

# Разработка сервиса отслеживания траектории перемещения объекта при движении внутри помещения с использованием нескольких камер

Н.С. Складнев<sup>1</sup>, П.Ю. Якимов<sup>1,2</sup>

<sup>1</sup>Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

<sup>2</sup>Институт систем обработки изображений РАН - филиал ФНИЦ «Кристаллография и фотоника» РАН, Молодогвардейская 151, Самара, Россия, 443001

**Аннотация.** Траектория перемещения людей в контексте помещении очень важна как для обеспечения безопасности закрытой территории, так и для социальных или маркетинговых исследований. Основная цель настоящего научного исследования разработать интеллектуальную систему видеонаблюдения за объектом внутри помещения, характеризующуюся минимальным количеством ошибок при переходе между камерами. Для достижения поставленной цели был разработан метод межкамерного трекинга с использованием алгоритма позиционирования объекта на плане помещения, рассмотрены его основные этапы: локализация объекта на изображении, проецирование объекта на план помещения и комплексирование данных с нескольких камер. Также были проведены экспериментальные исследования алгоритма позиционирования объекта на схеме.

## 1. Введение

Движение это неотъемлемая часть жизни – движущиеся объекты могут быть живыми или не живыми, двигаться сознательно или под действием других объектов. Параметры движения: направление, скорость, длительность и другие, в совокупности с некоторым контекстом дают множество полезной информации.

### *1.1 Область исследований и актуальность*

В охранных системах, системах контроля движения, маркетинговых задачах, автоматизации производства необходимым является отслеживание движения различных объектов в видеопотоке в контексте помещения. В одних системах это позволяет следить за положением объекта в реальном времени на некотором плане, в других производить последующий анализ траекторий объектов с целью извлечения более сложной комплексной информации [1].

В последнее время видеокамеры окружили человека. Камеры встроены в мобильные телефоны, установлены на многих компьютерах. Обработать, хранить и просматривать такой большой объем информации крайне трудно, поэтому все больше набирают популярность системы компьютерного зрения.

В качестве примеров таких систем можно привести: систему определения очередей, которая позволяет сигнализировать при достижении определенного количества человек у кассы. Пропускная система, которая отслеживает поток входящих и выходящих лиц и сверяет их с

определенными информационными базами. Системы технического зрения в спорте, которые помогают судьям решить спорные ситуации и другие.

### 1.2 Проблемы трекинга

Трекингом называется определение местоположения движущегося объекта (нескольких объектов) во времени с помощью камеры [2]. Попадая в область зрения камеры, система трекинга закрепляет за новым найденным объектом уникальный идентификатор (*identifier, ID*) и за тем может соотнести объекты на следующем кадре с существующими идентификаторами или выдать новый, используя информацию о предыдущих состояниях каждого объекта. Так, у каждого объекта в системе есть его ID и вектор состояний. Именно эта информация позволяет в последующем построить траекторию движения объекта и проанализировать её. Трекинг является основой множества интеллектуальных систем компьютерного зрения.

В основе большинства алгоритмов трекинга лежит «фильтр Калмана» - алгоритм перехода. Основная идея фильтра Калмана состоит в том, чтобы использовать доступные обнаружения и предыдущие прогнозы, чтобы получить лучшее предположение о текущем состоянии, сохраняя при этом возможность ошибок в процессе [3]. Как видно на рисунке 1, фильтр Калмана работает рекурсивно, где он берет предыдущие показания, чтобы предсказать текущее состояние, затем измеряет и обновляет прогнозы. Таким образом, все сводится к выводу нового распределения (прогнозов) из предыдущего распределения состояний и распределения измерений.

Для трекинга объектов с одной камеры использовался популярный и элегантный алгоритм отслеживания объектов DeepSORT. Тут фильтр Калмана является важным компонентом – состояние содержит 8 переменных;  $(u, v, a, h, u', v', a', h')$ , где  $(u, v)$  - центры ограничительных рамок,  $a$  - соотношение сторон, а  $h$  - высота изображения. Другие переменные - это соответствующие скорости переменных. DeepSort представляет еще одну метрику расстояния, основанную на «внешнем виде» объекта. Для этого строится классификатор и обучается, пока он не достигнет достаточно хорошей точности, а затем отбрасывается окончательный слой классификации. Предполагая классическую архитектуру, останется полносвязный слой, создающий один вектор признаков, ожидающий классификации [3]. Именно этот вектор признаков и является «дескриптором объекта».

