

# Распознавание перекрытия объектов для повышения точности ToF-трекинга в системах дополненной реальности

М.А. Голова<sup>1</sup>, М.И. Бояркин<sup>2</sup>, К.В. Быченков<sup>3</sup>, А.В. Никоноров<sup>1,4</sup>

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

<sup>2</sup>ООО «Детектор», ул. Мичурина, 23, оф. 433, Самара, Россия, 443110

<sup>3</sup>ООО «Медекс», пр. Ленина, 3, оф. 46, Самара, Россия, 443110

<sup>4</sup>Институт систем обработки изображений РАН – филиал ФНИЦ «Кристаллография и фотоника» РАН, ул. Молодогвардейская, 151, Самара, Россия, 443001

**Аннотация.** Системы трекинга решают большое количество современных задач. В области дополненной реальности трекинг часто используется для определения позиции и ориентации объекта наблюдения. В статье рассматривается проблема обработки перекрытий объекта наблюдения при работе TimeofFlight (ToF) камеры в задачах трекинга. Такая обработка препятствий при работе ToF камеры обеспечивает значительное увеличение точности. В работе рассмотрены основные понятия данной тематики, существующие решения поставленной задачи. Предложен алгоритм решения задачи распознавания перекрытия для повышения точности ToF-трекинга.

## 1. Введение

Дополненная реальность (augmentedreality, AR) – это ёмкий термин, который относится к большинству проектов, направленных на внедрение в реальную жизнь, в трехмерное поле восприятия человека какой-либо виртуальной информации [1]. Как правило, дополнение происходит в видеопотоке в режиме реального времени. Результат этого процесса можно наблюдать с помощью современных гаджетов, таких как очки дополненной реальности, планшеты и смартфоны с функцией AR и др.

В системах дополненной реальности для решения задач комбинирования, взаимодействия и отображения объектов используют трекинг. Трекингом называют процедуру определения местоположения движущегося объекта. Местоположение определяется путем анализа кадров видеопоследовательности и выделения координат расположения и углов ориентации объекта [2]. Для данного процесса важна высокая точность результата, так как неверные выходные данные трекинга влияют на корректность работы всей системы дополненной реальности.

В рамках данной работы нами была разработана система дополненной реальности, включающая в себя трекинг, предназначенная для работы в медицинской области. Система устанавливается в хирургической операционной и в режиме реального времени позволяет хирургу видеть модель пациента, сформированную на основе предоперационной КТ (компьютерной томографии). В процессе работы системы в реальном времени происходит

наложение виртуальных моделей внутренних органов и патологий на видеопоток с учетом их реального трехмерного положения. Обработанный таким образом видеопоток позволяет медицинскому персоналу легко анализировать положение объектов интереса относительно пациента и окружения без лишних хирургических разрезов. Это позволяет уменьшить время операции и снизить операционные риски.

Пациент находится под наркозом и двигаться самостоятельно во время операции не может, однако, хирургический персонал может незначительно изменить его положение во время операции для медицинских целей. Если система не будет учитывать эти изменения, то виртуальные данные о положении пациента не будут соответствовать действительности. Использование ToF-камеры позволяет решить данную проблему. ToF-камера располагается непосредственно над пациентом, система анализирует облако точек в реальном времени, при помощи алгоритма 3-х мерной регистрации определяет актуальное положение пациента и обновляет виртуальные данные о положении объектов интереса.

Важно, чтобы в данной ситуации между работающей ToF-камерой и пациентом не было никаких препятствий, так как в этом случае генерируемое облако точек будет некорректным и алгоритм регистрации отработает с неверным результатом. На основе наблюдений при тестировании системы было установлено что ситуация перекрытия возникает довольно часто, над пациентом перемещается и передается хирургическое оборудование, хирургический персонал выполняет различные манипуляции и т.д. Проблему можно решить если вовремя распознать что произошло перекрытие и внести соответствующие коррективы в работу системы в части регистрации.

В данной статье рассматривается решение задачи распознавания перекрытия ToF-камеры для повышения стабильности трекинга при помощи ToF-камеры.

