**АНАЛИЗ И СРАВНЕНИЕ КАЧЕСТВА РАСПОЗНАВАНИЯ МЕТОК ARUCO И APRILTAG С ДОПОЛНИТЕЛЬНЫМИ УСЛОВИЯМИ**

**© 2025 г. Т.Н. Белов, М. Курченко, Н. Кузнецов**

*НИИ «Московский физико-технический институт», Долгопрудный*

\* e-mail: [belov.tn@phystech.edu](mailto:belov.tn@phystech.edu)

\*\* e-mail:

\*\*\* e-mail:

Поступила в редакцию \_\_.\_\_.2025 г.

После доработки \_\_.\_\_.2025 г.

Принята к публикации \_\_.\_\_.2025 г.

Изображения маркеров семейств aruco, apriltag, ARTag, CALTag и других используются для обнаружения позиции через них. В работе выполнено сравнение качества обнаружения позиций. В работе представлено сравнение методов на основе aruco и apriltag. Также сравнивается обнаружение одного маркера и куба с маркерами на каждой стороне, обнаружение по отдельности и их, как последовательности, с фильтром Кальмана, обнаружение на основе сгенерированных фотографий и на основе реальных.

[Добавить кратко про результаты, когда будут]

**Ключевые слова**: aruco, apriltag, фильтр Кальмана, OpenCV, компьютерное зрение, композиция меток, виртуальное пространство

**Введение.** Задача определения расположения объекта, а именно его позиция и вращение относительно камеры, для устойчивости решается через обнаружение заранее определенных и расположенных на объекте маркеров. На данный момент основными являются aruco и apriltag с различными их вариациями, как ArUcoE (1), и другие комбинированные решения (2).

К задачам, в которых они используются, относятся дополненная и виртуальная реальность, навигация на складах, производствах, навигации дронов в каких-либо других пространствах и другое. В силу наличия разных методов, при использовании также стоит задача определения метода, который будет лучше в общем или лучше подходить под конкретную задачу. Некоторые исследования качества методов обнаружения уже проводились, как обнаружение в целях AR (3), исследование aruco (4), сравнение некоторых методов (5), добавление дополнительных методов, не связанных с самим обнаружением, как фильтр Кальмана (6), обнаружение сгруппированных тегов (7) и другие исследования.

Этот метод также ассоциирован с некоторыми проблемами, как частичная видимость объекта, решаемая использованием множества тегов на объекте и обнаружение тех, которые полностью видны (8), примеры: CALTag, ChArUco board. Также тени и размытие из-за движения или расфокусированной камеры (7) могут привести к нераспознованию маркера. Даже в случае распознования из-за того, что маркер плоский его вращение не может быть однозначно определено, из-за чего решения иногда дают неверный результат (9).

В силу этих причин для конкретных ситуация какой-то метод может быть лучше или хуже по приведенным причинам, а также по скорости работы и точности в целом. В случае этой работы основная цель это сравнение методов обнаружения меток для задачи определения положения робота на поле в робофутболе (на позиции головы закреплен куб с меткой на каждой из сторон, аналогично работе (10)). В соответствии с этим в работе также используются траектории, заданные по времени, на которых сравниваются обычные методы обнаружения и другие, подкрепленные фильтром Кальмана (на подобии (10), но без учета вращения) и дополнительными методами. В работе также сравнивается использование для обнаружения одной метки (плоской пластины с меткой) и использование ранее упомянутого куба с метками.

Дополнительно в работе приведено сравнение результатов на основе изображений, сгенерированных в виртуальном пространстве (vtk) и на основе изображений с реальной камеры, чтобы определить возможность проведения подобных исследований только на основе более простого в использовании виртуального пространства.

