ISSN 2658-5014 (Print) СИИТ ISSN 2686-7044 (Online)
http://siit.ugatu.su 2025. Т. 7, № 5(24)
СИСТЕМНАЯ ИНЖЕНЕРИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
Научная статья

УДК 004.8 DOI: 10.54708/2658-5014-SIIT-2025-no5-p40 EDN: UDALGS

Transport-by-Throwing – робототехнический переброс: эксперименты и реализация

К. В. Миронов

Перемещение предметов перебросом — это перспективный способ робототехнической транспортировки деталей в гибких производственных системах. Описываются численные эксперименты с траекториями выборки, хранящимися в наборе данных. Эти траектории наблюдались системой компьютерного видения в экспериментах по метанию объекта. Цель численного эксперимента – проверить точность алгоритма и найти наилучшие настройки. Описываются технические аспекты интеграции алгоритма в систему транспортировки. Версия алгоритма реализована на языке C++ и интегрирована в программный комплекс для отслеживания полета брошенного мяча. Описана итоговая проверка алгоритма. Эксперименты показали, что система со встроенным предиктором способна успешно ловить брошенные мячи.

Робототехническая система; транспортировка; переброс; захват; статистическая оценка баллистических кривых; трекинг движущихся объектов; алгоритм ближайших соседей.

Введение

Эта статья завершает цикл публикаций [Мир24а, Мир24б, Мир25а, Мир25б], посвящённых перемещению предметов перебросом (Transport-by-Throwing) – перспективному способу транспортировки объектов в гибких производственных системах.

В предыдущей работе [Мир25б] была представлена концепция алгоритма прогнозирования. В этой статье будет объяснено, как этот алгоритм применяется к реальной системе транспортировки объектов путем бросания и ловли. Рассматриваются два вопроса: как реализован алгоритм и насколько хорошо он работает.

В разделе 1 объясняются численные эксперименты с траекториями выборки, хранящимися в наборе данных. Эти траектории наблюдались системой видения в экспериментах по метанию объекта, как описано в работе [Мир25а]. Цель численного эксперимента – проверить точность алгоритма и найти наилучшие настройки. В разделе 2 описываются технические аспекты интеграции алгоритма в систему транспортировки. Версия с параметрами, определенными в разделе 1, реализована на языке C++ и интегрирована в программный комплекс для отслеживания полета брошенного мяча. Окончательная проверка алгоритма описана в разделе 3. Эксперименты показали, что система со встроенным предиктором способна успешно ловить брошенные мячи.

1 . Численные эксперименты с набором данных

Эксперименты, описанные в этом разделе, в основном связаны с прогнозированием движения объекта по траекториям, которые наблюдались в экспериментах по ловле [Мир25а]. Сравнение результатов прогнозирования с фактическими положениями шара, наблюдаемыми системой видения, используется для оценки точности прогнозирования. Это преимущество численных экспериментов по сравнению с экспериментами по улавливанию. В экспериментах

Миронов К. В. Transport-by-Throwing – робототехнический переброс: эксперименты и реализация // СИИТ. 2025. Т. 7, № 5(24). С. 40-56. DOI: 10.54708/2658-5014-SIIT-2025-no5-p40. EDN: UDALGS.

Mironov K. V. “Transport-by-throwing – robotic throwing of objects: experiments and implementation” // SIIT. 2025. Vol. 7, no. 5(24), pp. 40-56. DOI: 10.54708/2658-5014-SIIT-2025-no5-p40. EDN: UDALGS (In Russian).

по ловле точное отслеживание невозможно, когда мяч находится внутри рабочего пространства робота. Робот движется, что искажает точность стереопозиционирования, поэтому единственным возможным результатом экспериментов по ловле является выявление того, поймал ли робот мяч или нет. Численные же эксперименты позволяют оценить точность прогнозирования.

Алгоритм k ближайших соседей использует набор ранее наблюдаемых траекторий в качестве основы для прогнозирования. В экспериментах использовались два типа набора данных траекторий:

Искусственный набор данных. Траектории в таком наборе создаются путем моделирования баллистического движения теннисного мяча. Поскольку теннисный мяч является хорошо изученным аэродинамическим объектом, это моделирование является относительно точным [Мир25б]. Искусственная генерация набора позволяет свести к минимуму временные затраты. Это необходимо для выполнения экспериментов по ловле для создания большого набора данных реальных бросков. Хотя набор данных искусственно сгенерирован в экспериментах, он использовался для прогнозирования реальных траекторий, наблюдаемых в экспериментах по метанию.

Полученный набор данных. Это набор реальных траекторий, полученных системой стереокамер. Траектории для прогнозирования также взяты из этого набора. Метод исключения применяется, когда каждая траектория из набора прогнозируется с использованием всех других траекторий, кроме самой себя. Другими словами, траектория, которая в данный момент прогнозируется, удаляется из набора данных до прогнозирования и возвращается впоследствии. Таким образом удается избежать ситуации, когда траектория использовалась бы для прогнозирования самой себя.

Вопрос для обсуждения заключается в том, какое количество доступных кадров используется для прогнозирования. В соответствии с настройками алгоритма, описанными в [Мир25б, разд. 2.3 и 3.1], прогнозирование выполняется в первый раз после получения кадра с номером 40. Теоретически точность прогнозирования должна повышаться с увеличением числа доступных кадров. Фактическое влияние номера кадра на точность прогнозирования обсуждается в конце этой работы. Алгоритм, описанный в [Мир25б], имеет ряд параметров, выбор значений которых влияет на качество прогноза. Различные основные настройки алгоритма перечислены ниже:

1. *Метод оценки плоскости полета* [там же, разд. 2.3]. Были предложены основные методы: наименьших квадратов, надежных наименьших квадратов, консенсуса случайной выборки, среднего значения выборки, медианы выборки. Метод выборочной медианы был окончательно выбран в качестве наиболее надежного, точного и быстрого.

2. *Параметр для сортировки траекторий в наборе данных* [там же, разд. 4]. Были предложены четыре основных параметра, и ими были скаляр скорости v, высота броска φ, вертикальная скорость v_1, горизонтальная скорость v_3. Горизонтальная скорость была окончательно выбрана в качестве параметра кластеризации.

3. *Весовые коэффициенты*. Это числа w_1 и w_2 с возможными значениями от 0 до 1 [Там же, разд. 3]. Предложены два способа определения этих коэффициентов, которые заключаются в том, чтобы установить их оба, равными 0.5 (простые ближайшие соседи), или определить значения с учетом пропорции расстояния (взвешенные ближайшие соседи).

4. *Размер кластера.* С одной стороны, уменьшение размера кластера означает уменьшение объема вычислений (уменьшение числа траекторий для сравнения с текущей). С другой стороны, это может снизить точность. Если размер кластера невелик, ближайшие траектории могут выйти за пределы кластера. В этом случае они не были бы признаны ближайшими соседями.

5. *Количество взятых ближайших соседей k* [там же, разд. 3 и 4]. Предполагалось, что $k = 2$ (один из взятых ближайших соседей лежит выше текущей траектории, а другой – ниже). Предположение о том, что этот выбор k является наиболее точным, должно быть доказано.

1.1. Простые и взвешенные ближайшие соседи

На первом этапе численных экспериментов была проверена точность базовой предлагаемой установки. Прогноз делается на основе первых 40 кадров после броска. Размер кластера установлен равным 25 траекториям, а k было установлено равным 2 – с одним соседом выше и одним соседом ниже траектории. Эти версии с простым и взвешенным прогнозированием ближайших соседей были проверены. Прогноз делается на основе одного и того же набора данных с использованием метода исключения. Пример успешных результатов прогнозирования показан на рис. 1. Таким образом было предсказано 150 траекторий. Результаты приведены в табл. 1.