## 2. Система трекинга

Сложно представить полноценную систему дополненной реальности без трекинга, так как он позволяет отслеживать положение и ориентацию объекта интереса в каждый момент времени. На рисунке 1 представлена типовая структура системы трекинга[3], в которой можно выделить следующие подзадачи:

- выделение признаков объекта;
- математическое представление объекта;
- сопровождение объекта;
- управление объектами:
  - захват;
  - сопровождение;
  - освобождение;
- постобработка информации.

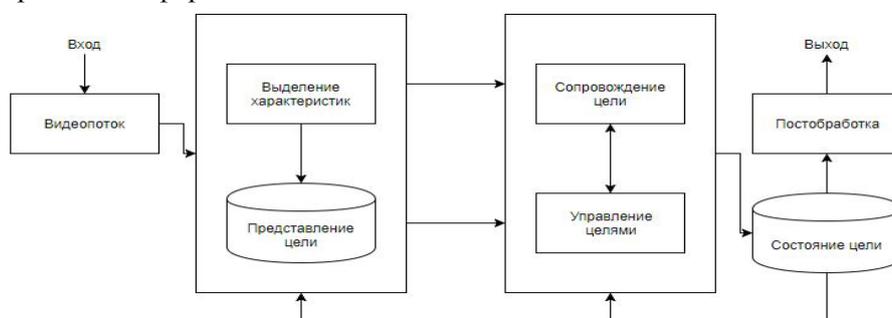


Рисунок 1. Типовая структура системы трекинга.

## 3. Использование ToF-камеры в задачах трекинга

В системах трекинга, как правило, используются одна или более камер. При использовании нескольких камер, можно получить более точные результаты, чем с одной камерой, но с

большими вычислительными затратами [4]. В обоих случаях необходимо извлекать из кадров информацию об объектах сцены и их движениях. Для этого можно использовать методы, основанные на точечных особенностях объектов [5]. Съемка, произведенная с помощью нескольких камер или с использованием дополнительных сенсоров, позволяет намного проще получить карту глубин сцены.

Карта глубин представляет собой двумерное одноканальное изображение, содержащее информацию о расстоянии от плоскости сенсора до объектов сцены [6]. Она используется для решения задач обработки видеопотока, чаще всего ее используют для задач сегментации и трекинга.

Рассмотрим получение карты глубин с помощью ToF-камеры (time-of-flight). Она измеряет время пролета испускаемого светового сигнала и отраженного каждой точкой поверхности получаемого изображения.

Выделяются три основных метода измерения дальности положения точки:

- pulsed modulation – импульсная модуляция;
- continuous modulation – постоянная модуляция;
- range gated imagers – затворная технология.

Первый метод является самым простым. Камера испускает импульс, и в каждой точке матрицы измеряется время его возвращения. Для реализации данного метода необходимо использовать оборудование с высокой точностью детектирования времени срабатывания.

В камерах второго типа излучатель испускает модулированную волну, а приемник находит максимум корреляции от полученной волны. Так определяется время, потраченное сигналом на отражение от объекта и путь до приемника. По формуле (1) считается дистанция до объекта  $d$ :

$$d = \frac{c}{4\pi w} \phi, \quad (1)$$

где  $c$  - скорость света;  $\phi$  - фазовый угол между излучением и отражением;  $w$  - модулирующая частота.

Рассмотрим третий метод. Перед матрицей стоит затвор. В момент времени  $0$  открывается затвор и освещается сцена, в момент времени  $t$  затвор закрывается. Объекты, которые расположены дальше  $d$  видны не будут:

$$d = \frac{t}{2c}, \quad (2)$$

где  $c$  - скорость света;  $t$  - время освещения сцены.

Точка поверхности, которая расположена вплотную к камере будет освещаться все время  $t$ , и иметь яркость  $I$ . Любая точка экспозиции имеет яркость в пределах  $[0, I]$ , и чем ярче точка, тем ближе она находится. В реальных системах необходимо добавить время закрытия затвора, поведение матрицы, неидеальный источник света и так далее.

На рынке представлено множество ToF-камер, например: Panasonic D-imager, PrimeSense, AsusXtion, Kinect. Более распространенным и востребованным является Kinect.