При отслеживании объекта на прямой дороге или в пустом пространстве справится даже самый простой трекер. Но в реальной ситуации возможны различные проблемы, которые либо снижают точность отслеживания, либо позволяют совсем потерять объект. Окклюзия объектов в видео является одним из наиболее распространенных препятствий для плавного отслеживания объектов (рисунок 2). Такая ситуация возникает когда два объекта расположены приблизительно на одной линии и один объект, расположенный ближе к камере частично или полностью закрывает видимость другого объекта. В таком случаи обычный трекер может потерять объект.

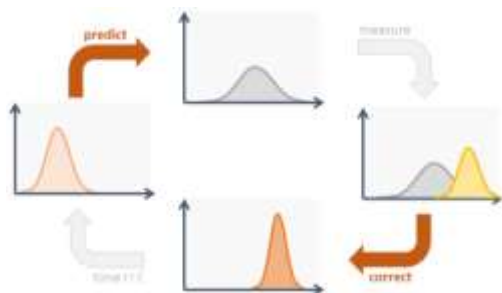


Рисунок 1. Цикл работы фильтра Калмана [3].



Рисунок 2. Пример окклюзии [3].

Под потерей объекта понимается ситуация, когда трекер не смог сопоставить найденный объект с уже имеющимися состояниями и выдал новый идентификатор, хотя на самом деле этот объект уже присутствовал на сцене ранее (также эту проблему называют *switch ID*) [4].

С другой стороны помещение, в котором применяется трекер, может быть сложной формы, такой, что объект покинет область видимости камеры на длительное время, а потом вернется назад – это также ситуация switch ID. И если с простыми окклюзиями трекер типа DeepSort справляется хорошо, то в сложных или многокомнатных помещениях проблема switch ID остается актуальной.

### 1.3 Постановка задачи

Основная цель настоящего научного исследования – разработать интеллектуальную систему видеонаблюдения с минимальным количеством switch ID.

Некоторые разработчики пытаются решить проблему switch ID расстановкой нескольких камер, видеопотоки с которых передаются в более совершенный DeepSort, использующий более «умный» дескриптор объекта, который позволяет сопоставить изображение объекта с разных, допустимых, точек зрения камеры [3].

Мы предполагаем, что использование информации о текущей позиции объекта на некоторой модели конкретного помещения в качестве состояния для трекера – более надежно и лучше масштабируется по количеству камер, чем модернизация универсального нейросетевого алгоритма DeepSort. Т.к. помимо перечисленных проблем окружения существует несколько способов «обмануть» DeepSort сознательно, например, если несколько человек будут использовать схожую одежду, что особенно актуально в сфере систем охраны закрытых территорий.

Для достижения поставленной цели был разработан метод межкамерного трекинга с использованием алгоритма позиционирования объекта на плане помещения (рисунок 3). Метод состоит из следующих этапов:

- локализация объекта в кадрах видеопотока каждой камеры
- перенос координат объекта на план помещения
- комплексирование данных с нескольких камер и трекинг.

