Параллельный алгоритм для аппроксимации рабочей области робота

А.Ю. Горчаков, А.Д. Игнатов, Д.И. Малышев, М.А. Посыпкин

***Аннотация*— Рабочей областью робота называется множество положений, которые может принимать его рабочий инструмент. Знание рабочей области необходимо при проектировании роботов, их размещении, оценке их функциональных возможностей, прокладке траектории движения робота. К настоящему времени разработано много методов для определения рабочей области. Следует заметить, что наибольшим потенциалом обладают детерминированные методы, позволяющие в автоматическом режиме получать аппроксимации с заданной точностью. Известным недостатком детерминированных методов является их высокая вычислительная сложность, которая препятствует эффективному широкому применению этих методов на практике. В работе предлагается параллельный алгоритм построения аппроксимации рабочей области с гарантированной точностью. Для создания многопоточного приложения применен пакет OpenMP. Разработана оригинальная техника, позволяющая равномерно распределять нагрузку по потокам без явной балансировки. Приводятся результаты экспериментов, показывающие высокую эффективность распараллеливания для многоядерных систем. Не менее важной задачей является визуализация построенных аппроксимаций. Рабочая область зачастую имеет сложную структуру с внутренними полостями. В данной работе предлагаются подходы для эффективной визуализации рабочей области робота, основанные на технологии дополненной реальности, позволяющие не только изучать строение объекта, но и помещать его в требуемый контекст.**

***Ключевые слова*—рабочая область роботов, аппроксимация с гарантированной точностью, параллельные алгоритмы, визуализация.**

I. Введение

Рабочей областью робота называется множество положений, которые может принимать его рабочий инструмент. Знание рабочей области необходимо при проектировании роботов, их размещении, оценке их функциональных возможностей. Рабочая область является основой для прокладки траектории движения робота [1]. В настоящее время разработано достаточно много подходов для определения рабочей области. Рассмотрим некоторые из них.

Для многих роботов рабочая область может быть найдена аналитически. Аналитическое представление компактно и точно определяет искомое множество. Вместе с тем, такие подходы обладают рядом недостатков. Во-первых, они требуют существенных усилий со стороны исследователя. Во-вторых, способ их применения не может быть унифицирован, отличаясь не только для различных роботов, но и для одного робота при различных параметрах. Это связано с тем, что при различных размерах конструктивных элементов робота форма рабочей области, в том числе, число компонентов связности, может существенно отличаться.

Другой класс методов решения подобных задач основан на детерминированных подходах, заимствованных из методов глобальной оптимизации и интервального анализа [2,3]. Рабочая область может быть задана системой уравнений и/или неравенств при интервальных ограничениях на параметры. Исходный параллелепипед итеративно разбивается на параллелепипеды меньшего размера. Для каждого параллелепипеда проверяется выполнимость уравнений и неравенств системы. Если показана несовместность хотя бы одного из условий, то параллелепипед исключается из дальнейшего рассмотрения.

Безусловным достоинством детерминированных методов является их универсальность и гарантированная точность получаемых аппроксимаций. Получаемое разбиение в виде совокупности непересекающихся параллелепипедов позволяет легко вычислять объем аппроксимации и решать задачу прокладки траектории. Однако, как показывают результаты экспериментов, эти методы обладают высокой трудоемкостью. Для получения высокоточных аппроксимаций рабочих областей планарных параллельных роботов могут потребоваться часы вычислений.

Одним из возможных направлений ускорения построения аппроксимации с помощью детерминированных методов является применение технологий параллельных и распределенных вычислений. Поскольку для построения аппроксимаций применяется схема ветвей и границ, то могут быть задействованы подходы, разработанные для распараллеливания метода ветвей и границ [4-7]. В работе предлагается параллельный алгоритм построения аппроксимаций, развивающий идеи, изложенные в [7] для задач глобальной оптимизации. Приводятся результаты экспериментов, показывающие высокую

Статья загружена 09.12.2018. Работа выполнена при поддержке проекта РНФ № 16-19-00148.
А. Ю. Горчаков, старший научный сотрудник ВЦ РАН ФИЦ ИУ РАН. (e-mail: andrgor12@gmail.com).
А. Д. Игнатов, магистр ВМК МГУ им. М.В. Ломоносова (e-mail: rayignatov@outlook.com).
Д. И. Малышев, аспирант БГТУ им.В.Г.Шухова. (e-mail: Malyshev.d.i@yandex.ru).
М. А. Посыпкин, главный научный сотрудник ВЦ РАН ФИЦ ИУ РАН. (e-mail: mposypkin@gmail.com).