# Существующие сравнения методов

## 1.1. Общая теория обнаружения тегов

Процесс идентификации тегов aruco и в том числе apriltag описан на странице opencv (11). Он состоит из:

* Конвертации пикселей изображения в черный и белый цвет
* Обнаружение контуров тегов
* Убирание перспективы
* Обнаружение белых и черных квадратов тега
* Идентификация тега по композиции цветов составляющих квадратов
* Улучшение положения обнаруженных углов тега
* Решение задачи перспективы n точек (определение положения тега относительно камеры)

В силу принципа обнаружения у рассматриваемых тегов различают два параметра: количество битов (соответствует количеству пикселей, которые могут быть черными или белыми), расстояние Хемминга.

Чем больше количество битов, тем больше доступных тегов. Однако и при одинаковом размере тега, на изображении его квадраты будут занимать меньшее количество пикселей и хуже распознаваться.

Расстояние Хемминга же показывает сколько из доступных тегов используется. При большем расстоянии, меньше тегов используется так, чтобы при неправильном обнаружении одного или нескольких пикселей тега можно было правильно определить номер исходного тега.

Подобное сравнение проводилось на основе aruco и apriltag с использованием двух семейств тегов (12).

### 1.2. Обнаружение заслоненных тегов

В силу принципа обнаружения тега нужно чтобы он был полностью виден на изображении. В задачах же часто тег может быть частично заслонен или быть частично в тени. Также уровень освящения и искажения изображения, например размытие, могут влиять на распознавание.

В таких случая часто решают задачу через обнаружение набора тегов на одной плоскости. Обнаружения хотя бы некоторых на полностью видимой части равносильно обнаружению всей группы тегов. К таким относятся CALTag и ChArUco board.

В силу невозможности обнаружения обычного частично заслоненного тега их сравнение невозможно. Тем не менее проводили сравнения решений из библиотек и выполненными через оригинальные методы, включая машинное обучение (1) (4).

## 1.3. Проблемы двойственности решения задачи обратной проекции

Теги являются плоскими, что упрощает решение задачи обратной проекции, но при этом приводит к двойственности решения этой задачи.

Два положения тега симметричных относительно плоскости зрения камеры будут изображены с небольшими отличиями на основе размера в силу перспективы, которые на большом расстоянии часто не позволяют определить действительное вращение тега.

Проблема часто решается через тени, окружение тега (8), модификации тега (13), выбор решения по истории обнаружений тега (10).

## 1.4. Использование последовательных обнаружений тега

В задачах часто тег нужно обнаруживать на протяжении некоторого времени. В таких условиях можно применить фильтр Кальмана и другие методы, использующие информацию о предыдущих позициях тега для корректировки последующих обнаружений.

Часто используют более простой, линейный фильтр Кальмана (5). Поскольку в задаче участвует вращение иногда используют расширенный кватернионный фильтр Кальмана (EQKF) (10).

1. **Предлагаемый метод анализа**

## 2.1. Семейства тегов

Основная цель эксперимента — определения более качественного детектора между aruco и apriltag, поэтому во всех экспериментах участвует сравнение этих двух методов. Используется aruco из библиотеки opencv и apriltag3 из публичного репозитория (apriltag). Всего использовали два семейства тегов aruco\_5x5 и apriltag36h11.

## 2.2. Композиции тегов

В задаче, на которую нацелена работа, используется куб. В силу этого также проводится сравнение качества обнаружения при участии куба (с тегом на пяти сторонах) [добавить изображение или сноску на него] с обнаружением только одного тега (тот же куб, но все теги кроме одного заслонены) [добавить изображение или сноску на него]. В обоих случаях по тегам определяется положение куба с системой координат по центру [добавить изображение или сноску на него]. Для определения позиции куба при нескольких обнаруженных тегах использовалось среднее позиций :

(где – позиция куба полученная используя i-ый обнаруженный тег) и среднее вращение, минимизирующее квадраты норм отклонений:

(где – вращение куба полученное используя i-ый обнаруженный тег).