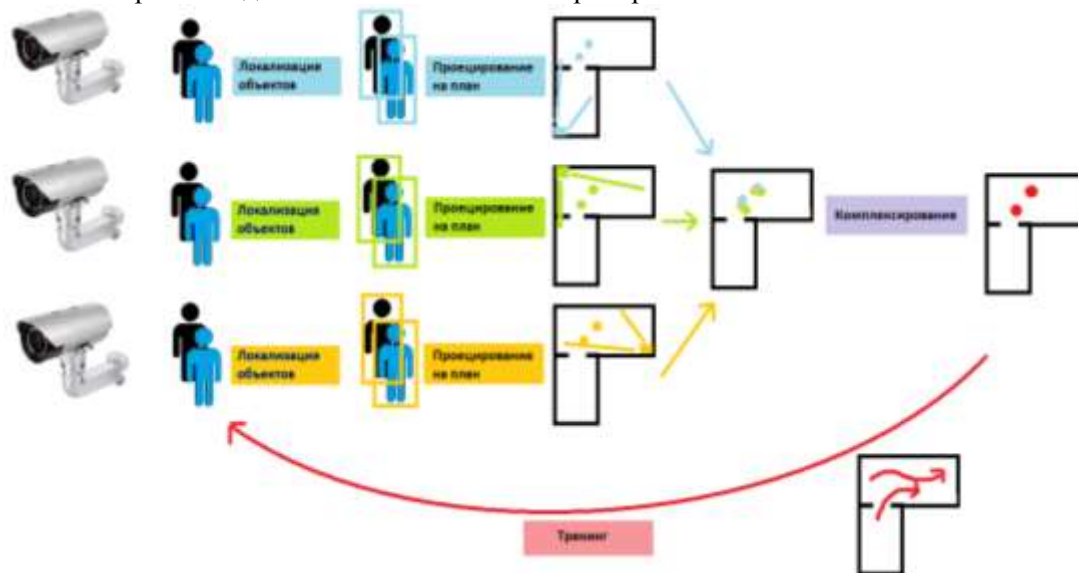


Рисунок 3. Схема метода межкамерного трекинга.

## 2. Метод межкамерного трекинга

### 2.1 Локализация объектов на кадре

Первым этапом работы системы межкамерного трекинга является локализация или обнаружение – т.е. определение границ всех объектов на кадре с каждой камеры.

Видео поток представляет последовательность изображений, не дающих никакой семантической информации. Чтобы получить из изображения некоторый смысл нужно

провести тот или иной анализ, такой как, например, детектирование и классификация объектов на изображении.

Под задачей обнаружения (детектирования) динамических объектов понимается задача обнаружения и выделения изменяющихся участков изображения в последовательности кадров [5]. Соответственно, под обнаружением определённого объекта понимается выбор одного или нескольких обнаруженных динамических объектов, которые имеют некоторые схожие признаки с заданным объектом поиска. Признаки выбираются согласно алгоритму.

Для обнаружения объектов на изображении была выбрана сверточная нейронная сеть YOLO, реализованная на основе библиотеки Tensorflow. Скорость обработки изображений позволяет применять YOLO для обработки видео с высокой частотой кадров. Кроме высокой скорости YOLO на определённых наборах данных превосходит по точности распознавания такие детекторы, как SSD [6] и Faster R-CNN [7].

Принцип работы модели YOLO представлен на рисунке 4. На входное изображение накладывается регулярная сетка, разделяющая его на  $S \times S$  областей. Для каждой области нейронная сеть определяет 5 ограничивающих рамок объекта, уровень достоверности (confidence score) обнаружения рамки, которая отражает степень уверенности модели в том, что поле содержит объект:

$$\text{confidence} = \Pr(\text{object}) * \text{IOU}(b, \text{object}), \quad (1)$$

где  $\text{IOU} = \frac{S_1}{S_2}$ ,  $S_1$  – площадь пересечения эталонной ограничивающей области и поля  $b$ ,  $S_2$  – площадь их объединения. Если объект в области  $b$  не существует, то confidence будет равен нулю. В результате работы сети выбирается область с наибольшим уровнем достоверности ответа сети.

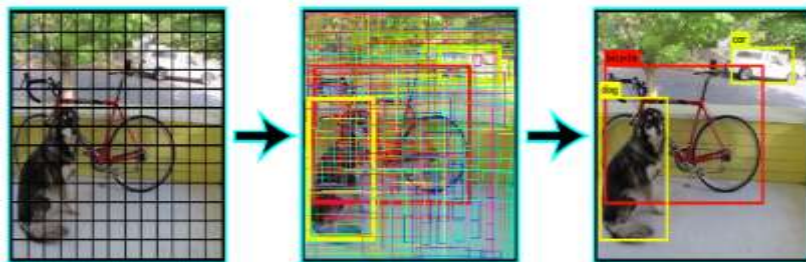


Рисунок 4. Принцип работы модели YOLO [8].

## 2.2 Позиционирование объектов на сцене

В общем случае, определение координат объекта по снимку с одной камеры является невыполнимой задачей. Но используя некоторые ограничения и допущения можно построить достаточно эффективный метод. Логично предположить, что наблюдаемый объект будет расположен на полу, а положение камеры в пространстве и её параметры остаются неизменными (рисунок 5а).

На рисунке 5б чёрная пунктирная линия определяет ось, проходящую через изображение объекта (голубая точка). Она смещена от главной оси на угол  $\beta$ .