эффективность распараллеливания для многоядерных систем с общей памятью.

Также в статье затрагивается важная и актуальная проблема визуализации полученной аппроксимации рабочей области в трехмерном случае. Разработано программное обеспечение, позволяющее эффективно визуализировать рабочую область робота на мобильном устройстве с применением технологий «дополненной реальности».

II. Алгоритм построения аппроксимации множества решений системы неравенств

Приведем описание последовательного алгоритма. Требуется найти множество X, задаваемое набором неравенств и интервальных ограничений:

$$g_j(x) \le 0, j = 1, \dots, m,$$
$$x_i \in [a_i, b_i], i = 1, \dots, n. \quad (1)$$

Пусть на некотором параллелепипеде B могут быть получены оценки $m_j(B), M_j(B)$ для функций, стоящих в левых частях функциональных неравенств:

$$m_j(B) \le g_j(x) \le M_j(B) \text{ для всех } x \in B.$$

Существую различные способы получения оценок. Часто используются интервальные оценки, которые получаются применением арифметических операций интервальной арифметики. Однако, такие оценки не всегда дают хороший результат.

Мы в дальнейшем будем использовать Липшицевы оценки. Функция $f(x): R^n \to R$ удовлетворяет условию Липшица на множестве $S \subseteq R^n$, если

$$|f(x) - f(y)| \le L||x - y||.$$

Величина L называется константой Липшица. Очевидно, константа Липшица определяется неоднозначно. Если L – константа Липшица, то любое число $L', L' \ge L$ также будет константой Липшица. Через $L_f(S)$ будем обозначать некоторую константу Липшица, а через $L_f^*(S)$ – минимальную.

Константа Липшица может быть оценена детерминированным [8], либо эвристическим [9] способом. На практике обычно применяют второй вариант. В данной работе применялась следующая эвристика. На параллелепипеде B строилась равномерная прямоугольная сетка с заданной гранулярностью, определяемым числом узлов N по каждому измерению. Общее число узлов в сетке составляет N^n.

В каждом узле сетки вычисляется значение целевой функции. Обозначим множество всех узлов сетки через M. Тогда константу Липшица оценим как

$$L \approx \max_{u,v \in M} \frac{|f(u) - f(v)|}{||u - v||}. \quad (2)$$

В формуле (2) предполагается, что $u \ne v$. Здесь и далее используется Евклидова норма. Чем больше число точек в сетке, тем точнее оценка. Пусть $\underline{f}(S) = \min_{u \in M} f(u)$, $\overline{f}(S) = \max_{u \in M} f(u)$.

Тогда положим

$$m(S) = \underline{f}(S) - \alpha(\delta) L \delta, \quad (3)$$

$$M(S) = \overline{f}(S) + \alpha(\delta) L \delta, \quad (4)$$

где δ – диаметр ячейки (расстояние между двумя самыми удаленными точками), а $\alpha(\delta) \ge 1$ – поправочный коэффициент, отражающий точность оценки константы Липшица. Данный коэффициент является возрастающей функцией от диаметра ячейки сетки. Для вычислительных экспериментов было выбрано значение $\alpha(\delta) = 1 + \gamma \frac{d(B)}{d(B_0)}$, где $d(B_0)$ – диаметр исходного, а $d(B)$ – текущего параллелепипеда. Количество узлов сетки при этом не зависит от диаметра параллелепипеда.

Рассмотрим вспомогательную функцию $eval(B)$, которая относит параллелепипед B к одному из классов: «внутренний» (internal), «внешний» (external) или «граничный» (boundary). В цикле производится проверка всех ограничений. Параллелепипед классифицируется как внешний, если хотя бы одно из ограничений заведомо несовместно ($m_i(B) > 0$). Если таких ограничений нет, но хотя бы для одного из них выполняется $M_i(B) > 0$, то параллелепипед считается «граничным», т.к. нельзя однозначно утверждать, что все его точки удовлетворяют системе неравенств. Если все неравенства заведомо выполнены на рассматриваемом параллелепипеде, то он классифицируется как «внутренний».

procedure eval (B)
$rv := internal$
for $i := 1$ **to** m **do**
  if $m_i(B) > 0$ **then**
    $rv := external$
    break;
  else if $M_i(B) > 0$ **then**
    $rv := boundary$
  endif
done
return rv

Последовательный алгоритм SEQAPPROX для построения аппроксимации множества решений системы (1) основан на концепции ветвей и границ. Алгоритм работает с четырьмя списками. Текущий список Q используется для реализации МВГ. Изначально он инициализируется исходным параллелепипедом B_0. Списки Q_I, Q_E, Q_B содержат внутренние, внешние и граничные параллелепипеды соответственно. Изначально эти списки пусты.