Kinect фиксирует глубину и цвет сцены одновременно с частотой 30 кадров в секунду. Датчик состоит из сенсора глубины, цветной камеры и микрофонной решетки. Программное обеспечение KinectSDK позволяет осуществлять отслеживание объекта в трехмерном пространстве, определять мимику человека и другое. Датчик глубины – инфракрасный проектор и монохромная CMOS-матрица, благодаря этому Kinect может получать трехмерное изображение при любом естественном освещении. Для работы в разработанной системе был выбран KinectforWindowsv2.

#### 4. Проблема перекрытия в задачах трекинга

Во время трекинга в системах дополненной реальности отслеживаемый объект может быть перекрыт другим объектом или объектами. Система должна распознавать эти ситуации, так как отслеживание положения пациента необходимо продолжать в обычном режиме. Перекрытия объекта наблюдения оказывают большое влияние на точность трекинга. Был проведен анализ

существующих алгоритмов решения проблемы обработки перекрытия объекта наблюдения в задачах трекинга. Ниже представлены самые типовые и известные решения данной проблемы.

#### 4.1. Адаптация к изменениям объекта

Данный алгоритм подстраивает модель к медленным изменениям объекта посредством обучения во время работы [7]. Он использует три компонента-распределения:

- S - стабильный;
- W - переходный;
- L - выбросы.

Стабильный компонент S обучается весь период работы, переходный W – за два кадра. Перекрытия характеризуются компонентом выбросов, который имеет равномерное распределение. Стабильный и переходный компоненты имеют распределение Гаусса. Обучение модели происходит с помощью EM-алгоритма, который можно подстроить под работу в реальном времени.

Существует подход, определяющий перекрытие при превышении установленного порога. Данное решение работает только в случае совпадения статистических параметров перекрытия с ожидаемыми. В реальности такие условия встречаются редко.

#### 4.2. Блочный контурный трекинг

В данном случае перекрытие анализируется с помощью сравнения параметров движения между объектом и блоками изображения, которые не могут быть скомпенсированы последующими кадрами [8]. Объект представляется маской, которая получается в несколько шагов: выполнение многозначной сегментации по 4 направлениям и поиск области одинаково движущимися сегментами.

Алгоритм определяет блоки кадра, которые принадлежат объекту, его границе и заднему фону. После этого работа ведется с этими блоками. Путем аффинных преобразований к начальным блокам, определяется движение между кадрами.

В данном алгоритме ситуации перекрытия и освобождения от перекрытия рассматриваются, как взаимнообратные. Алгоритм определяет движение и если какая-то область прекратила его, то эту область можно считать перекрытой. Если данная область возобновляет свое движение, то перекрытия нет. Путем кластеризации векторов движения областей объекта и вычислением расстояния от вектора движения рассматриваемой области до центра этого кластера можно определить возобновление движения перекрытого объекта [9].

#### 4.3. Отслеживание многозначного шаблона

Данный алгоритм использует шаблоны для представления объекта [10]. Шаблон подстраивается под изменение объекта с помощью сглаживания выделенных признаков во времени, используя фильтр Калмана для каждого пикселя кадра. Алгоритм отслеживает объекты в реальном времени с частичными перекрытиями и изменениями в освещенности.

Шаблон является вектором признаков для каждой точки изображения. Все векторы обрабатываются независимо друг от друга, а их вероятность считается с помощью формулы Байеса. С помощью гауссова шума для заданного математического ожидания получаем предсказание для вектора признаков в следующий момент времени, но оно становится неверным при перекрытии. Можно уменьшить влияние перекрытия путем замены в выражении распределения Гаусса корня нормой ошибки измерения. Функция распределения содержит в себе функцию Хубера для обработки ошибки измерения. Если ошибка превышает заданный порог, то текущее состояние объекта не подчиняется распределению Гаусса, и в формуле распределения квадратичная норма заменяется на линейную.

Пиксель изображения считается перекрытым, если ошибка превышает порог. От количества перекрытых пикселей зависит включение режима перекрытия и остановки адаптации шаблона. Это происходит, когда количество перекрытых пикселей превышает заданное число.