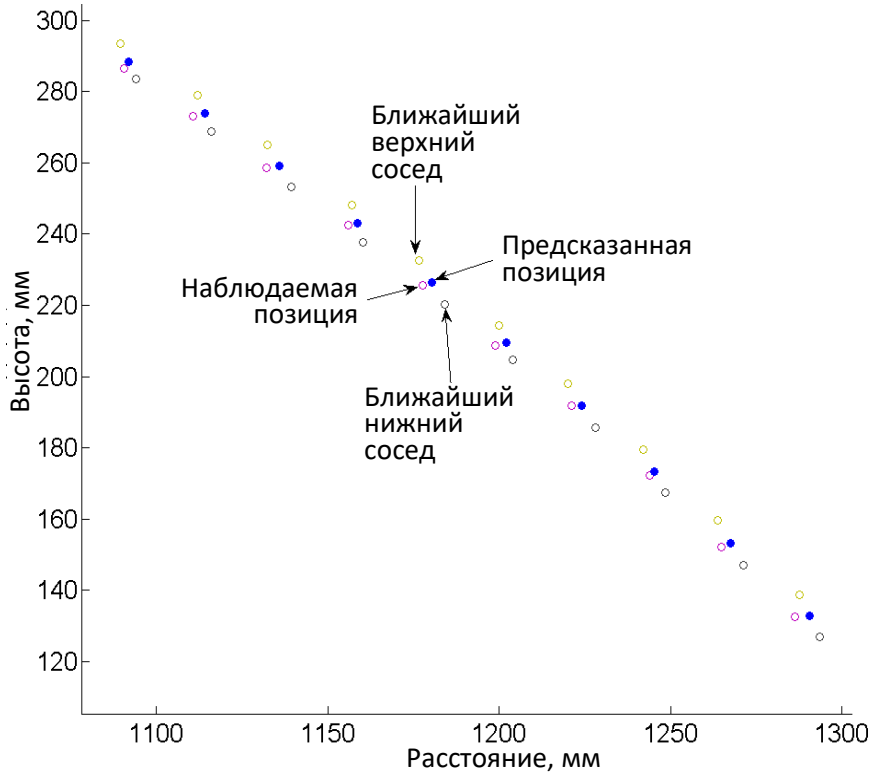

Рис. 1 Исходные положения шара с текущей траектории, его нижнего соседа и его верхнего соседа. Заполненные кружки показывают прогнозируемое положение объекта

Таблица 1

Точность прогноза

Параметр	Простой kNN	Взвешенный kNN
Процент бросков с ошибкой < 30 мм	92	85
Процент бросков с ошибкой < 20 мм	85	73
Средняя погрешность, мм	10	13

Процент траекторий с ошибкой прогнозирования меньше указанных пороговых значений приведен в табл. 1. Пороговые значения 30 и 20 мм выбраны в соответствии с [Bir11], где указано, что они позволили успешно поймать мяч робототехнической рукой. Простой алгоритм ближайшего соседа неожиданно продемонстрировал лучшую точность, чем алгоритм взвешенных ближайших соседей. Возможной причиной этого является высокая чувствительность пропорций к погрешности измерения. В дальнейших экспериментах использовалось простое предсказание ближайших соседей.

1.2. Размер кластера

Оценка размера кластера проводилась как с помощью искусственного набора данных, так и с помощью полученного набора данных. Эксперименты проводились с использованием набора данных, который состоял из 100 выборочных траекторий и 20 тестовых траекторий. Радиус кластера r равен 0.5 ($s - 1$), где s – количество траекторий в кластере. Если траектория с наиболее близким значением v_3 к текущей имеет порядковый номер i в отсортированном наборе данных, в кластер включаются траектории с порядковыми номерами от $i - r$ до $i + r$. Остальные настройки были такими же, как и в разделе 1.1. Средняя ошибка для различных размеров кластеров представлена в табл. 2. Здесь ∞ означает, что сортировка и кластеризация вообще не применялись; поиск ближайших соседей производился по всему набору данных.

Таблица 2

Средние ошибки прогнозирования для различных размеров кластера

r	e_{med}, мм	r	e_{med}, мм
1	25.3	7	10.6
2	14.2	8	7.8
3	13.3	9	7.8
4	12.5	10	7.8
5	11.9	11	6.6
6	10.6	12	6.6
		∞	16.7

Знак ∞ означает, что сортировка и распределение кластеров не производились. Прогноз был сделан на основе полученного набора данных реальных траекторий.

Результаты показывают, что ошибки уменьшаются с увеличением размера кластера. Это улучшение прекращается после $r = 11$. Для $r = 12, 13, 14, ...$ средняя ошибка сохраняет значение 6.6 мм. Это означает, что даже при большем размере кластера поиск возвращает ближайших соседей внутри кластера с $r = 11$. Когда размер кластера приближается к размеру всего набора данных, размер ошибки увеличивается. Результат показывает, что использование сортировки повышает не только скорость вычислений, но и точность прогнозирования.

Средняя ошибка прогнозирования может быть не лучшим параметром для оценки качества прогнозирования на относительно небольших наборах данных. Более важно знать, какой процент траекторий был предсказан точно, чем какова была ошибка прогнозирования для «средней траектории». На рис. 2 показано количество траекторий, предсказанных наиболее точным способом для различных размеров кластера. Прогнозирование наиболее точным способом означает, что для этих траекторий не существует размера кластера, который обеспечивал бы лучшую точность, чем текущий размер кластера. Эта гистограмма показывает те же результаты, что и в табл. 2, а именно, что наиболее эффективная работа алгоритма достигается при $r = 11$. Это почти 1/8 от всего объема набора данных (поскольку он включает 100 траекторий).

Результаты аналогичной оценки с искусственно созданным набором данных, состоящим из 2048 траекторий, представлены в табл. 3. В первом столбце представлена информация о точности, во втором – временные затраты. Результаты показывают, что минимальная медианная ошибка уже достигнута, когда размер кластера равен 256 траекториям.

Для реализации алгоритма в реальном времени проводится анализ времени, и эта реализация обсуждается в следующем разделе.

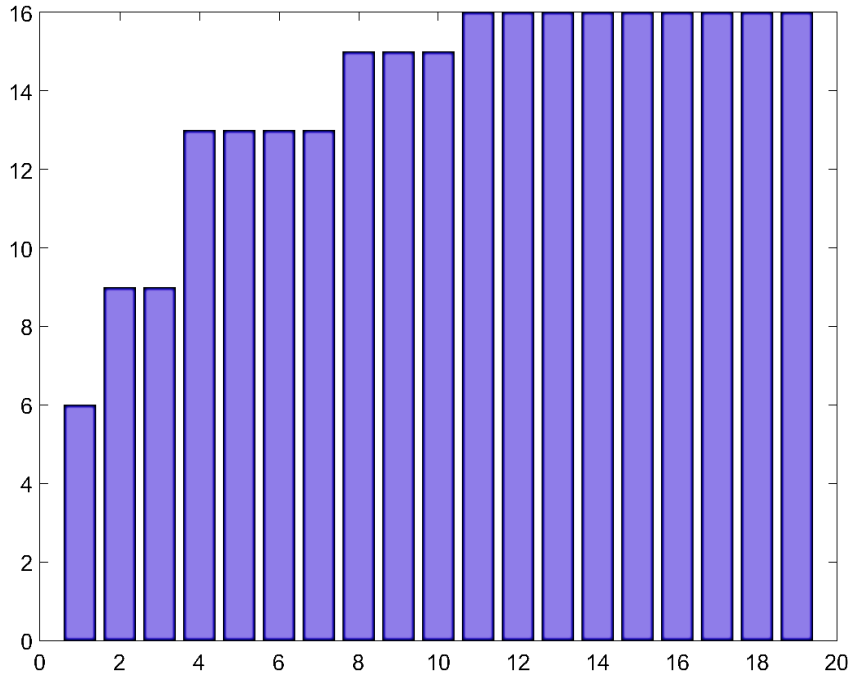

Рис. 2 Соответствие между количеством траекторий, которые прогнозируются с наилучшей точностью, и размером кластера

Таблица 3

Влияние размера кластера на точность и скорость прогнозирования

Размер кластера	Средняя ошибка прогнозирования, мм	Время вычисления (верхняя граница для 99,97%), мс
25	14.2	1.4
64	12.5	2.1
128	11.9	2.7
256	10.6	4.3
512	10.6	7.6
1024	10.6	15.2

Для искусственно сгенерированной базы данных 2048 выборок.