На каждой итерации основного цикла из списка извлекается очередной параллелепипед B. Функция $eval(B)$ позволяет определить тип параллелепипеда (внутрений, внешний, граничный) и добавить его к соответствующему списку. Если параллелепипед является «граничным» и его диаметр не превосходит заданную точность ε, то он добавляется к списку Q_B. В противном случае, параллелепипед разбивается на два равных параллелепипеда вдоль самого длинного измерения. Полученные параллелепипеды добавляются к списку Q.

algorithm SEQAPPROX
$Q_I \coloneqq \emptyset$
$Q_E \coloneqq \emptyset$
$Q_B \coloneqq \emptyset$
$Q := \{B_0\}$
while $Q \neq \emptyset$ **do**
  take B **from** Q
  if eval(B) $=$ internal **then**
    $Q_I \coloneqq Q_I \cup B$
  else if eval(B) = external **then**
    $Q_E \coloneqq Q_E \cup B$
  else if $d(B) \leq \varepsilon$ **then**
    $Q_B \coloneqq Q_B \cup B$
  else
    $(B_1, B_2) \coloneqq split(B)$
    $Q \coloneqq Q \cup B_1 \cup B_2$
  endif
done

Алгоритм останавливается, как только список текущих параллелепипедов становится пустым. Конечность алгоритма следует из того, что задана нижняя граница диаметра параллелепипеда ε и того, что при разбиении вдоль максимального измерения происходит гарантированное уменьшение диаметра параллелепипеда. В результате будут получены три списка параллелепипедов Q_I, Q_E, Q_B обладающие следующими свойствами:

$$\cup_{B \in Q_I} B \subseteq X \subseteq (\cup_{B \in Q_I} B) \cup (\cup_{B \in Q_B} B).$$

Таким образом построена внутренняя и внешняя аппроксимация искомого множества.

III. Параллельный алгоритм

Существуют различные подходы к распараллеливанию метода ветвей и границ. Особенностью рассмотренного варианта метода является отсутствие необходимости обмена рекордными значениями целевой функции между параллельными потоками, что избавляется от лишней синхронизации.

Целевые списки Q_I, Q_E, Q_B используются только для хранения результата. Поэтому каждый поток может работать с локальной копией, которые после завершения параллельного участка объединяются в общий список.

Список Q, вообще говоря, является общим и должен модифицироваться всеми потоками одновременно. Это можно было бы реализовать с помощью критических секций, однако такой подход очень плохо масштабируется. Поэтому была применена другая идея, впервые предложенная в [7].

Параллельный алгоритм PARAPPROX работает с двумя векторами списков V_1 и V_2. Число списков в векторе совпадает с числом параллельных потоков. Потоки совместно обрабатывают все множество элементов списков вектора V_1. При этом, созданные в результате разбиения параллелепипеды добавляются каждым потоком в «свой» список из соответствующего элемента массива V_2. На следующей итерации списки V_1 и V_2 меняются местами.

Предполагается, что в параллельном участке, заключенном между операторами **parallel** – **endparallel** стартует nt потоков. Номер каждого потока хранится в локальной переменной t и используется для обращения к соответствующим элементам векторов. Основной внешний цикла while повторяется до момента, когда общее число элементов $size(V_1)$ в списках вектора V_1 не станет равным нулю.

