 3D-позу по принципу максимального прироста площади покрытия; “Уменьшение неопределенности”, также сканирующий объект со всех доступных ракурсов, выбирая 3D-позу по принципу максимального уменьшения количества вокселей “неопределенности” в объемной сетке. Заметим, что последние два алгоритма используют все доступные изображения для принятия решения о планировании следующего ракурса, и их результаты приведены из соображений полноты сравнения.

В качестве количественных показателей эффективности для сравнительного исследования выбрана погрешность 3D-реконструкции, выражаемая расстоянием Хаусдорфа d_{H} между исходной и реконструированной 3D-полигональными сетками и количество 3D-поз сенсора n_{I}, необходимое для достижения горизонта H.

3.2. Результаты экспериментальных исследований

В Табл. 1 представлены результаты сравнения эффективности предложенного в данной работе метода обучения с подкреплением и базовых алгоритмов. Результаты демонстрируют, что обученный RL-агент не совершает сканирование со всех доступных $n_{\mathrm{C}} = 100$ ракурсов, а

Таблица 1. Результаты количественного исследования методов последовательного планирования съемки на коллекции ABC. Алгоритмы “Наивный жадный” и “Уменьшение неопределенности”, использующие полный набор доступных 3D-поз сенсора, помечены знаком *.

Метод	Число снимков n_{I}	Погрешность d_{H}
Полный (5)	5.0	0.149
Полный (100)	100.0	0.014
Случайный	9.3	0.411
Наиболее удаленная поза	10.2	0.172
* Наивный жадный	3.3	0.083
* Уменьшение неопределенности	8.0	0.311
RL-агент (предложенный метод)	4.8	0.079

Таблица 2. Исследование вклада компонент функции вознаграждения (1).

Конфигурация	Число снимков n_{I}	Погрешность d_{H}
RL-агент (предложенный метод)	4.8	0.079
Без учета зоны неопределенности	5.2	0.087
Без учета погрешности реконструкции	5.6	0.091
Без учета зоны неопределенности и погрешности реконструкции	5.1	0.102
Без учета покрытой поверхности и зоны неопределенности	5.4	0.077
Только штраф за совершение действий	5.8	0.113

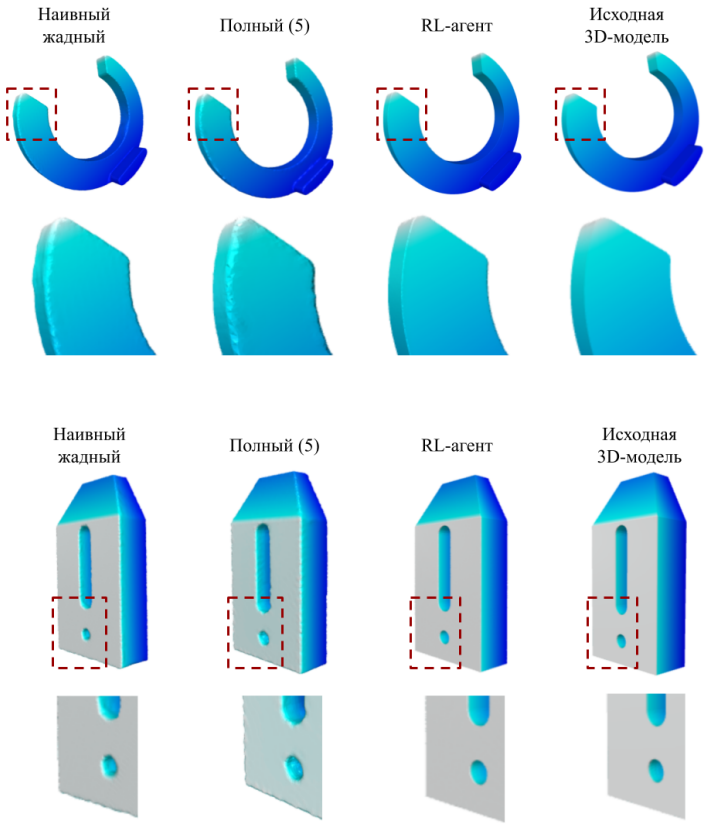

Рис. 3. Результаты трехмерной реконструкции при использовании жадного, полного (5) и предложенного методов.