Длительное перекрытие может изменить шаблон до такой степени, что его нельзя будет соотнести с уже не перекрытым объектом.

### 5. Описание алгоритма распознавания перекрытия в разработанной системе

Во время проведенного выше анализа было установлено, что не существует универсального решения проблемы обработки перекрытия объекта наблюдения в задачах трекинга. Все существующие решения опираются на конкретный алгоритм трекинга, который используется в системе.

#### 5.1. Анизотропный итеративный алгоритм ближайших точек

Классическим решением задачи сведения к минимуму разницы между двумя облаками точек является итеративный алгоритм ближайших точек (ICP) [11]. Данный алгоритм подразумевает, что облака точек наблюдаются с нулевым математическим ожиданием и изотропным гауссовым шумом. В случае получения облака точек с помощью ToF-камеры, шум локализации точек не является изотропным. По этой причине для совмещения облака точек и модели пациента в разработанной системе был выбран алгоритм A-ICP (anisotropic iterative closest point).

A-ICP является модификацией алгоритма ICP для возможности учета анизотропного шума локализации точек [12]. Локализация ошибки в каждой точке  $\vec{p} \in X \cup Y$  распределена по нормальному закону с нулевым средним и ковариационной матрицей  $K_{\vec{p}} = V_{\vec{p}} S_{\vec{p}}^2 V_{\vec{p}}'$ . Учитывая эти матрицы для каждой точки, алгоритм ведет поиск матрицы вращения  $R$  и вектора перемещения  $\vec{t}$  модели таким образом, чтобы взвешенная метрика ошибки  $FRE_{weighted}^2(R, \vec{t})$  сводилась к минимуму:

$$FRE_{weighted}^2(R, \vec{t}) = \sum_{i=1}^{N_x} \|W_i(R\vec{x}_i + \vec{t} - \vec{y}_{idx(i)})\|_2^2, \tag{3}$$

где  $R$  - матрица вращения;  $\vec{t}$  - вектор перемещения;  $W_i$  - функция матрицы вращения  $R$ ;  $idx(i)$  - индекс точки  $\vec{y}_{idx(i)} = C_{new}(R\vec{x}_i + \vec{t}, Y)$ ;

$$W_i = w(RK_{\vec{x}_i}R' + K_{\vec{z}_i})^{-\frac{1}{2}}, \tag{4}$$

где  $w$  - переменная для нормализации;  $(\vec{x}_i, \vec{z}_i)$  - точки соответствия в совмещаемых моделях;

Рассмотрим работу алгоритма A-ICP:

1. Инициализация переменных:

- a.  $k = 1$  - номер итерации;
- b.  $FRE_{weighted}^0 = \infty$  - нулевая метрика ошибки;
- c.  $R^0 = I$  - нулевая матрица вращения;
- d.  $\vec{t}^0 = \vec{0}$  - нулевой вектор перемещения;
- a.  $K_{\vec{x}_i}^0 = K_{\vec{x}_i}$  - нулевая ковариационная матрица;

2. Вычисление текущих соответствующих точек  $Z^k = \{\vec{z}_i^k\}$ ,  $i = 1, \dots, N_x$ . И  $\vec{z}_i^k \in Y$ , как ближайшую точку к  $\vec{x}_i \in X^k$  в соответствии с мерой расстояния:

$$idx(i)^k = \arg \min_j d_{new}(\vec{x}_i^k, \vec{y}_j); \tag{5}$$

$$\vec{z}_i^k = \vec{y}_{idx(i)^k}. \tag{6}$$

Для этого необходимо вычислить текущую ковариационную матрицу  $K_{\vec{x}_i}^k$  по формуле (7), текущую кроссковариационную матрицу  $K_{ij}^k$  по формуле (8), весовую матрицу  $W_{ij}^{k-1}$  по