Обработка множества элементов списка вектора V_1 производится в параллельном цикле **parfor**, итерации которого распределяются между параллельными потоками.

algorithm PARAPPROX
$Q_I \coloneqq \emptyset$
$Q_E \coloneqq \emptyset$
$Q_B \coloneqq \emptyset$
$V_1[1] \coloneqq \{B_0\}$
while $size(V_1) > 0$ **do**
  parallel
  for $i \coloneqq 1$ **to** nt **do**
    parfor $j \coloneqq 1$ to $size(V_1[i])$ **do**
      $B \coloneqq V_1[i][j]$
      if eval(B) $=$ internal **then**
        $V_I[t] \coloneqq V_I[t] \cup B$
      else if eval(B) = external **then**
        $V_E[t] \coloneqq V_E[t] \cup B$
      else if $d(B) \leq \varepsilon$ **then**
        $V_B[t] \coloneqq V_B[t] \cup B$
      else
        $(B_1, B_2) \coloneqq split(B)$
        $V_2[t] \coloneqq V_2[t] \cup B_1 \cup B_2$
      endif
    done
  done
  endparallel
  $clear(V_1)$
  $swap(V_1, V_2)$
done
$Q_I \coloneqq \cup_{i=1}^{nt} V_I[i]$
$Q_E \coloneqq \cup_{i=1}^{nt} V_E[i]$
$Q_B \coloneqq \cup_{i=1}^{nt} V_B[i]$

Алгоритм PARAPPROX был реализован с помощью OpenMP[10]. Для создания параллельного участка использовалась директива **parallel**, а для распределения итераций по потокам – директива **for**.

IV. Вычислительный эксперимент

Для вычислительного эксперимента была выбрана задача определения рабочей области планарного параллельного робота 3-RPR[1]. Данный робот представлен на рисунке 1. В названии 3 означает число степеней свободы, а буквы R,P,R специфицирует типы подвижных соединений. Данный работ, представленный на рисунке 1, представляет собой планарный манипулятор с шестью ротационными шарнирами для закрепления штанг и тремя линейными мехатронными двигателями с шарико-винтовой передачей. Двигатели изменяют длину штанг, соединяющих неподвижное основание с углами треугольной плоской платформы, тем самым управляя ее движением и ориентацией. В

дальнейшем будем считать основание и платформу равносторонними треугольниками.

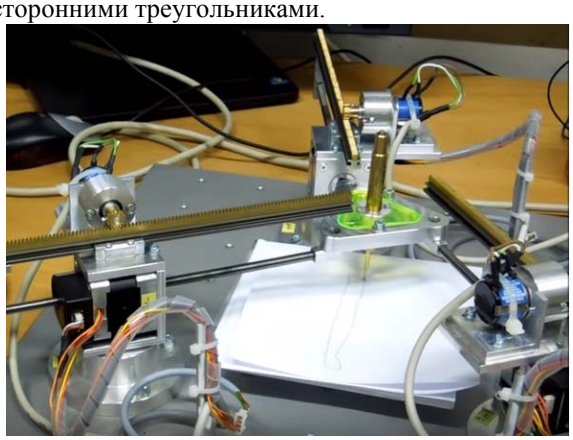

Рис. 1. Робот-манипулятор типа 3-RPR

Кинематическая схема робота представлена на рисунке 2. Положение манипулятора характеризуется параметрами, перечисленным в таблице 1.

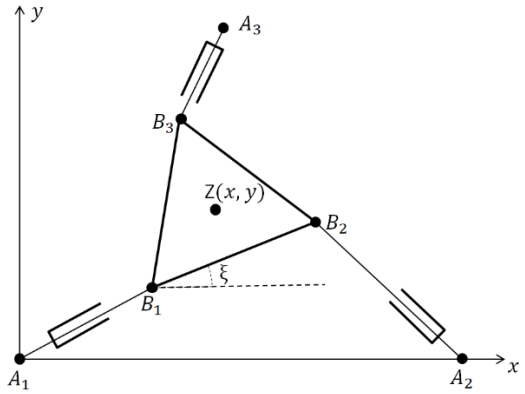

Рис. 2. Кинематическая схема робота 3-RPR.