Алгоритм использует информацию о камере и ее связи с системой координат помещения. Необходимая информация - это расстояние между центром камеры и полом, угол наклона камеры, фокусное расстояние камеры и размер датчика CMOS (или углы обзора камеры).

Основная идея этого метода состоит в том, чтобы полностью изменить проекцию камеры со сцены мира на плоскость изображения. Из обратной проекции можно получить исходную трехмерную позицию проецируемых точек. Это возможно из предположения, что объект стоит на полу. Без этого предположения можно было бы найти только линию  $l_p$  – совпадающую с главной осью камеры [9].

Линия в трехмерном пространстве может быть выражена с помощью скалярных параметрических уравнений ниже:

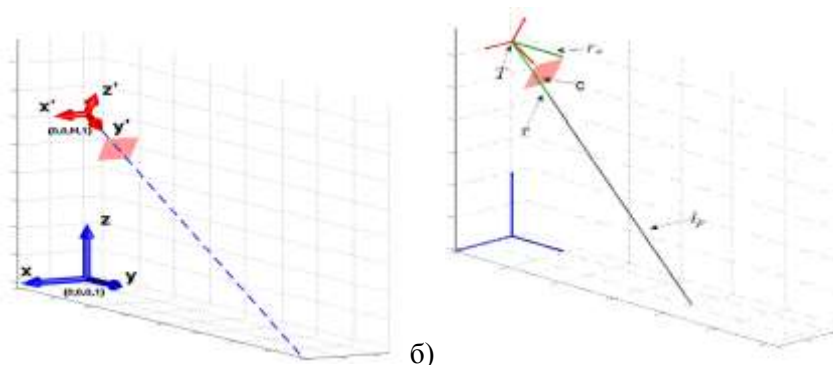


Рисунок 5. Направление главной оптической оси камеры и координат помещения.

$$\begin{cases} x = x_0 + a \cdot t \\ y = y_0 + b \cdot t \\ z = z_0 + c \cdot t \end{cases} \quad (-\infty < t < \infty) \quad (2)$$

где  $T = [x_0, y_0, z_0]^T$  – точка на линии,  $t$  – параметр, определяющий расстояние от камеры до объекта, а  $r = [a, b, c]^T$  – вектор с тем же направлением, что и линия  $l_p$ .

Как упомянуто выше, камера установлена таким образом, что центр камеры расположен в точке  $T = [0, 0, H]^T$  с углом наклона. Следовательно,  $T$  является точкой на  $l_p$  и, следовательно, остается найти только  $r$  и, добавив ограничение  $\|r\| = 1$ , полученное решение будет уникальным.

Можно получить реальные углы смещения объекта относительно центра камеры  $T$ :  $\theta' = \theta + \beta_y$ ;  $\varphi' = \varphi + \beta_x$ , где  $\beta_y, \beta_x$  – углы отклонения о главной оптической оси камеры.

$$\text{Тогда } r = R_x(\theta')R_z(\varphi')rs$$

Если  $r$  известно, можно легко подставить переменные, выведенные выше, в систему уравнений и тем самым получить  $l_p$ . Теперь предположение, что объект стоит на полу, пригодится, поскольку это означает, что точка в реальном мире имеет  $Z$ -координату, равную нулю. Это, в свою очередь, означает, что можно найти значение параметра  $t$ , а затем и координаты  $X$  и  $Y$ . Это один из способов получения исходного трехмерного положения точки, спроецированной на плоскость изображения, если хотя бы одна из ее исходных координат (в данном случае  $Z = 0$ ) известна с самого начала [10].

$$t = -\frac{z_0}{c} \quad (3)$$

$$X = x_0 + at \quad (4)$$

$$Y = y_0 + bt \quad (5)$$

### 2.3 Комплексирование данных и межкамерный трекинг

На данном этапе алгоритм имеет текущие координаты на плане помещения для каждого объекта с каждой камеры. Они получены с помощью алгоритма проецирования и умножения на матрицу смещения соответствующей камеры относительно начала системы координат связанной с помещением. Пусть  $\{X_{ij}, Y_{ij}\}$  – координаты проекции  $i$ -ого объекта с  $j$ -ой камеры. Для определения отношения объектов  $i$  и  $k$  с камеры  $j$  и  $h$  соответственно используется неравенство:

$$\sqrt{(X_{ij} - X_{kh})^2 + (Y_{ij} - Y_{kh})^2} < \rho \quad (6)$$

где  $\rho$  – критическое расстояние. Если неравенство выполняется, то это один и тот же объект, т.е.  $i \equiv k$  и тогда в качестве его координат на плане помещения берём среднее арифметическое координат этого объекта с каждой камеры:

$$X_i = \frac{\sum_{j=1}^n X_{ij}}{n} \quad (7)$$

где  $n$  – количество камер, для которых условие комплексирования было выполнено. Иначе эти два объекта считаются разными и их координаты не пересчитываются.

После комплексирования координат они в качестве измеренных состояний объектов отправляются в классический трекер, который, таким образом, работает в контексте плана помещения.

### 3. Экспериментальные исследования

Экспериментальное исследование заключалось в тестировании алгоритма проецирования и состояло из двух частей. В обоих экспериментах в качестве детектора использовалась нейросетевая модель YOLOv3 – как универсальный детектор и классификатор объектов.

В качестве камеры используем IP-камеру: Digma DiVision-200 [11], имеющую:

- Фокусное расстояние – 2,8 мм
- Горизонтальный угол обзора – 65 градусов
- Вертикальный угол обзора – 52 градуса

За позицию объекта в кадре считали центральную точку нижней грани прямоугольника, описывающего объект.

В качестве объекта в эксперименте выступает фигура человека.

В первой части алгоритм тестировался по изображениям с разных точек зрения, имитируя камеры, расположенные в разных частях помещения. План помещения представлен на рисунке 6.

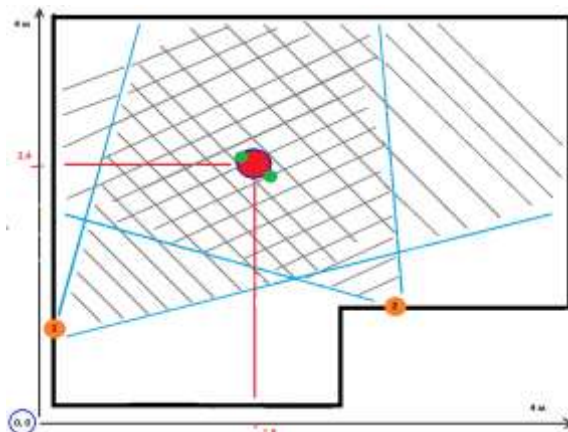


Рисунок 6. План помещения.

Положение №1 –  $T=[0; 0.7; 2.1]^T$ ,  $\theta = -24$ ,  $\varphi=-43$

Положение №2 –  $T=[2.9; 0; 2.1]^T$ ,  $\theta = -20$ ,  $\varphi=-35$

Полученные координаты от программы при анализе первого изображения (рисунок 7а)

$X^*=1.7$  м,  $Y^*=2.5$  м

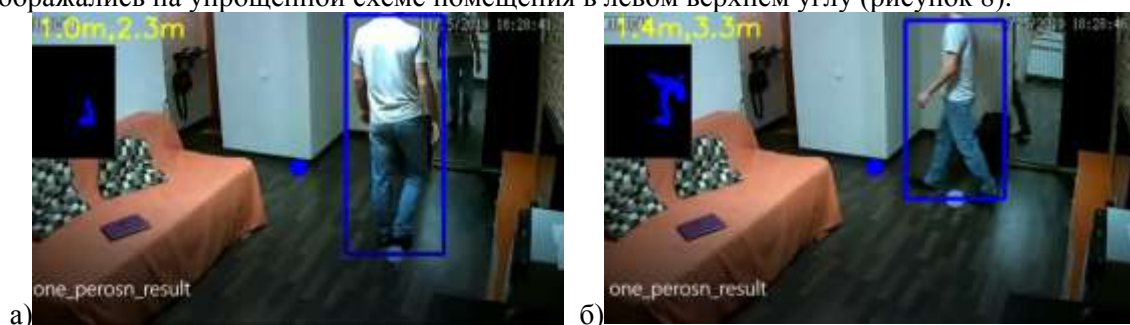
Полученные координаты от программы при анализе второго изображения (рисунок 7б)

$X^*=2$  м,  $Y^*=2.3$  м



Рисунок 7. Анализируемые изображения.