формуле (9), текущее расстояние  $d_{new}(\vec{x}_i^k, \vec{y}_j)$  для каждого преобразования точки  $\vec{x}_i^k \in X^k$  и  $\vec{y}_j \in Y$  по формуле (10):

$$K_{\vec{x}_i}^k = R^{k-1} K_{\vec{x}_i}^{k-1} (R^{k-1})'; \tag{7}$$

$$K_{ij}^k = K_{\vec{x}_i}^k + K_{\vec{y}_j}; \tag{8}$$

$$W_{ij}^{k-1} = w(K_{ij}^k)^{-\frac{1}{2}}; \tag{9}$$

$$d_{new}(\vec{x}_i^k, \vec{y}_j) = W_{ij}^{k-1} (\vec{x}_i^k - \vec{y}_j). \tag{10}$$

3. Вычисление матрицы вращения  $R^k$ , вектора перемещения  $\vec{t}^k$  по формуле (6). И взвешенной метрики ошибки  $FRE_{weighted}^k$  по формуле (7):

$$(R^k, \vec{t}^k) = \arg \min_{R, \vec{t}} \sum_{i=1}^{N_x} \|W_i^k (R\vec{x}_i^k + \vec{t} - \vec{z}_i)\|_2^2; \tag{11}$$

$$FRE_{weighted}^k = \sqrt{\sum_{i=1}^{N_x} \|W_i^k (R\vec{x}_i^k + \vec{t} - \vec{z}_i)\|_2^2}. \tag{12}$$

4. Применение преобразования, вычисленного на предыдущем шаге, к  $X^k$  и получение преобразованного набора точек  $X^{k+1}$  для следующей итерации  $k+1$ :  $x_i^{k+1} = R^k x_i^k \vec{t}^k$ .

5. Если формула (7) верна или достигнуто заданное максимальное число итераций, работа алгоритма завершается. Иначе, к  $k$  прибавляется единица и совершается переход ко 2 шагу

$$|FRE_{weighted,k} - FRE_{weighted,k-1}| < \varepsilon \tag{13}$$

### 5.2. Алгоритм решения задачи распознавания перекрытия

Опираясь, на рассмотренный алгоритм A-ICP, решаем поставленную задачу распознавания перекрытия объекта наблюдения. FiducialRegistrationError(FRE) – является выходным значением алгоритма. Анализируя это значение можно определить момент перекрытия объекта наблюдения. Данное значение будет высоким, если объект быстро движется или его что-то перекрыло.

#### 5.2.1. Анализ работы системы

Был проведен анализ значений  $FRE$  для трех случаев:

1. объект не движется и не перекрывается;
2. объект движется, но не перекрывается;
3. объект не движется, но перекрывается.

Данные были собраны с использованием торса манекена, который является прототипом пациента. В лабораторных условиях были воспроизведены условия операционной. Манекен был расположен на столе и непосредственно над ним Kinect на штативе. На рисунке 2 продемонстрированы данные условия. Работу данной системы можно увидеть на рисунках 3 и 4, на которых представлен интерфейс системы во время эксперимента.

Средние значения показателя качества  $FRE$  до внедрения алгоритма распознавания перекрытия представлены на рисунке 5.



Рисунок 2. Экспериментальные условия.

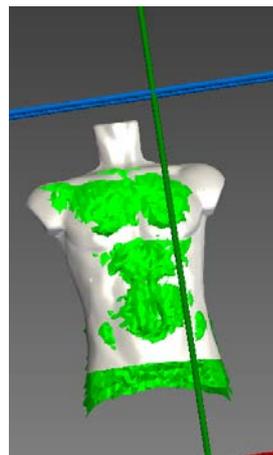


Рисунок 3. Совмещение облака точек ToF-камеры с моделью торса манекена.