В обоих случаях алгоритм обнаружения тега определяет позицию самого тега. По полученному номеру тега и информации о конструкции тегов применяется обратное преобразование локальной позиции тега по отношения к системе координат всей конструкции (куба) для определения положения самой конструкции по отношению к камере. [добавить изображение или сноску на него]

– положение тега (t) в системе координат камеры (c)

– вращение тега (t) в системе координат камеры (c)

– обратное вращение к

– результат применения вращения к вектору

Аналогично присутствуют объекты: конструкция (k), гриппер (g), манипулятор (m).

## 2.3. Типы позиций

Использовали, как наборы отдельных позиций, не связанных друг с другом, так и траектории с параметром времени и приближающими теоретическую съемку движения объекта. В случае с траекториями сравнивали также их последовательное распознавание с использованием линейного фильтра Кальмана на позицию и их распознавание, как отдельных изображений без информации о других. В случае фильтра Кальмана также сравнивали использование дополнительных возможностей: определение лучшей из двойственных позиций и использование только тега видного под меньшим углом из всех.

## 2.4. Окружение и задание позиций

Сравнили использование изображений, сгенерированных на основе виртуального пространства (vtk), и полученных от реальной камеры. Информация о камере и ее позиции используется для экспериментов с виртуальным пространством для проведения аналогичных замеров. Далее мы их сравниваем для определения возможности использования vtk, как единственного источника данных в подобных анализах качества методов обнаружения.

В обоих случаях мы храним и используем для выставления и анализа позиции в виде (от камеры к конструкции).

В случае реальной камеры позицию выставляются использую манипулятор. Так как манипулятор задает положение своего гриппера, мы замерили позицию объекта в координатах гриппера ( ) и применяем преобразование:

Потом по известному локальному переходим к позиции родителя:

В случае vtk проводятся аналогичные преобразования, пропуская второй этап, так как в этом случае условная позиция гриппера соответствует позиции конструкции.

Также в vtk мы повторяем положение камеры и ее внутренние параметры [если надо, добавить объяснение что именно имеется в виду], полученные на основе реальной камеры.

Внутренние параметры:

Размер изображения ( – пикселей по ширине, - по высоте). Тогда центр окна (), угол зрения (va) и трансформация будут (t):

У камеры также есть дисторсия, которую мы не повторяли в vtk, так как алгоритмы определения позиции тега исправляют дисторсию изображения перед своей работой.

1. **Эксперименты**

### 3.1. Установка с манипулятором

[картинка с обозначением манипулятора, кубика, камеры]

### 3.2. Независимые варьирования

Эти эксперименты направлены на выявление зависимости ошибки обнаружения от позиции тега. Для упрощения анализа в каждой траектории варьируются только два параметра из шести возможных (3 координаты позиции и 3 координаты вращения). Также для упрощения проведения замеров на манипуляторе траектории изначально заданы как варьирование относительно манипулятора. Эти диапазоны указаны, как .

Также приведен примерный перевод в координаты камеры, соответствующие замеренному на нашей установке, как .

Вращения указаны углами Эйлера. Если последовательность применения имеет значения, то она указана снизу диапазона.

В двух участвуют только смещения, в двух смещение и вращение (все вращения в градусах):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название |  |  |  |  |
| “x\_y” |  |  |  |  |
| “x\_z” |  |  |  |  |
| “x\_rx” |  |  |  |  |
| “x\_ry” |  |  |  |  |

### 3.3. Траектории

Эти эксперименты заданы формулой, описывающей движение объекта в зависимости от времени. Они были заданы так с целью симулировать съемку движения объекта с какой-то частотой кадров. Далее при обработке изображений в «хронологическом» порядке (по увеличению параметра времени в формуле), считая их за реальное движение объекта, можно применить фильтр Кальмана и другие приемы, использующие историю обнаружений одного и того же объекта.

Мы используем две траектории, одна без вращения и одна с вращением:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название |  |  |  |  |
| “traj\_1” |  |  |  |  |
| “traj\_2” |  |  |  |  |

### 3.4. Повторение в виртуальном пространстве

Далее мы проводим аналогичные измерения на vtk.