Влияние размера набора данных на скорость обработки не было значительным для размера базы данных от 64 до 2048 траекторий; оно составляло менее 1 мс для кластера с 25 траекториями каждый раз. Однако размер кластера оказывает существенное влияние на скорость вычислений (см. табл. 3). Он растет с увеличением размера кластера. Частота кадров наблюдателя составляет 100 кадров в секунду, поэтому межкадровый период равен 10 мс. Если время вычисления составляет менее 10 мс, то можно повторно вычислять прогноз после каждого нового полученного кадра. Для 256 траекторий в кластере время вычисления верхней границы равно 4.3 мс, что меньше 10 мс и, следовательно, достаточно.

Таким образом, в обеих установках достигается высокая точность при удовлетворительной производительности, когда кластер почти в 8 раз меньше всего набора данных. Результат не универсальный. Когда траектории в наборе данных более рассредоточены (например, скорость запуска варьируется от 2 до 7 м / с, но не от 4 до 5 м / с), относительный размер кластера должен быть меньше, но для текущих настроек размеры кластера были определены следующим образом: 25 траекторий в наборе для полученной базы данных и 256 траекторий в наборе для искусственной базы данных.

1.3. Значение *k*

В базовой настройке значение *k* было установлено равным 2. Цель этого подраздела – проверить, может ли этот выбор обеспечить наилучшую точность. Другой вопрос заключается в том, следует ли выбирать одну более высокую и одну более низкую ближайшую траекторию для обеспечения большей точности, а не просто выбирать двух ближайших соседей. С этой целью точность прогноза была проверена для различных значений *k*.

Когда *k* было установлено равным 1, прогноз траектории был просто установлен равным найденному ближайшему соседу. Когда он был установлен на 2, 3, ..., прогноз рассчитывался как среднее значение соответствующего числа ближайших соседей. Средние ошибки прогнозирования для этих версий алгоритма с различным *k* перечислены в табл. 4. Здесь $k = 2$ означает, что были взяты просто два ближайших соседа, в то время как $k = 1 + 1$ означает, что был взят один ближайший сосед выше и один ближайший сосед ниже. Остальные настройки алгоритма были такими же, как и в разделе 1.1.

Таблица 4

Средние ошибки прогнозирования для различного числа взятых соседей

k	Средняя погрешность, мм	*k*	Средняя погрешность, мм
1	15.7	4	12.9
2	10.1	5	13.0
1+1	6.6	6	13.8
3	10.6	7	15.6

1 + 1 означает, что был взят один ближайший сосед выше и один ближайший сосед ниже.

Результаты показали, что предложенное правило взятия соседей (двух ближайших соседей, один из которых выше, а другой ниже текущей траектории) обеспечивает наиболее точный прогноз. Это, в частности, обеспечивает более точные результаты, чем взятие двух ближайших соседей без каких-либо дополнительных правил. Использование одного прогноза ближайшего соседа снижает точность, а также увеличивает число соседей более чем до 2.

Набор численных экспериментов показал следующую настройку алгоритма, которая обеспечила наилучшие результаты с точки зрения точности и достаточные результаты с точки зрения производительности. Прогноз траектории рассчитывается как среднее значение соответствующих значений ближайшей более высокой траектории из набора данных и ближайшей более низкой траектории из набора данных. Поиск этих ближайших траекторий производится через кластер с 25 траекториями, если используется полученный набор данных со 100 траекториями, или 256 траекториями, если используется искусственный набор данных с 2048 траекториями. Значения размера кластера не являются универсальными; они хороши для этих конкретных экспериментальных установок. Реализация алгоритма была произведена на основе настроек, описанных выше.

2. Реализация

Алгоритм прогнозирования траектории реализован для использования в автоматизированной системе транспортировки путем метания. В этом разделе обсуждаются детали этой реализации. В подразделе 2.1 обсуждается, как алгоритм интегрирован в экспериментальное оборудование и программный комплекс для автоматизированного броска и ловли. Подраздел 2.2 посвящен тому, как будет выглядеть низкоуровневая реализация алгоритма прогнозирования.

2.1. Интеграция в транспортную систему

Окончательная проверка алгоритма проводится с помощью экспериментов по улавливанию. Эти эксперименты проводятся на специальном экспериментальном программном обеспечении и аппаратном комплексе для автоматизированного броска и ловли в Техническом университете Вены [Pon16]. Эта система включает в себя:

- метательное устройство с числовым программным управлением ([Мир24б, разд. 1.1]);
- роботизированный манипулятор KUKA LWR 4+ с 7 степенями свободы, который используется для ловли мячей;
- стереокамеры ([Мир25а, разд. 1.2]);
- подсистема обработки информации.

Структурная схема системы показана на рис. 3.