Таблица 1. Параметры, характеризующие положение робота 3-RPR

Параметр	Описание
x, y	координаты центр подвижной платформы
ξ	угол поворота платформы относительно нейтрального положения
l_1, l_2, l_3	длины штанг $\overline{A_1, B_1}$, $\overline{A_2, B_2}$, $\overline{A_3, B_3}$ соответственно

Обозначим фиксированные координаты точек A_i в системе координат Oxy через $x_i^A, y_i^A, i = 1,2,3$. Свяжем с платформой подвижную систему координат, с началом в центре платформы. Предполагается, что в начальном положении оси подвижной системы координат параллельны осям системы Oxy. Фиксированные координаты точек B_i в подвижной системе координат, связанной с платформой, обозначим через $\hat{x}_i, \hat{y}_i, i = 1,2,3$. Величина ξ определяет угол поворота подвижной системы координат относительного начального положения против часовой стрелки. Тогда координаты точки B_i в системе Oxy задаются формулами:

$$x_i^B = x + \hat{x}_i \cos\xi - \hat{y}_i \sin\xi,$$
$$y_i^B = y + \hat{x}_i \sin\xi + \hat{y}_i \cos\xi. \quad (5)$$

Уравнения связи для робота 3-RPR определяются соотношениями $|A_i B_i| = l_i, i = 1,2,3$. Запишем уравнения связи для данного робота, используя введенные переменные:

$$(x_i^B - x_i^A)^2 + (y_i^B - y_i^A)^2 = l_i^2, \quad i = 1,2,3.$$

Подставляя выражения (1), получим:

$$(x + \hat{x}_i \cos\xi - \hat{y}_i \sin\xi - x_i^A)^2 + (y + \hat{x}_i \sin\xi + \hat{y}_i \cos\xi - y_i^A)^2 = l_i^2, i = 1,2,3. \quad (6)$$

Таким образом, получена система из трех нелинейных уравнений относительно шести переменных, перечисленных в таблице 1. Рабочая область определяется уравнениями (2) и системой интервальных ограничений, задающих диапазон $[l_i^{\min}, l_i^{\max}]$ изменения длин штанг l_i. Также целесообразно задать границы изменения переменных (x, y, ξ). Без ограничения общности можно положить:

$\xi \in [0,2\pi]$,

$$x \in \left[\max_{i=1,2,3}(x_i^A - l_i^{max}) - r, \min_{i=1,2,3}(x_i^A + l_i^{\max}) + r\right],$$
$$y \in \left[\max_{i=1,2,3}(y_i^A - l_i^{max}) - r, \min_{i=1,2,3}(y_i^A + l_i^{\max}) + r\right],$$

где r– расстояние от центра до углов платформы.

Можно упростить полученные уравнения, сведя их к системе неравенств, исключив переменные $l_i, i = 1,2,3$:

$$(x + \hat{x}_i \cos\xi - \hat{y}_i \sin\xi - x_i^A)^2 + (y + \hat{x}_i \sin\xi + \hat{y}_i \cos\xi - y_i^A)^2 - (l_i^{max})^2 \le 0, i = 1,2,3,$$
$$(l_i^{min})^2 - (x + \hat{x}_i \cos\xi - \hat{y}_i \sin\xi - x_i^A)^2 - (y + \hat{x}_i \sin\xi + \hat{y}_i \cos\xi - y_i^A)^2 \le 0, i = 1,2,3,$$
$$\xi \in [0,2\pi],$$
$$x \in \left[\max_{i=1,2,3}(x_i^A - l_i^{max}) - r, \min_{i=1,2,3}(x_i^A + l_i^{\max}) + r\right],$$
$$y \in \left[\max_{i=1,2,3}(y_i^A - l_i^{max}) - r, \min_{i=1,2,3}(y_i^A + l_i^{\max}) + r\right].$$

Эксперименты проводились двух платформах: персональном компьютере с процессором Intel Core i7-6700 Skylake 3400MHz и 16GB оперативной памяти и сервере 2-х процессорном сервере IBM Power Systems AC922, содержащем 2 процессора IBM Power 9 с частотой 3.8 Гц и 1024 Гб оперативной памяти. Каждый процессор Power 9 содержит 20 ядер с поддержкой технологии SMT4, что обеспечивает общее количество доступных логических ядер, равным 160.

Для экспериментов были выбраны следующие параметры расчетной модели:

$$||A_1 - A_2|| = ||A_2 - A_3|| = ||A_3 - A_1|| = 6,$$
$$||B_1 - B_2|| = ||B_2 - B_3|| = ||B_3 - B_1|| = 2,$$
$$l_i^{\min} = 1, l_i^{\max} = 6, i = 1,2,3.$$

Изменение угла ξ было ограничено диапазоном $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Гранулярность покрытия ε была установлена равной 10^{-2}. Время расчета и ускорение для различного числа потоков приведено в таблицах 2 и 3 для каждой из рассматриваемых платформ.