Для возможности сопоставления результатов мы используем те же внутренние параметры камеры, что и замеренные в реальности.

Также положения относительно виртуальной камеры равны тем, что использовались в реальности, с точностью до результатов алгоритма hand-eye (подробнее позже).

### 3.5. Дополнительные этапы

Обе задачи, описанные тут, мы решаем с помощью функций из библиотеки OpenCV.

Для работы с камерой необходимы ее внутренние параметры и дисторсия. Для этого мы решаем задачу калибровки, используя изображения с шахматным полем.

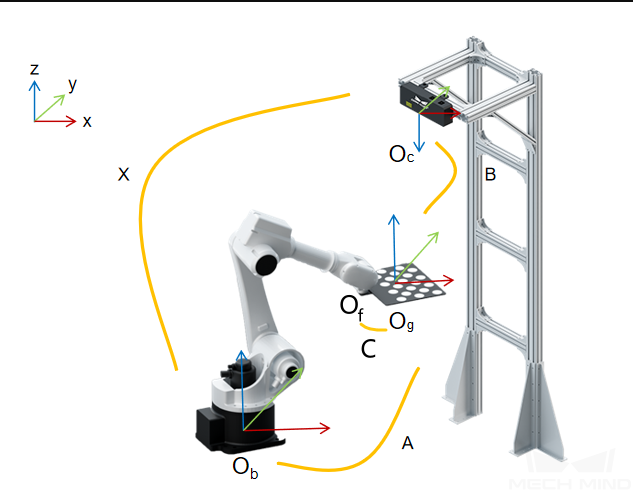
Также в случае с манипулятором мы задаем ему . Однако в случае распознавания позиции камерой мы получаем . По заранее заданным локальным преобразованием мы получаем одно значение .

В общем наш приоритет на распознавании, что не привязано к манипулятору. Поэтому мы храним сразу же , выставляем их на манипуляторе и используем для анализа.

Для их преобразования в , необходимо преобразование , которое и определяется алгоритмом hand-eye в конфигурации eye-to-hand.

Алгоритм по наборам и , включающими минимум два непараллельным вращения, определяет .

Примерная схема показана на картинке (по наборам , находится ).



1. **Результаты**
2. **Заключение**

# Ссылки

(б.д.). Получено из https://openaccess.thecvf.com/content\_CVPR\_2019/html/Hu\_Deep\_ChArUco\_Dark\_ChArUco\_Marker\_Pose\_Estimation\_CVPR\_2019\_paper.html

(б.д.). Получено из https://ieeexplore.ieee.org/abstract/document/1115078/

(б.д.). Получено из https://link.springer.com/chapter/10.1007/978-3-031-61582-5\_12

(б.д.). Получено из https://link.springer.com/chapter/10.1007/978-3-030-11292-9\_27

(б.д.). Получено из https://ieeexplore.ieee.org/abstract/document/8441049/

(б.д.). Получено из https://link.springer.com/article/10.1007/s10846-020-01307-9

(б.д.). Получено из https://www.sciencedirect.com/science/article/pii/S0031320314000235

(б.д.). Получено из https://ieeexplore.ieee.org/abstract/document/9196902/

(б.д.). Получено из https://ieeexplore.ieee.org/abstract/document/9650050/

(б.д.). Получено из https://www.researchgate.net/publication/316592734\_Relative\_Vessel\_Motion\_Tracking\_using\_Sensor\_Fusion\_Aruco\_Markers\_and\_MRU\_Sensors

(б.д.). Получено из https://docs.opencv.org/4.x/d5/dae/tutorial\_aruco\_detection.html

(б.д.). Получено из https://april.eecs.umich.edu/media/pdfs/krogius2019iros.pdf

(б.д.). Получено из https://www.researchgate.net/publication/362902063\_A\_novel\_encoding\_element\_for\_robust\_pose\_estimation\_using\_planar\_fiducials

(б.д.). Получено из https://github.com/AprilRobotics/apriltag