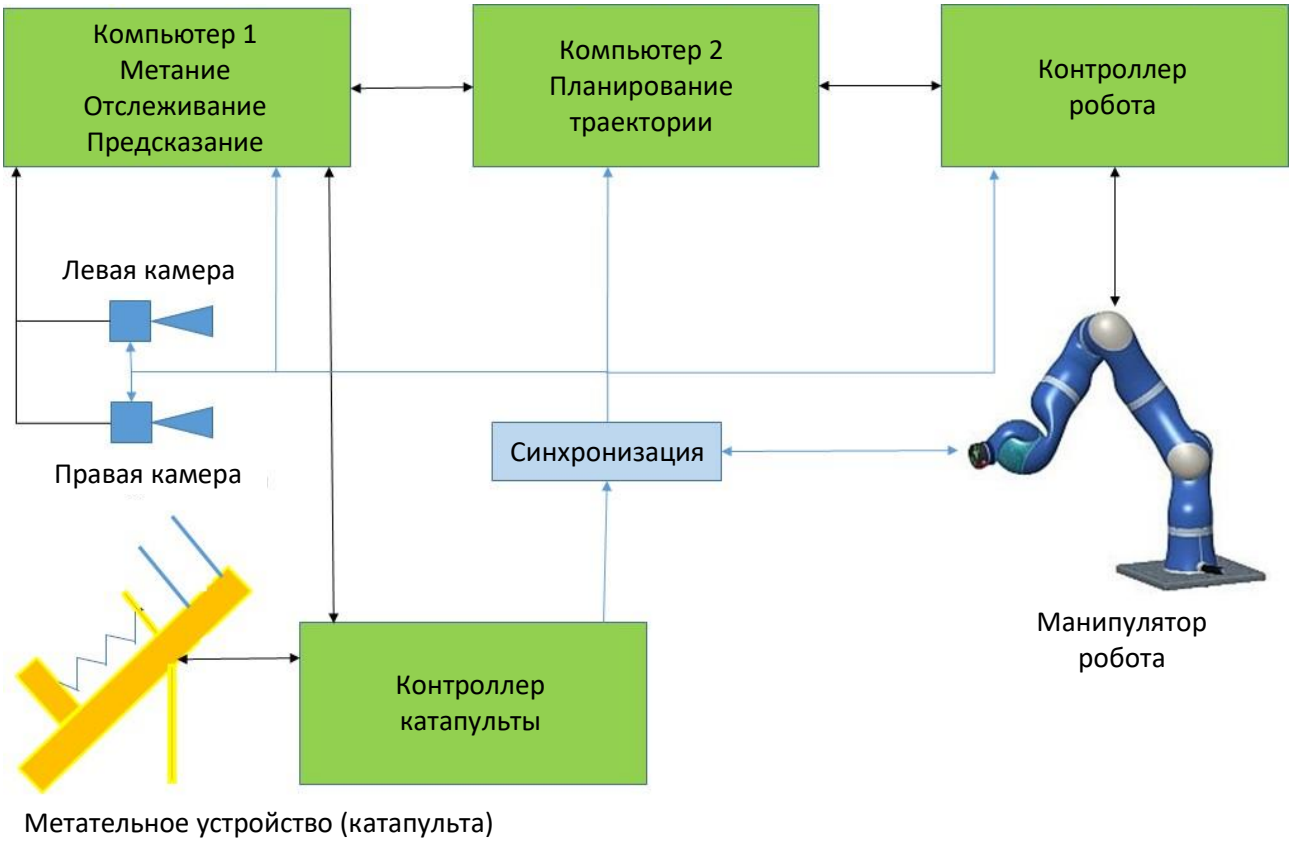

Рис. 3 Структура системы транспортировки с помощью метания

Связь между различными компонентами системы организована следующим образом [Pon16]. Обработка данных осуществляется в 2 персональных компьютерах (ПК-1 и ПК-2). ПК-1 предназначен для обработки данных с камер и для прогнозирования, а ПК-2 используется для определения инструкций по захвату. Метательное устройство управляется с ПК-1. Когда шар пересекает световой барьер, уведомление об этом событии отправляется генератору синхронизации, который координирует работу всей системы на основе времени цикла робота. Когда новая пара кадров поступает на ПК-1 с камеры, из изображений извлекается новое положение мяча, обновляется прогноз и выбирается новая точка захвата в пространстве и времени. Эта информация отправляется на ПК-2, где обновляются инструкции для робота.

Структурная схема всего алгоритма обработки данных показана на рис. 4. Алгоритм получает два изображения с камер в качестве входных данных. В качестве вывода он должен возвращать координаты точки, где будет происходить улов, и время этого события. Затем эти данные передаются на ПК-2.

Рис. 4 Структура алгоритма программного обеспечения

Сначала пиксельные координаты центра шара извлекаются из двух изображений с помощью распознавания кругов RANSAC (см. [Мир25а, разд. 1.2]). Этот шаг требует больших вычислительных затрат, поэтому он выполняется параллельно графическим процессором [Goe15]. Все остальные шаги выполняются последовательно в центральном процессоре. 3D-координаты центра шара извлекаются из координат пикселей с помощью операции стереотриангуляции [Goe15, разд. 1.3]. После этого делается новое предсказание. Результат предсказания является ориентиром для будущих положений центра шара.

Одна из этих позицийвыбирается в качестве позиции захвата [Pon16] и затем передается на ПК-2.

Этот алгоритм реализован в виде исполняемого кода на C++. Параллельное распознавание центра шара реализовано с использованием библиотеки CUDA. Предиктор реализован в виде функции C++. Детали этой реализации обсуждаются в следующем подразделе.

2.2. Предсказатель в реальном времени

Как было описано в [Мир25б, разд. 5], предиктор включает в себя два различных алгоритма: алгоритм прогнозирования и алгоритм обучения. Алгоритм обучения возвращает отсортированный набор данных выборочных траекторий, который затем используется алгоритмом прогнозирования. Поскольку алгоритм обучения выполняется до запуска транспортной системы, нет необходимости выполнять его в режиме реального времени. Таким образом, для создания отсортированного набора данных была использована простая реализация алгоритма обучения в MATLAB.

Алгоритм прогнозирования, в отличие от алгоритма обучения, должен работать в режиме реального времени и взаимодействовать с другими программными компонентами. Поэтому он был реализован как функция C++ в рамках общего программного проекта для обработки данных. В основном это прямая реализация вычислительных операций, описанных [Мир25б, разд. 2, 3, 4]. В реализации C++ предиктора следует упомянуть три особенности:

1. *Управление множественными вызовами*. Предсказатель вызывается каждый раз, когда новый кадр получен от системы видения; однако первое предсказание может быть сделано только после получения кадра с номером 40. До этого предсказатель должен сохранять только полученные координаты в статических переменных. После кадра с номером 40 предсказатель вычисляет прогноз после каждого нового кадра.

2. *Управление набором данных*. Набор данных выборочных траекторий занимает много памяти (например, набор данных с 2048 траекториями представляет собой массив размером 2048×100×3 с объемом более 1 мегабайта), и его инициализация может повлиять на время вычислений. Самый быстрый способ инициализации такого массива – вставить его непосредственно в программный код. Поэтому массив MATLAB, возвращаемый алгоритмом обучения, автоматически преобразуется в строку кода C++, которая затем вставляется в функцию прогнозирования.

3. *Необходимость быстрого выполнения двух матричных операций*. Это умножение матрицы 4×4 на вектор 4×1 и определение значения вектора 4×1 по известному результату умножения известной матрицы 4×4 на этот вектор. Эти операции используются много раз для привязки преобразований координат (см. [Мир25б, разд. 2]). Реализация этих операций обсуждается ниже.

В среде MATLAB матричные операции вставляются в языковую функциональность. Для матричных операций существуют специальные библиотеки In C++, однако использование этих библиотек приводит к более высоким временным затратам. Поэтому умножение матриц и противоположная операция были реализованы на низком уровне. В соответствии с общепринятыми правилами умножения матриц, если есть необходимость умножить матрицу 4×4 B и матрицу 4×1 Y, результатом будет матрица 4×1 Z:

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}, \quad Z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}.$$

Значения четырех строк матрицы Z можно найти следующим образом:

$$\begin{aligned} z_1 &= b_{11}y_1 + b_{12}y_2 + b_{13}y_3 + b_{14}y_4, \\ z_2 &= b_{21}y_1 + b_{22}y_2 + b_{23}y_3 + b_{24}y_4, \\ z_3 &= b_{31}y_1 + b_{32}y_2 + b_{33}y_3 + b_{34}y_4, \\ z_4 &= b_{41}y_1 + b_{42}y_2 + b_{43}y_3 + b_{44}y_4. \end{aligned}$$

Обратное преобразование определения неизвестного Y на основе известного Z производится путем решения системы линейных уравнений, выраженных матричным уравнением $B\,Y = Z$. Существуют ряд математических подходов к решению такой системы, но здесь выбрано правило Крамера из-за его вычислительной простоты. В методе Крамера значение Y определяется с помощью следующей последовательности вычислений:

$$\Delta = \det(B),$$

$$\Delta_1 = \det\begin{pmatrix} z_1 & b_{12} & b_{13} & b_{14} \\ z_2 & b_{22} & b_{23} & b_{24} \\ z_3 & b_{32} & b_{33} & b_{34} \\ z_4 & b_{42} & b_{43} & b_{44} \end{pmatrix}, \quad \Delta_2 = \det\begin{pmatrix} b_{11} & z_1 & b_{13} & b_{14} \\ b_{21} & z_2 & b_{23} & b_{24} \\ b_{31} & z_3 & b_{33} & b_{34} \\ b_{41} & z_4 & b_{43} & b_{44} \end{pmatrix},$$

$$\Delta_3 = \det\begin{pmatrix} b_{11} & b_{12} & z_1 & b_{14} \\ b_{21} & b_{22} & z_2 & b_{24} \\ b_{31} & b_{32} & z_3 & b_{34} \\ b_{41} & b_{42} & z_4 & b_{44} \end{pmatrix}, \quad \Delta_4 = \det\begin{pmatrix} b_{11} & b_{12} & b_{13} & z_1 \\ b_{21} & b_{22} & b_{23} & z_2 \\ b_{31} & b_{32} & b_{33} & z_3 \\ b_{41} & b_{42} & b_{43} & z_4 \end{pmatrix},$$

$$y_1 = \frac{\Delta_1}{\Delta}, \quad y_2 = \frac{\Delta_2}{\Delta}, \quad y_3 = \frac{\Delta_3}{\Delta}, \quad y_4 = \frac{\Delta_4}{\Delta}.$$