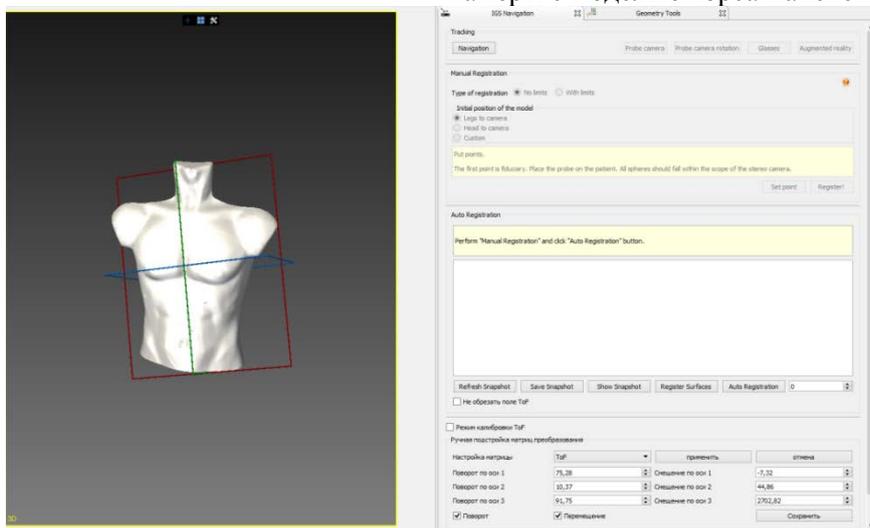


Рисунок 4. Интерфейс разработанной системы.



Рисунок 5. Средние значения показателя качества FRE до внедрения алгоритма распознавания перекрытия.

На основе графика можно сделать следующие выводы:

- Если объект находится в неподвижном состоянии и его не перекрывает ничего, то *FRE* имеет довольно низкое значение, так же не наблюдаются резкие скачки во время работы алгоритма.
- Если объект перемещается, то *FRE* возрастает, можно наблюдать небольшие скачки во время работы алгоритма.

- Если объект перекрыт, то  $FRE$  намного выше случая с перемещением объекта. Из графика видно, что  $FRE$  растет с течением времени.

На последней итерации объект не был перекрыт, но значение  $FRE$  было выше нормы. Получается, что если убирать препятствие, то А-ICP так же выдает не верный результат.

Опираясь на эти выводы и данные, был разработан следующий алгоритм:

1. Заполнение массива  $M$  размерностью 20 значениями  $FRE$ , которые получены от неподвижной и не перекрытой модели.
2. Вычисление среднего значения  $s$  массива  $M$ .
3. Сравнение текущего  $FRE$  с  $s$ :
  - а. если  $|FRE - s| > t$ , где  $t$  - порог. То произошло перекрытие и отслеживать объект не нужно;
  - б. если  $|FRE - s| < t$ , то отслеживаем объект и обновляем массив  $M$  текущим  $FRE$ ;
4. Повторяем шаги для каждого  $FRE$ .

Производим реализацию данного алгоритма внутри нашей системы. На рисунке б продемонстрирован график  $FRE$  при работе алгоритма распознавания перекрытий.



**Рисунок 6.** Снижение значения показателя качества FRE при обнаружении перекрытий по предложенному алгоритму.

Из рисунка 6 видно, что разработанный и реализованный алгоритм выполняет свою задачу. Распознавание перекрытия объекта ведется в режиме реального времени, что позволяет повысить точность ToF-трекинга в разработанной системе. С помощью разработанного алгоритма удалось уменьшить среднее значение TRE с 55,017 до 20,186.

### 6. Благодарности

Работа выполнена при поддержке Федерального агентства научных организаций (соглашение № 007-ГЗ/Ч3363/26), гранта Президента Российской Федерации МД-2531.2017.9 и грантов РФФИ (проекты 16-47-630721 p\_a, № 17-29-03112-офи-м, № 18-07-01390-А).

### 7. Заключение

В статье рассмотрена задача трекинга в системах дополненной реальности и проблема обработки препятствий при работе ToF-камеры. Был проведен анализ существующих решений данной проблемы. Анализ показал, что существующие решения обработки препятствий подходят только для определенных алгоритмов трекинга. На данный момент в открытом доступе не представлен универсальный алгоритм. Была поставлена и решена задача разработки алгоритма распознавания перекрытия для алгоритма А-ICP, используемого в разработанной системе дополненной реальности для хирургических операций. Разработанный алгоритм был реализован и протестирован на тестовом объекте (торсманекена). Результаты этого тестирования показали, что алгоритм распознает перекрытия объекта в режиме реального

времени и уменьшает среднее значение TRE с 55,017 до 20,186. Это позволяет повысить точность и увеличить стабильность ToF-трекинга в разработанной системе.