Таблица 2. Время расчета и ускорение для Intel

Число потоков	1	2	4	8	16
Время (с)	68,79	35,23	18,61	11,88	11,87
Ускорение	1	1,95	3,69	5,79	5,79

Таблица 3. Время расчета и ускорение для IBM

Число потоков	1	2	20	40	80	160
Время (с)	117,22	59,84	7,89	5,53	3,91	3,26
Ускорение	1	1,95	14,85	21,19	29,97	35,95

Как показывают полученные результаты, для первой из рассматриваемых платформ время расчетов снижается при увеличении числа потоков до 8. После этого время перестает уменьшаться, что легко объясняется тем, что максимальная степень параллелизма с учетом гипертрейдинга для данного процессора составляет 8. Аналогичные наблюдения можно сделать для платформы IBM, для которой максимальное ускорение достигается при 160 потоков.

V. Визуализация результатов расчетов

Визуализация рабочей области робота является сложной задачей в случае, если она строится в пространстве 3х и более измерений. Рабочая область рассматриваемого в статье робота 3-RPR имеет сложную структуру с полостями внутри, что затрудняет ее восприятие.

Для визуализации рабочей области робота было спроектировано и разработано мобильное приложение для платформы Apple iOS 11. Приложение позволяет строить набор параллелепипедов в дополненной реальности (Augmented Reality, AR), повышая наглядность результата и, вследствие этого, улучшая его восприятие пользователем. Результаты визуализации рабочей области робота 3-RPR при изменении угла ξ было в диапазоне $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ представлены на рисунках 3 и 4.

Технология дополненной реальности заключается в размещении виртуальных объектов в физической среде с отображением полученной комбинации на экране устройства. Рядом исследователей было отмечено, что применение дополненной реальности значительно повышает восприимчивость пользователя к отображаемой информации, благодаря чему этот инструмент широко используется в образовательной, музейной и производственной деятельности. Этим также объясняется интерес многих компаний к данной технологии. В частности, в разработанном приложении используется ARKit – набор инструментов для моделирования дополненной реальности, представленный корпорацией Apple в 2017 году.

Принцип работы ARKit заключается в отслеживании позиций характерных точек пространства для измерения расстояний, что позволяет наиболее достоверно интегрировать виртуальные объекты в реальную инфраструктуру. Таким образом, в ARKit используется безмаркерная технология дополненной реальности, где в противовес маркерному подходу у размещаемого виртуального объекта нет привязки к области фиксированного вида (QR-код или распечатанное изображение).

Интерфейс приложения, приведенный на Рис. 3, представляет собой вывод обработанных кадров с камеры устройства в реальном времени. После запуска приложения происходит анализ физической среды, после которого желтый индикатор вверху экрана сменяется зеленым. После этого пользователь может нажатием на экран указать место, где будет расположена модель, и набор полученных ранее параллелепипедов будет построен в этом месте. Когда модель построена, ее можно повернуть перемещением одного нажатия в сторону (pan gesture), масштабировать сближением или отдалением двух нажатий (pinch gesture), а также можно «срезать» модель до определенного уровня с помощью шкалы с ползунком (slider) в правой части экрана.

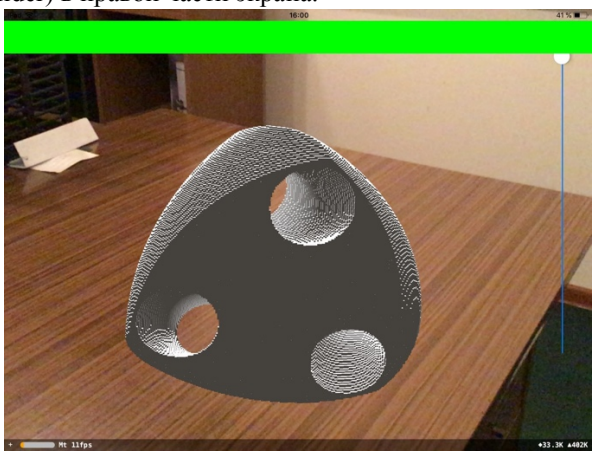

Рис. 3. Интерфейс приложения

Для рендеринга сцены используется стандартный для операционной системы iOS набор инструментов SceneKit. Виртуальная сцена понимается как набор узлов (SCNNode), имеющих определенную геометрию (SCNGeometry). Для построения модели в указанной пользователем точке создается «фиктивный» (без геометрии) виртуальный узел, а затем параллелепипеды добавляются в него как дочерние элементы. Такая организация позволяет при необходимости поворота модели поворачивать лишь основной узел, а не все параллелепипеды. При добавлении новых узлов используются стандартные средства SceneKit, в результате чего разработка облегчается отсутствием необходимости ручной реализации рендеринга.