Характеристики производительности для реализованного алгоритма прогнозирования обсуждались выше в подразделе 1.2. Верхняя граница времени выполнения в 99.7% запусков равна 4.3 мс, что более чем в два раза меньше межкадрового периода. Это означает, что можно пересчитывать прогноз после каждого нового полученного кадра. Когда выполняются эксперименты по улавливанию (следующий раздел), прогнозирование каждый раз выполняется в течение межкадрового периода.

3. Эксперименты по захвату

Были проведены ряд экспериментов по ловле, чтобы проверить способность алгоритма давать удовлетворительные результаты в реальной системе транспортировки с помощью броска. Эксперименты проводились на экспериментальной транспортной системе, описанной в предыдущем разделе. Как уже упоминалось, роботизированная рука KUKA LWR 4+ используется в качестве улавливающего устройства. Захват установлен на рычаге. В нем используется принцип масляной *y*-сети. Внешний вид этого концевого эффектора показан на рис. 5. Сетка установлена на металлическом кольце диаметром 11 см. Поскольку диаметр шара равен 7 см, это кольцо допускает погрешность прогнозирования до 2 см.

Рис. 5 Устройства захвата, используемые для ловли мяча

Была выполнена серия бросков, чтобы поймать мяч. Номинальная скорость броска была установлена на 5 м/с, и в качестве основы для прогнозирования использовался третий набор данных. Из-за большого разброса фактических скоростей броска от номинальных значений траектории часто отличались от их типичного внешнего вида, и мячи не были пойманы успешно. Список бросков, отсортированных по скоростям броска, приведен в табл. 5. Привязка позиций объекта для примера успешного улова показана на рис. 6.

Согласно табл. 5, все мячи, брошенные со скоростью, близкой к номинальной (от 4.8 до 5.2 м/с), были успешно пойманы. Более высокие колебания скорости приводят к тому, что механизм теряет мяч. Дисперсия скоростей броска в экспериментах по ловле была выше, чем в обучающем наборе данных. Поэтому ожидается потеря мяча на экстремальных скоростях. Использование более рассредоточенных наборов данных должно уменьшить влияние этого эффекта. Успешные захваты на скоростях, близких к номинальным, доказывают, что алгоритм может быть успешно применен при транспортировке путем броска.

Таблица 5

Результаты экспериментов по ловле для 20 испытаний с различными скоростями от 4.5 до 5.5 м/с

№	v, м/с	Результат	№	v, м/с	Результат
1	4.54	потерян	11	5.02	пойман
2	4.58	потерян	12	5.04	пойман
3	4.67	потерян (отскок от рамы)	13	5.07	пойман
4	4.68	потерян (отскок от рамы)	14	5.13	пойман
5	4.68	потерян (отскок от рамы)	15	5.24	проиграл (отскок от сетки)
6	4.69	потерян (отскок от рамы)	16	5.25	потерян (отскок от рамы)
7	4.96	пойман	17	5.28	потерян
8	4.98	пойман	18	5.36	потерян
9	5.00	пойман	19	5.46	потерян
10	5.00	пойман	20	5.50	потерян

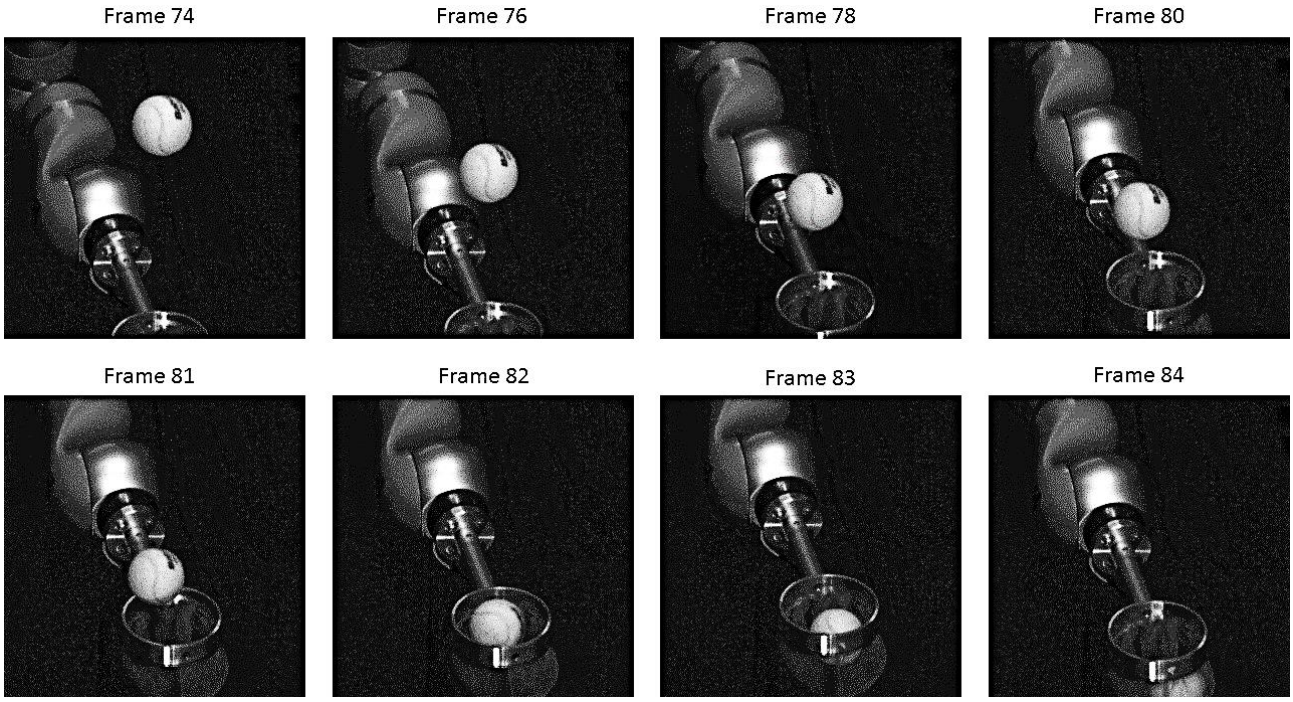

Рис. 6 Последовательность кадров камеры, представляющих успешный захват мяча

Заключение

Полученные результаты

В рамках данного исследования предложен, разработан и оценен алгоритм прогнозирования траектории брошенного тела, основанный на выборке. Алгоритм позволяет прогнозировать координаты летящего тела в режиме реального времени на основе измеренных координат на начальном этапе полета. Алгоритм применяется в роботизированной системе для ловли брошенных предметов, включающей в себя следующие компоненты.

Устройство для метания с числовым программным управлением. Оно бросает объект с определенной скоростью; однако реальная скорость броска имеет случайные отклонения от номинальной настройки. Метательное устройство снабжено двумя световыми барьерами, которые измеряют время и фактическую скорость броска.

Роботизированный манипулятор KUKA LWR4+ с 7 степенями свободы – используется для захвата мячей.