использует лишь около 5 снимков. Предложенный метод добивается близких к оптимальным в смысле "Наивного жадного" значений количества 3D-поз камеры, превосходя этот алгоритм по качеству реконструкции.

В Табл. 2 приведено сравнение RL-агентов, обученных с различной структурой функции вознаграждения (1), задаваемой весами $w_1, \ldots, w_4$, значения которых приняты равными нулю или единице для анализа значимости компонент. Результаты позволяют заключить, что компонент r^{Rec} вносит вклад в качество реконструкции; компоненты r^{Dat} и r^{UR} необходимы для лучшего покрытия поверхности объекта; штраф за совершение действий r^{Act} заставляет RL-агента минимизировать количество 3D-поз камеры. Оптимальная функция вознаграждения состоит из всех перечисленных слагаемых, при ее использовании агент выучивает логику максимизации качества реконструкции за наименьшее число шагов. Результаты восстановления полигональных 3D-сеток приведены на Рис. 3 и демонстрируют, что наилучшее визуальное качество восстановления 3D-сетки достигается разработанным RL-алгоритмом.

4. ЗАКЛЮЧЕНИЕ

В работе разработан и исследован алгоритм последовательного планирования ракурсов съемки 3D-форм на основе обучения с подкреплением. Экспериментальные исследования разработанного метода позволяют сделать вывод о его перспективности в контексте автоматической 3D-реконструкции. Возможными направлениями дальнейшей работы являются (1) переход к непрерывному пространству действий и (2) увеличение числа степеней свободы 3D-сенсора, (3) исследование компонент функции вознаграждения, отражающих расстояние между последовательными 3D-позами, а также (4) обобщение разработанного подхода на объекты с более сложной топологией 3D-формы.

СПИСОК ЛИТЕРАТУРЫ

1. Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000.

2. Fabio Remondino. Heritage recording and 3d modeling with photogrammetry and 3d scanning. *Remote sensing*, 3(6):1104–1138, 2011.

3. Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP'06, pages 61–70. Eurographics Association, 2006.

4. M. D. Kaba, M. G. Uzunbas, and S. N. Lim. A reinforcement learning approach to the view planning problem. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5094–5102, 2017.

5. Glenn H Tarbox and Susan N Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61(1):84–111, 1995.

6. S. Y. Chen and Y. F. Li. Vision sensor planning for 3-d model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):894–904, 2005.

7. J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–433, 1993.

8. William R. Scott, Gerhard Roth, and Jean-Francois Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, 2003.

9. B. Yamauchi. A frontier-based approach for autonomous exploration. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997.

10. A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. Receding horizon "next-best-view" planner for 3d exploration. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, 2016.

11. S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484, 2016.

12. Miguel Mendoza, J. Irving Vasquez-Gomez, Hind Taud, L. Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.

13. Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

14. David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint 1712.01815*, 2017.

15. Adria Puigdomenech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. *arXiv preprint 2003.13350*, 2020.

16. Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! *International Conference on Learning Representations*, 2019.

17. Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for page-wise recommendations. *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

18. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *NIPS Deep Learning Workshop*, 2013.

19. Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

20. Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint 1606.01540*, 2016.

21. Ingo Wald, Sven Woop, Carsten Benthin, Gregory S Johnson, and Manfred Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):1–8, 2014.

22. Martin Roberts. Evenly distributing points on a sphere. *Extreme Learning Blog*, 2018.

23. Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9601–9611, 2019.

Next Best View Planning via Reinforcement Learning for Scanning of arbitrary 3D Shapes

Potapova S.G., **Artemov A.V.**, **Sviridov S.V.**, **Musatkina D.A.**, **Zorin D.N.**, **Burnaev E.V.**

Reconstructing 3D objects from scanned sensor measurements is a fundamental task in computer vision. A central factor for the effectiveness of 3D reconstruction is the choice of sensor views while performing scanning. The latter remains an open problem in the 3D geometry processing area, known as the next-best-view planning, and is commonly approached by combinatorial or greedy methods. In this work, we propose

a reinforcement learning-based approach to sequential next-best-view planning. The method is implemented based on the gym environment including 3D reconstruction, next-best-scan planning, and image acquisition features. We demonstrate this method to outperform the baselines in terms of the number of required scans and the obtained 3D mesh reconstruction accuracy.

KEYWORDS: 3D model, next best view, depth map, CAD model, reinforcement learning, mesh.