При создании «среза» (Рис. 4) происходит сравнение координат центра каждого параллелепипеда (x_c, y_c, z_c) и его высоты h с H_{max} – максимальной высотой модели (определяемой исходными данными и текущим масштабом), умноженной на значение шкалы масштабирования (от 0 до 1). Если $y_c - H_{max} \geq {}^{h}/_{2}$, параллелепипед становится полностью невидимым; если $y_c - H_{max} < {}^{h}/_{2}$, параллелепипед становится полностью видимым; иначе вычисляется значение непрозрачности: $opacity = \frac{H_{max} - y_c + {}^{h}/_{2}}{h} \in [0,1]$, и оно устанавливается для параллелепипеда.

Входными данными для приложения является файл в формате JSON, который содержит массив описаний каждого параллелепипеда. Параллелепипед задается значениями для шести ключей: x_center, y_center, z_center (координаты его центра), w, h, d (его размеры – ширина, высота и глубина). При этом важно отметить, что SceneKit использует систему координат, где ось y направлена вверх, ось x вправо, а ось z вдаль от зрителя.

Разработка описанного приложения позволила эффективно визуализировать полученные результаты и

представить их в реальной среде с высокой долей достоверности. Помимо этого, благодаря возможности создания «срезов» модели было показано, что область действия робота может быть изучена максимально детально.

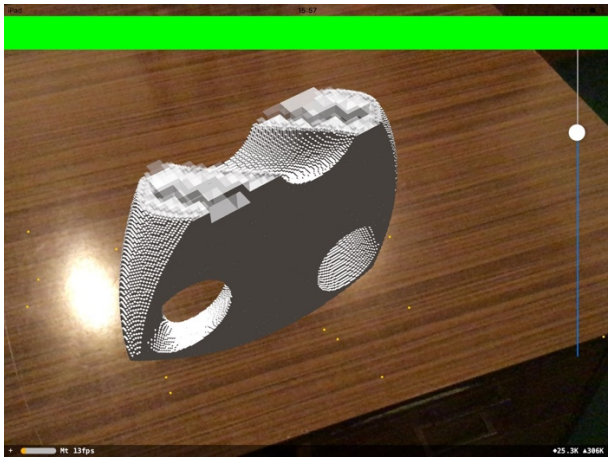

Рис. 4. Пример «среза» модели

VI. Заключение

В работе представлен программный комплекс для аппроксимации и визуализации рабочей области робота. Для построения аппроксимации рабочей области применен параллельный вариант метода неравномерных покрытий. За счет распараллеливания удалось добиться существенного ускорения расчетов. Для визуализации рабочей области применена технология «дополненной реальности», позволяющая эффективно визуально изучать рабочую область и помещать ее различный контекст. Также можно строить срезы, что позволяет изучать внутреннюю структуру рабочей области.

VII. Благодарности

Для выполнения расчетов был использован гибридный высокопроизводительный вычислительный комплекс ФИЦ ИУ РАН [11]. Коллектив авторов выражает благодарность руководству и сотрудникам ФИЦ ИУ РАН, предоставившим доступ и обеспечившим поддержку выполнения работа на этой платформе.

Библиография

[1] Merlet J. P. Parallel Robots. — Springer Publishing Company, Incorporated, 2010.

[2] Merlet J. P. Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries //The International Journal of Robotics Research. – 1999. – V. 18. – №. 9. – pp. 902-916.

[3] Jaulin Luc. Applied interval analysis: with examples in parameter and state estimation, robust control and robotics. — Springer Science & Business Media, 2001.

[4] Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization // Computer Science-Research and Development. – 2009. – V. 23. – №. 3-4. – pp. 211-215.

[5] Chakroun I., Melab N. Towards a heterogeneous and adaptive parallel Branch-and-Bound algorithm //Journal of Computer and System Sciences. – 2015. – V. 81. – №. 1. – pp. 72-84.