## 8. Литература

- [1] Azuma, R.T. A Survey of Augmented Reality / R.T. Azuma // *Teleoperators and Virtual Environments*. – 1997. – Vol. 6. – P. 355-385.
- [2] Yilmaz, A. Object tracking: A survey / A. Yilmaz, O. Javed, M. Shah // *ACM Journal of Computing Surveys*. – 2006. – Vol. 38. – P. 45.
- [3] Maggio, E. Video tracking theory and practice / E. Maggio, A. Cavallaro. – Wiley, 2011.
- [4] Suau, X. Real-time head and hand tracking based on 2.5D data / X. Suau, J. Ruiz-Hidalgo, J.R. Casas // *IEEE transactions on multimedia*. – 2012. – Vol. 14(3). – P. 575-585.
- [5] Конушин, А. Слежение за точечными особенностями сцены (Pointfeaturetracking) / А.Конушин // *Компьютерная Графика и Мультимедиа Сетевой Журнал*. – 2003. – Т. 1, № 5. – [Электронный ресурс]. – Режим доступа: <http://cgm.computergraphics.ru/>.
- [6] Воронин, В.В. Восстановление карты глубины на основе комбинированной обработки многоканального изображения / В.В. Воронин, А.В. Фисунов, В.И. Марчук, И.С. Свиринов, С.П. Петров // *Современные проблемы науки и образования*. – 2014. – Т. 6.
- [7] Jepson, D. Robust Online Appearance Models for Visual Tracking / D. Jepson, D.J. Fleet, T.F. El-Maraghi // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2003. – P. 1296-1311.
- [8] Hariharakrishnan K. Fast object tracking using adaptive block matching // K. Hariharakrishnan, D. Schonfeld // *IEEE Trans. on Multimedia*, 2005. – P. 853-859.
- [9] Тассов, К.Л. Обработка перекрытий в задачах отслеживания объектов в видеопотоке / К.Л. Тассов, Д.Е. Бекасов // *Инженерный журнал: наука и инновации*. – 2013. – Вып. 6.
- [10] Nguyen, H.T. Fast occluded object tracking by a robust appearance filter / H.T. Nguyen, A.W.M. Smeulders // *IEEE Trans. on Pattern Analysis and Machine Intelligence*. – 2004. – P. 1099-1104.
- [11] Besl, P.J. A Method for Registration of 3-D Shapes / P.J. Besl, N.D. McKay // *IEEE Transactions on pattern analysis and machine intelligence*. – 1992. – Vol. 14(2). – P. 239-256.
- [12] Maier-Hein, L. Convergent Iterative Closest-Point Algorithm to Accommodate Anisotropic and Inhomogeneous Localization Error / L. Maier-Hein, A.M. Franz, T.R. dos Santos, M. Schmidt, M. Fangerau, H.P. Meinzer, J.M. Fitzpatrick // *IEEE Trans. on Pattern Analysis and Machine Intelligence*. – 2012. – Vol. 34.

# Recognition an overlap of objects to increase an accuracy of ToF-tracking in augmented reality systems

M. Golova<sup>1</sup>, M. Boyarkin<sup>2</sup>, K. Bychenkov<sup>3</sup>, A.Nikonorov<sup>1,4</sup>

<sup>1</sup>Samara National Research University, Moskovskoyeshosse, 34, Samara, Russia, 443086

<sup>2</sup>ООО "Detector", Michurina street, 23, of. 433, Samara, Russia, 443110

<sup>3</sup>ООО "Medx", Leninaave., 3, of. 46, Samara, Russia, 443110

<sup>4</sup>Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskayastreet 151, Samara, Russia, 443001

**Abstract.** The tracking systems solve the modern tasks that require the greatest accuracy of the system. Tracking determines the position and orientation of the observation object in augmented reality systems. The article provides an overview of solving the problem of an overlap of objects during the Time of Flight (ToF) camera operating time in the tracking systems. This processing an overlap of objects during the Time of Flight (ToF) camera operating time provides an increase in accuracy. The paper discusses the basic concepts of this subject and existing solutions to the problem. An algorithm for solving overlap detection problems is proposed to increase the accuracy of ToF tracking.

**Keywords:** Tracking systems, Time of Flight (ToF) camera, Augmented reality, Overlap.