Подсистема камер. Отслеживание летящего сферического объекта в реальном времени осуществляется с помощью стереосистемы, состоящей из двух камер. Обе камеры оснащены датчиками изображения с разрешением 2048×2048 пикселей. Реконструкция местоположения объекта в пространстве осуществляется с помощью стереотриангуляции.

Подсистема обработки информации. Включает в себя два персональных компьютера, один из которых используется для обработки данных с камер, а другой – для формирования команд роботу. Подсистема также включает в себя контроллеры для управления роботом и метательным устройством.

Аппаратное обеспечение для синхронизации и связи между компонентами системы.

В ходе разработки алгоритма были решены следующие промежуточные задачи:

1. *Анализ производительности и точности наблюдателя.* Эта задача обсуждалась в работе [Мир25а]. Были проведены ряд экспериментов по метанию, чтобы изучить точность отслеживания и собрать наборы данных для изучения прогнозирования траектории. Определение положения пикселя шара на изображениях выполняется с помощью определения круга RANSAC [Мир25а, разд. 1.2]. Определение 3D-положения шара по его пиксельным позициям осуществляется с помощью стереотриангуляции [Мир25а, разд. 1.3]. Анализ точности показал, что наблюдатель определяет трехмерное местоположение статических сферических объектов со стандартным отклонением 2.25 мм [Мир25а, разд. 2]. Влияние объектов на сцене делает невозможным точное позиционирование летящего мяча на больших расстояниях [Мир25а, разд. 3.1]. Поэтому для преодоления этого влияния применяется вычитание фона. При использовании вычитания точность позиционирования для летающих объектов становится аналогичной результатам для статических объектов. Однако погрешности расстояния от мяча до камер слишком велики, когда это расстояние превышает два метра [Мир25а, разд. 3.2]. Когда камеры расположены за метательным устройством, эти ошибки локализуются в одном пространственном измерении, поэтому такое расположение камеры является предпочтительным. Полиномиальная подгонка в измерении глубины применяется для устранения ошибок на больших расстояниях [Мир25а, разд. 3.2].

2. *Разработка алгоритма прогнозирования траектории на основе машинного обучения*. В качестве базового метода прогнозирования траектории обоснован выбор метода k ближайших соседей (kNN) [Мир24б, разд. 5.4]. Прогноз делается на основе выборочных траекторий, хранящихся в наборе данных. Прогнозирование kNN выполняется в два этапа. На первом этапе выполняется поиск по набору данных. Этот поиск направлен на поиск траекторий в наборе, наиболее похожих (ближайших) к текущей траектории. Разработка операции поиска обсуждалась в [Мир25б, разд. 3.2]. Расстояние между соответствующими точками на траекториях используется в качестве показателя для определения сходства траекторий. В результате поиска возвращаются две траектории из набора: ближайшая более высокая траектория и ближайшая более низкая траектория. Второй шаг алгоритма состоит в прогнозировании текущих траекторий на основе известных ближайших соседей [там же, разд. 3.1]. Будущая часть текущей траектории определяется как средневзвешенное значение соответствующих частей ближайших соседей. Веса соседних траекторий могут быть определены двумя способами. Во-первых, в простых ближайших соседях веса обоих соседей устанавливаются равными 0.5. Во-вторых, во взвешенных ближайших соседях веса определяются относительно средних расстояний между соответствующими точками на текущей траектории и на ее соседях. Дальнейшее исследование показало, что простые ближайшие соседи обеспечивают более высокую точность, чем взвешенные ближайшие соседи.

Были внесены два улучшения в предсказатель ближайшего соседа:

1) В алгоритм была вставлена ссылка на преобразования координат. Стереотриангуляция возвращает 3D-координаты объекта в систему координат, подключенную к оптическому центру левых камер. В [Мир25б, разд. 2] предложено перевести эти координаты в новую систему координат (систему координат с нулевой точкой), которая определяется следующим образом. Начало координат системы координат состоит в том, чтобы поместить точку, измеренную

после броска, на траекторию. Одна координатная ось установлена так, чтобы она была коллинеарной с направлением силы тяжести, а другая установлена так, чтобы она была коллинеарной с горизонтальной проекцией скорости запуска. Траектории выборки преобразуются в эту систему координат и сохраняются в базе данных в координатах нулевой точки. Когда прогнозирование текущей траектории завершено, она сначала переводится в систему координат с нулевой точкой, затем прогнозируется, и, наконец, результат переводится обратно в мировую систему координат. Использование системы координат с нулевой точкой делает алгоритм инвариантным к пространственному положению точки запуска, горизонтальному направлению броска и расположению камер. Это также упрощает процесс обнаружения ошибочных измерений положения.

2) Сравнение траектории с относительно небольшим подмножеством всего большого набора данных [Мир25б, разд. 4]. Для этого для каждой траектории из набора оценивается средняя горизонтальная скорость полета. После этого все траектории в наборе сортируются по этому параметру. Когда прогнозируется текущая траектория, она сравнивается только с теми выборочными траекториями, которые имеют аналогичные значения горизонтальной скорости. Это приводит к значительному уменьшению объема вычислений, а также к ускорению поиска подмножества траекторий, аналогичных текущей.

Программное обеспечение для прогнозирования включает два основных модуля: алгоритм прогнозирования и алгоритм обучения [Мир25б, разд. 5]. Алгоритм обучения включает несколько операций, выполняемых с набором данных до прогнозирования: перевод траекторий выборки в систему координат с нулевой точкой; оценку средней горизонтальной скорости для каждой траектории выборки; сортировку траекторий в наборе по средней горизонтальной скорости. Алгоритм прогнозирования включает в себя набор операций, выполняемых при прогнозировании текущей траектории: перевод текущей траектории в систему координат с нулевой точкой; оценку средней горизонтальной скорости для текущей траектории; выделение подмножества аналогичных траекторий из всего набора данных; поиск в этом подмножестве двух ближайших соседей; прогнозирование текущей траектории на основе найденных ближайших соседей; перевод прогноза в систему координат робота.

3. *Оценка точности построенной модели*. Оценка производится путем применения алгоритма к траекториям, наблюдаемым камерами в экспериментах по метанию. Эти численные эксперименты описаны в разд. 1. Они используются для определения настроек алгоритма, обеспечивающих наилучшую точность и достаточную производительность. Эксперименты показали, что метод простых ближайших соседей демонстрирует лучшую точность, чем метод взвешенных ближайших соседей. Для текущей экспериментальной установки наилучшая точность обеспечивается, когда подмножество аналогичных траекторий составляет приблизительно 1/8 всего набора данных (разд. 1.2).

4. *Интеграция модуля прогнозирования в систему транспортировки путем броска*. Предсказатель встраивается в программный модуль для обработки данных с камер (разд. 2.1). Результаты прогнозирования затем передаются программному модулю, который впоследствии определяет захватное движение захвата. Предиктор реализован в виде функции C++, которая вызывается для обработки данных (раздел 2.2). Алгоритм обучения реализован в виде программы MATLAB. Он генерирует отсортированный набор данных, который затем используется предсказателем. Тестирование предсказателя показало, что он способен выполнять все предсказания в течение тайм-аута между получением двух кадров с камер.

5. *Итоговая оценка процесса транспортировки путем броска с помощью интегрированного алгоритма прогнозирования на основе обучения*. На заключительном этапе экспериментов транспортная система смогла поймать летающие шары на основе результатов прогнозирования, предоставленных алгоритмом (раздел 3). Это свидетельствует о применимости алгоритм для робототехнического захвата.