[6] Mezmaz M. et al. A multi-core parallel branch-and-bound algorithm using factorial number system //Parallel and Distributed Processing Symposium, 2014 IEEE 28th International. – IEEE, 2014. –pp. 1203-1212.

[7] Горчаков А.Ю., Посыпкин М.А., Ямченко Ю.В. Экспериментальное исследование трех вариантов реализации метода неравномерных покрытий для многоядерных систем с общей памятью //International Journal of Open Information Technologies. – 2018. – Т. 6. – №. 11 – С. 34-41.

[8] Evtushenko Y., Posypkin M. A deterministic approach to global box-constrained optimization //Optimization Letters. – 2013. – V. 7. – №. 4. – pp. 819-829.

[9] Strongin R. G., Sergeyev Y. D. Global optimization with non-convex constraints: Sequential and parallel algorithms. – Springer Science & Business Media, 2013.

[10] The OpenMP API specification for parallel programming. URL: https://www.openmp.org.

[11] Гибридный высокопроизводительный вычислительный комплекс ЦОД ФИЦ ИУ РАН. URL: http://www.frccsc.ru/hhpcc

Parallel algorithm for approximating the work space of a robot

A. Yu. Gorchakov, A. D. Ignatov, D. I. Malyshev, M. A. Posypkin

Abstract: The workspace of a robot is the set of positions that can be taken by its working tool. The understanding of the workspace is necessary when designing robots, planning their location, assessing their functionality, the path planning of the robot. So far many methods have been developed to determine the workspace. It should be noted that deterministic methods have the greatest potential, allowing to obtain approximations with a given accuracy in the automatic mode. A known disadvantage of deterministic methods is their high computational complexity, which prevents the effective wide application of these methods in practice. The paper proposes a parallel algorithm for constructing the approximation of the working area with guaranteed accuracy. The OpenMP package is used to create a multithreaded application. An original technique has been developed that allows to evenly distribute the load across the threads without explicit balancing. The results of experiments showing high efficiency of parallelization for multicore systems are presented. Another important problem is to visualize the constructed approximations. The workspace often has a complex structure with internal cavities. In this paper, we propose approaches for effective visualization of the working area of the robot, based on augmented reality technology, allowing not only to study the structure of the object, but also to place it in the desired visual context.

Keywords – working space of robots, approximation with guaranteed accuracy, parallel algorithms, visualization.

REFERENCES

[1] Merlet J. P. Parallel Robots. — Springer Publishing Company, Incorporated, 2010.
[2] Merlet J. P. Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries //The International Journal of Robotics Research. – 1999. – V. 18. – #. 9. – pp. 902-916.
[3] Jaulin Luc. Applied interval analysis: with examples in parameter and state estimation, robust control and robotics. — Springer Science & Business Media, 2001.
[4] Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization // Computer Science-Research and Development. – 2009. – V. 23. – #. 3-4. – pp. 211-215.
[5] Chakroun I., Melab N. Towards a heterogeneous and adaptive parallel Branch-and-Bound algorithm //Journal of Computer and System Sciences. – 2015. – V. 81. – #. 1. – pp. 72-84.
[6] Mezmaz M. et al. A multi-core parallel branch-and-bound algorithm using factorial number system //Parallel and Distributed Processing Symposium, 2014 IEEE 28th International. – IEEE, 2014. –pp. 1203-1212.
[7] Gorchakov A.Ju., Posypkin M.A., Jamchenko Ju.V. Jeksperimental'noe issledovanie treh variantov realizacii metoda neravnomernyh pokrytij dlja mnogojadernyh sistem s obshhej pamjat'ju //International Journal of Open Information Technologies. – 2018. – T. 6. – #. 11 – S. 34-41.
[8] Evtushenko Y., Posypkin M. A deterministic approach to global box-constrained optimization //Optimization Letters. – 2013. – V. 7. – #. 4. – pp. 819-829.
[9] Strongin R. G., Sergeyev Y. D. Global optimization with non-convex constraints: Sequential and parallel algorithms. – Springer Science & Business Media, 2013.
[10] The OpenMP API specification for parallel programming. URL: https://www.openmp.org.
[11] Gibridnyj vysokoproizvoditel'nyj vychislitel'nyj kompleks COD FIC IU RAN. URL: http://www.frccsc.ru/hhpcc