Во второй части исследований алгоритм тестировался в реальном времени с той же камеры с одной статической позиции. Объект перемещался по помещению, а результаты алгоритма отображались на упрощенной схеме помещения в левом верхнем углу (рисунок 8).



**Рисунок 8.** Результаты программы построения траектории одного объекта.

Из результатов видно, что общее направление траектории, представленной точками с каждого кадра на упрощенной схеме, верное. Но также заметен разброс точек, что говорит о погрешностях проецирования.

Максимальная величина отклонения от реального положения составила приблизительно 0.5 метров (для конкретной системы помещение-камера). Что может служить оценкой размера квадрата, в границах которого возможна коллизия.

В дальнейшем разработанный метод планируется протестировать на системе видеонаблюдения, состоящей из двух камер, расположенных в разных помещениях, а также с несколькими объектами наблюдения.

#### 4. Заключение

В статье рассмотрены методы детектирования и отслеживания объектов в видеопотоке, рассмотрены подходы по межкамерному трекингу объектов. Реализован и протестирован метод обратной проекции для проецирования изображения объекта на схему помещения. Результаты исследований показали, что описанный подход применим для реализации полноценной системы отслеживания и анализа траектории объекта внутри помещения, а также возможности расширения для конфигурации, включающую систему из нескольких камер размещенных по всему зданию.

#### 5. Благодарности

Работа выполнена в рамках государственного задания по теме FSSS-2020-0017 при частичной поддержке РФФИ: проект № 17-29-03112 офи\_м и проект № 19-29-01235 мк».

#### 6. Литература

- [1] Алфимцев, А.Н. Метод обнаружения объекта в видеопотоке в реальном времени / А.Н. Алфимцев, И.И. Лычков // Вестник ТГТУ, Тверь. – 2011. – Т. 17, № 1. – С. 44-55
- [2] Трекинг (компьютерная графика) [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/?oldid=67077116> (09.11.2019).
- [3] DeepSORT: Deep Learning to Track Custom Objects in a Video [Electronic resource]. – Access mode: <https://nanonets.com/blog/object-tracking-deepsort/> (10.11.2019).
- [4] Multiple Object Tracking using Person Re-identification [Electronic resource]. – Access mode: <https://medium.com/datadriveninvestor/multiple-object-tracking-using-person-re-identification-f9b7360cda1a> (11.11.2019).
- [5] Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3 [Electronic resource]. – Access mode: [https://medium.com/@jonathan\\_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088](https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088) (26.10.2019).
- [6] Liu, W. SSD: single shot multibox detector / W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed // CoRR, 2015. – N. 1512.02325.

- [7] Ren, S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren, K. He, R. Girshick, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 39(6). – P. 1137-1149.
- [8] Отслеживание объектов в видеопотоке. Методы построения траекторий [Электронный ресурс]. – Режим доступа: <https://macroscop.com/o-kompanii/publikaczii/2015/otslezhivanie-obektov-v-videopotoke.-metodyi-postroeniya-traektorij.html> (05.10.2019).
- [9] Vester, J. Estimating the Height of an Unknown Object in a 2D image // Master of Science Thesis – Stockholm, Sweden, 2012. – P. 6-45.
- [10] Adams, R. Calculus - A Complete Course, Sixth Edition, Pearson - Addison-Wesley, Toronto, 2006. – P. 205-211.
- [11] Вопросы и ответы по IP камерам Digma 100/200 [Электронный ресурс]. – Режим доступа: <http://ddxinfo.ru/283> (3.09.2019).

## Development of a service for tracking the trajectory of the object when moving in room using multiple cameras

N.S. Skladnev<sup>1</sup>, P.Y. Yakimov<sup>1,2</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

<sup>2</sup>Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

**Abstract.** The trajectory of people moving in the context of the room is very important both for ensuring a security of a closed territory, and for social or marketing research. The main goal of this scientific research is to develop an intelligent video tracking system for an object indoors, characterized by a minimum number of errors when switching between cameras. To achieve this goal, a method of inter-camera tracking was developed using an algorithm for positioning an object on the floor plan, its main steps were considered: localizing the object in the image, projecting the object on the floor plan and combining data from several cameras. Also, experimental studies of the positioning algorithm of an object on the plan were conducted.