Таким образом, главным результатом исследования является новый метод прогнозирования траектории брошенного сферического тела. Основные отличия от известных предикторов состоят в следующем:

- точная физическая модель полета не требуется – предсказатель работает исключительно на основе предыстории;
- не требуются массивная параллельная обработка, мгновенная обратная связь и использование специального оборудования – все вычисления производятся на стандартном процессорном блоке за достаточное время;
- возможность сравнивать текущую траекторию с большим количеством траекторий, хранящихся в базе данных – все эти траектории учитываются без больших вычислительных затрат;
- независимость алгоритма от положения точки запуска, горизонтального направления броска и местоположения наблюдателя.

Перспективы развития темы

Проведенное исследование показало теоретическую применимость метода k ближайших соседей для прогнозирования траектории в системе транспортировки объектов путем броска и ловли. Этот результат может быть расширен за счет разработки новых версий алгоритма, ориентированных на применение в практических системах транспортировки. Эксперименты по захвату проводились на установке со следующими специфическими ограничениями, от которых целесообразно освободиться в будущем.

Относительно небольшое изменение параметров броска. Тренировочный набор включал броски со скоростью запуска от 3.7 до 5.2 м/с, в основном от 4 до 5 м/с. Это означает, что алгоритм способен точно предсказывать броски в определенном диапазоне скоростей запуска. Эксперименты по метанию показали, что захват пропускает мячи, брошенные со скоростью менее 4.7 м/с и более 5.2 м/с. Расширение работы алгоритма для более широких диапазонов скоростей потребовало бы сбора больших и более разрозненных наборов данных траекторий. Исследование точности и производительности алгоритма с использованием большего числа различных диапазонов траекторий является важным направлением работы.

Линейное метание предметов. Другие способы метания (например, броски с использованием быстро вращающегося рычага) не рассматривались. Линейные метательные устройства хороши тем, что они обеспечивают хорошую динамическую стабильность полета и позволяют свести к минимуму вращение (например, рычажный метатель, используемый в экспериментах в [Pon09], не мог бросать предметы без вращения). Однако бросание рычага может выполняться такими же робототехническими схватами, которые используются для ловли. Эти транспортные системы были бы более гибкими и универсальными, чем системы, использующие специальные метательные устройства. Предсказуемость траекторий, вызванных бросками на основе рычага, требует дополнительного изучения.

Унифицированные сферические объекты. В качестве экспериментального объекта для броска и захвата использовались теннисные мячи, аэродинамические свойства которых хорошо изучены. Однако реальные объекты, которые можно было бы транспортировать путем метания и захвата в промышленных условиях, естественно, будут иметь более сложную форму. Предсказание траекторий этих объектов приводит к ряду вопросов, которые необходимо решить. Прежде всего, это привело бы к увеличению числа параметров для прогнозирования. Положение сферических тел определяется тремя параметрами, то есть координатами центра масс в трехмерном пространстве. Представление траекторий в плоскости полета сокращает количество измерений до двух. Если объект имеет более сложную форму, появляются еще три параметра, которые выражают ориентацию объекта в пространстве [Kim12]. Эти параметры также нуждаются в разработке и применении алгоритма их прогнозирования. Эта задача рассматривалась в современной литературе, например, в [Fra12] и [Kim12], для других

алгоритмов прогнозирования. Еще один важный аспект применения алгоритма для прогнозирования асимметричных тел заключается в том, применимо ли представление плоскости полета или нет, или, другими словами, отклоняется ли траектория летящего объекта определенной формы в сторону или нет? Для некоторых объектов траектории не изгибаются, например, цилиндры из [Fra12], но для других они изгибаются. Дифференциация объектов требует дальнейшего изучения. Даже если траектория объекта изгибается, ее можно преобразовать в независимое от направления представление, выровняв одну из осей координат по горизонтальному направлению броска. В таком случае необходимо решить задачу определения этого направления для криволинейной траектории.

Особые возможности ловящего устройства. В экспериментах по ловле использовался захват, основанный на принципе сачка. Эта форма устанавливает жесткое соответствие между ошибкой прогнозирования и показателем успеха. Допустимая погрешность сделана относительно небольшой – 20 мм. Существуют ряд других конструкций захватов, которые могли бы обеспечить более высокую устойчивость к ошибкам прогнозирования [Nis97, Cig15]; однако ловля с использованием активного захвата [Nam03, Fra07, Bae11, Cig15] потребует разработки более сложных стратегий ловли.

Стратегия броска. Предмет бросается таким образом, чтобы облегчить улов, то есть бросающий хочет, чтобы ловец успешно поймал мяч. В промышленных условиях метатель настраивается таким образом, чтобы обеспечить максимально легкий улов. Стратегии для случаев, когда бросающий не хочет, чтобы объект был пойман, могут быть полезны для других применений прогнозирования траектории, например, для робототехнического настольного тенниса.

Использование системы стереовидения с двумя камерами. Увеличение количества камер может повысить точность системы видения. Например, размещение двух дополнительных камер там, где они могут наблюдать за зоной захвата, может устранить эффект увеличения ошибок в направлении глубины на больших расстояниях.

Сортировка траекторий по горизонтальной скорости. Поиск ближайших соседей производится в подмножестве траекторий с горизонтальной скоростью, аналогичной текущей. Были предложены еще два параметра: скаляр скорости и высота броска. Они были отклонены, так как их трудно оценить, но трудно – не значит, невозможно. Кроме того, следует найти дополнительные способы достижения распределения подмножеств.

Использование евклидова расстояния в качестве показателя близости. В качестве ближайших были выбраны соседи с наименьшим евклидовым расстоянием между соответствующими точками. Поиск и исследование других показателей для определения ближайших соседей также представляет интерес.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

[Bae11] Baeuml B., Schmidt F. et al. Catching flying balls and preparing coffee: Humanoid Rollin'Justin performs dynamic and sensitive tasks // IEEE International Conference on Robotics and Automation, Shanghai, China.2011. May. Pp. 3443 to 3444. DOI: 10.1109/ICRA.2011.5980073.

Baeuml, B., Schmidt, F., et al.: Catching Flying Balls and Preparing Coffee: Humanoid Rollin'Justin Performs Dynamic and Sensitive Tasks, IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 3443 to 3444, May 2011. DOI: 10.1109/ICRA.2011.5980073.

[Bir11] Birbach O., Frese U., Baeuml B. Realtime perception for catching a flying ball with a mobile humanoid // IEEE International Conference on Robotics and Automation, Shanghai, China. 2011. October. Pp. 5955 to 5962. DOI: 10.1109/ICRA.2011.5980138.

Birbach, O., Frese, U., Baeuml, B.: Realtime Perception for Catching a Flying Ball with a Mobile Humanoid, IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 5955 to 5962, October 2011. DOI: 10.1109/ICRA.2011.5980138.

[Cig15] Cigliano P., Lippiello V. et al. Robotic ball catching with an eye-in-hand single-camera system // IEEE Transactions on Control System Technology. 2015. Vol. 23. No. 5. May. Pp. 1657–1671. DOI: 10.1109/TCST.2014.2380175.

Cigliano, P., Lippiello, V., et al.: Robotic Ball Catching with an Eye-in-Hand Single-Camera System, IEEE Transactions on Control System Technology, Vol. 23, No. 5, pp. 1657-1671, May 2015. DOI: 10.1109/TCST.2014.2380175.

[Fra07] Frank H., Barteit D. et al. Autonomous mechanical controlled grippers for capturing flying objects // IEEE Int. Conf. on Industrial Informatics. Vienna. Austria. 2006. June. Pp. 431 to 436. DOI: 10.1109/INDIN.2007.4384796.

Frank, H., Barteit, D., et al.: Autonomous Mechanical Controlled Grippers for Capturing Flying Objects, IEEE Int. Conf. Industrial Informatics, Vienna, Austria, pp. 431 to 436, June 2006. DOI: 10.1109/INDIN.2007.4384796.

[Fra12] Frank T., Janoske U. et al. Automated throwing and capturing of cylinder-shaped objects // IEEE Int. Conf. Robotic and Automation. Saint Paul. Minnesota. USA. 2012. May. Pp. 5264 to 5270. DOI: 10.1109/ICRA.2012.6224565.

Frank, T., Janoske, U., et al.: Automated Throwing and Capturing of Cylinder-Shaped Objects, IEEE Int. Conf. Robotic and Automation, Saint Paul, Minnesota, USA, pp. 5264 to 5270, May 2012. DOI: 10.1109/ICRA.2012.6224565.

[Goe15] Goetzinger M. Object Detection and Flightpath Prediction: Diploma Thesis / Faculty of Electrical Engineering. Vienna University of Technology. 2015. June.

Goetzinger, M.: Object Detection and Flightpath Prediction, Diploma Thesis, Faculty of Electrical Engineering, Vienna University of Technology, June 2015.

[Kim12] Kim S., Billard A. Estimating the non-linear dynamics of free-flying objects" // Robotics and Autonomous Systems. 2012. June. Vol. 60. Pp. 1108-1122. DOI: 10.1016/j.robot.2012.05.022.

Kim, S., Billard, A.: Estimating the non-linear dynamics of free-flying objects, Robotics and Autonomous Systems, Vol. 60, pp. 1108-1122, June 2012. DOI: 10.1016/j.robot.2012.05.022.

[Nam03] Namiki A., Ishikawa M. Robotic catching using a direct mapping from visual information to motor command // IEEE Int. Conf. Robotics & Automation, Taipei. 2003. Sept. Pp. 2400-2405. DOI: 10.1109/ROBOT.2003.1241952.

Namiki, A., Ishikawa, M.: Robotic Catching Using a Direct Mapping from Visual Information to Motor Command, IEEE Int. Conf. Robotics & Automation, Taipei, pp. 2400-2405, Sept. 2003. DOI: 10.1109/ROBOT.2003.1241952.

[Nis97] Nishiwaki K., Konno A. et al. The humanoid Saika that catches a thrown ball // IEEE Int. Workshop on Robot and Human Communication, Sendai. Japan. 1997. October. Pp. 94 to 99. DOI: 10.1109/ROMAN.1997.646959.

Nishiwaki, K., Konno, A., et al.: The Humanoid Saika that Catches a Thrown Ball, IEEE International Workshop on Robot and Human Communication, Sendai, Japan, pp. 94 to 99, October 1997. DOI: 10.1109/ROMAN.1997.646959.

[Pon09] Pongratz M. Object Touchdown Position Prediction: Diploma Thesis / Faculty of Electrical Engineering. Vienna University of Technology. 2009. September.

Pongratz, M.: Object Touchdown Position Prediction, Diploma Thesis, Faculty of Electrical Engineering, Vienna University of Technology, September 2009.

[Pon16] Pongratz M. Bio-Inspired Transport by Throwing System: Dissertation / Faculty of Electrical Engineering. Vienna University of Technology. 2016. January.

Pongratz, M.: Bio-Inspired Transport by Throwing System, Dissertation, Faculty of Electrical Engineering, Vienna University of Technology, January 2016.

[Мир24а] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: обсуждение научно-технической задачи // СИИТ. 2024. Т. 6, № 1(16). С. 43–53. EDN: QGFZBW.

Mironov K. V. Transport-by-Throwing – a robotic method of moving objects by throwing: discussion of the scientific and technical problem. SIIT, 2024., Vol. 6, No. 1(16), pp. 43–53. EDN: QGFZBW.

[Мир24б] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: обзор используемых методов // СИИТ. 2024. Т. 6, № 3(18). С. 3–48. EDN: FUUPEN.

Mironov K. V. Transport-by-Throwing – a robotic method of moving objects by throwing: a review of the methods used. SIIT, 2024, Vol. 6, No. 3(18), pp. 3–48. EDN: FUUPEN.

[Мир25а] Миронов К. В. Transport-by-Throwing – робототехнический способ перемещения предметов перебросом: эксперименты по наблюдению за траекторией объекта // СИИТ. 2025. Т. 7, № 2(21). С. 3–29. EDN: AGHGVU.

Mironov K. V. Transport-by-Throwing — a robotic method of moving objects by throwing: experiments on observing the trajectory of an object. SIIT, 2025, Vol. 7, No. 2(21), pp. 3–29. EDN: AGHGVU.

[Мир25б] Миронов К. В. Transport-by-Throwing – робототехнический переброс предметов: алгоритм прогнозирования траектории // СИИТ. 2025. Т. 7, № 4(23). С. 3–28. EDN: TBEBLS.

Mironov K. V. Transport-by-Throwing – robotic transfer of objects: trajectory prediction algorithm. SIIT, 2025, Vol. 7, No. 4(23), pp. 3–28. EDN: TBEBLS.

ОБ АВТОРЕ | ABOUT THE AUTHOR

МИРОНОВ Константин Валерьевич
Уфимский университет науки и технологий, Россия.
mironovconst@gmail.com ORCID: 0000-0002-4828-1345
Доц. каф. вычислительной техники и защиты информации. Дипл. спец. по защите инф-и (Уфимск. гос. авиац. техн. ун-т, 2012). PhD (Техн. ун-т Вены, 2016). Иссл. в обл. робототехники, применения ИИ в техн. системах.

MIRONOV Konstantin Valeryevich
Ufa University of Science and Technology, Russia.
mironovconst@gmail.com ORCID: 0000-0002-4828-1345
Assoc. Prof., Department of Computer Science and Information Security. Information Security Specialist (Ufa State Aviation Tech. Univ., 2012). PhD (Vienna University of Technology, 2016). Research in the field of robotics and intelligent control.

МЕТАДАННЫЕ | METADATA

Заглавие: Transport-by-Throwing – робототехнический переброс: эксперименты и реализация.

Авторы: Миронов К. В.

Аннотация: Перемещение предметов перебросом — это перспективный способ робототехнической транспортировки деталей в гибких производственных системах. Описываются численные эксперименты с траекториями выборки, хранящимися в наборе данных. Эти траектории наблюдались системой компьютерного видения в экспериментах по метанию объекта. Цель численного эксперимента – проверить

Title: Transport-by-throwing – robotic throwing of objects: experiments and implementation.

Authors: Mironov K. V.

Abstract: Throwing objects is a promising method of robotic transportation of parts in flexible manufacturing systems. Numerical experiments with sample trajectories stored in a data set are described. These trajectories were observed by a computer vision system in object throwing experiments. The purpose of

точность алгоритма и найти наилучшие настройки. Описываются технические аспекты интеграции алгоритма в систему транспортировки. Версия алгоритма реализована на языке C++ и интегрирована в программный комплекс для отслеживания полета брошенного мяча. Описана итоговая проверка алгоритма. Эксперименты показали, что система со встроенным предиктором способна успешно ловить брошенные мячи..

Ключевые слова: Робототехническая система; транспортировка; переброс; захват; статистическая оценка баллистических кривых; трекинг движущихся объектов; алгоритм ближайших соседей.

Язык: Русский.

Статья поступила в редакцию 15 июля 2025 г.

the numerical experiment is to check the accuracy of the algorithm and find the best settings. Technical aspects of integrating the algorithm into the transportation system are described. The version of the algorithm is implemented in C++ and integrated into a software package for tracking the flight of a thrown ball. The final verification of the algorithm is described. The experiments showed that the system with a built-in predictor is capable of successfully catching thrown balls.

Key words: Robotic system; transportation; transfer; capture; statistical evaluation of ballistic curves; tracking of moving objects; nearest neighbor algorithm.

Language: Russian.

The article was received by the editors on 15 July 2025.