электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК: 519.6

Метод отслеживания объектов с подвижной камеры

***Меньшов Н.Д.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»

Научный руководитель: Тассов К.Л., старший преподаватель
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
irudakov@bmstu.ru

Введение

В настоящее время широко используются системы, обрабатывающие данные с видеокамер. Эти системы выполняют различный анализ видео, но у всех систем есть общий этап - поиск данных в видео потоке. Используются различные методы поиска объектов, в зависимости от типа объектов и типа камеры. Данные можно искать в зависимости от характеристик ищущихся объектов, например, характер движения, параметров объектов (форма и размер), цвет и т.д. Так же методы различаются из-за характера движения камеры. Камеры бывают статичные и движущиеся. Методы, используемые для статичных камер не применимы для движущихся камер.

В данной работе рассматриваются методы отслеживания движущихся объектов на видео с подвижной камеры. Системы, использующие данный метод могут использоваться на видео регистраторах, на камерах, которые несут в руках, и в других случаях. Также данные системы могут применятся на беспилотных летательных аппаратах для контролирования транспортного потока. В настоящее время не существует открытых систем отслеживания подвижных объектов с движущейся камеры, не привязывающихся к типу объектов и системам позиционирования (GPS, Глонасс) .

Таким образом, целью данной работы является разработка метода отслеживания объектов с подвижной камеры.

1. Существующие решения

В настоящее время существуют следующие методы отслеживания объектов:

- Image Segmentation in Video Sequences: A Probabilistic Approach

Основная идея метода, изложенного в данной работе, это вычитание текущего изображения из усреднённого изображения заднего фона. Но, в отличие от простого усреднения фона, дополнительно проводится классификация каждого пикселя для получения сведений о том, как пиксель должен выглядеть. Использование классификации позволило учитывать медленно движущиеся объекты и тени, что нельзя делать в простом алгоритме усреднения фона. При полной остановке движения на видео могут происходить ошибки распознавания. Недостатком метода является статичная камера.

- Adaptive background mixture models for real-time tracking

В данной работе каждый пиксель представляет собой сумму нескольких нормальных распределений. Каждое из распределений описывает какой-то процесс (например, освещённость). В процессе вычислений определяется результирующее значение, которое наиболее точно описывает фон. Параметры для каждого из распределений постоянно обновляются.

Данный алгоритм может работать в реальном времени, при любых изменениях освещённости и погоды, а также с медленно движущимися объектами. Однако данный метод требует точного задания входных параметров для распределений и камера должна быть неподвижной.

- A Texture-Based Method for Modeling the Background and Detecting Moving Objects

В данной статье изложен алгоритм моделирующий пиксели заднего фона с помощью локальных бинарных шаблонов, которые вычисляются в радиусе от данного пикселя. Алгоритм локальных бинарных шаблонов устойчив к различному освещению и не требует нормализации входного изображения. Так же данный алгоритм способен работать в режиме реального времени. В качестве недостатка можно указать плохую работу с монотонными областями, такими как небо. Предполагается, что камера не двигается.

- Tracking Learning Detection

Этот метод используется для долгосрочного отслеживания объектов в видео потоке. В данной работе задача отслеживания объекта разбивается на три подзадачи: поиск, отслеживание и обучение. Трекер отслеживание перемещение объекта от кадра к кадру. Детектор ищет все появления объекта и корректирует трекер, если надо. При обучении происходит обнаружение ошибок детектора с последующим обновлением детектора, для избежания этих ошибок в будущем. Для обнаружения ошибок

используется P-N обучение. P - отслеживает пропущенные обнаружения, N - отслеживает ложные срабатывания.

Данный метод может работать с подвижной камерой. Недостатком метода является отслеживание только одного объекта. Так же метод не способен отслеживать неизвестные объекты. Детектор должен быть заранее обучен на поиск объекта или объект должен быть помечен обрамляющей рамкой.

- Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera

В этой работе происходит отслеживание объектов с подвижной камеры. Используя параметры камеры (например, фокусное расстояние), определяются координаты объектов относительно камеры. Отслеживание объектов происходит с помощью Венгерского алгоритма. К недостаткам данного метода можно отнести отсутствие детектора, подходящего для решения поставленной задачи.

Рассмотренные решения полностью не удовлетворяют поставленной задачи, поэтому необходимо создавать собственный метод, комбинирующий достоинства существующих систем и убирающий их недостатки.

2. Разработанные метод

Концептуально процесс отслеживания объектов представлен на диаграмме IDEF0 первого уровня в виде функциональных блоков (рис. 1):

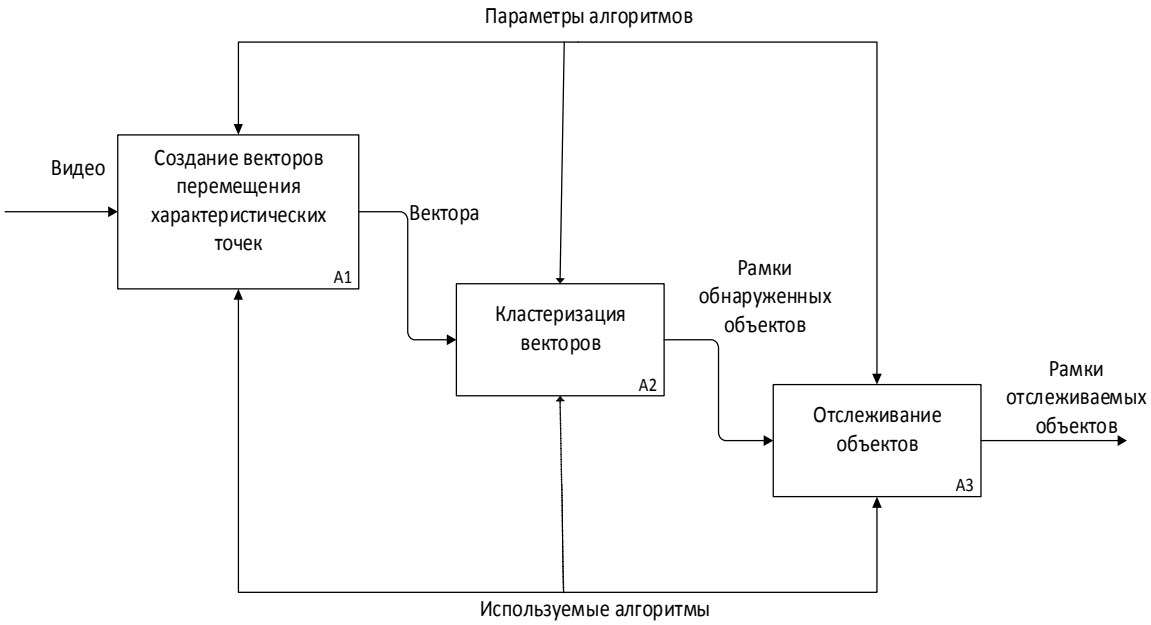


Рис. 1. Получение рамок отслеживаемых объектов

Далее приводится детальное описание разрабатываемых алгоритмов, представляющих собой блоки A1, A2 и A3, показанных на рис. 1, составляющих предлагаемый метод для отслеживания объектов.

2.1. Создание векторов перемещения характеристических точек

Создание векторов основано на использовании характеристических точек найденных с помощью детектора Харриса и определения их перемещения с помощью оптического потока методом Лукаса - Канаде. Использование данного подхода позволяет создать массив векторов, отражающих разницу между двумя изображениями. В результате работы алгоритма получается набор векторов вида:

$$\vec{V} = (x, y, v, \alpha) \quad (1)$$



Рис. 2. Генерация характеристических точек

Общая схема алгоритма создания векторов показана на рис. 3. Входными данными являются изображение Image и номер кадра index в видео потоке . Выходные данные - набор векторов vectors.

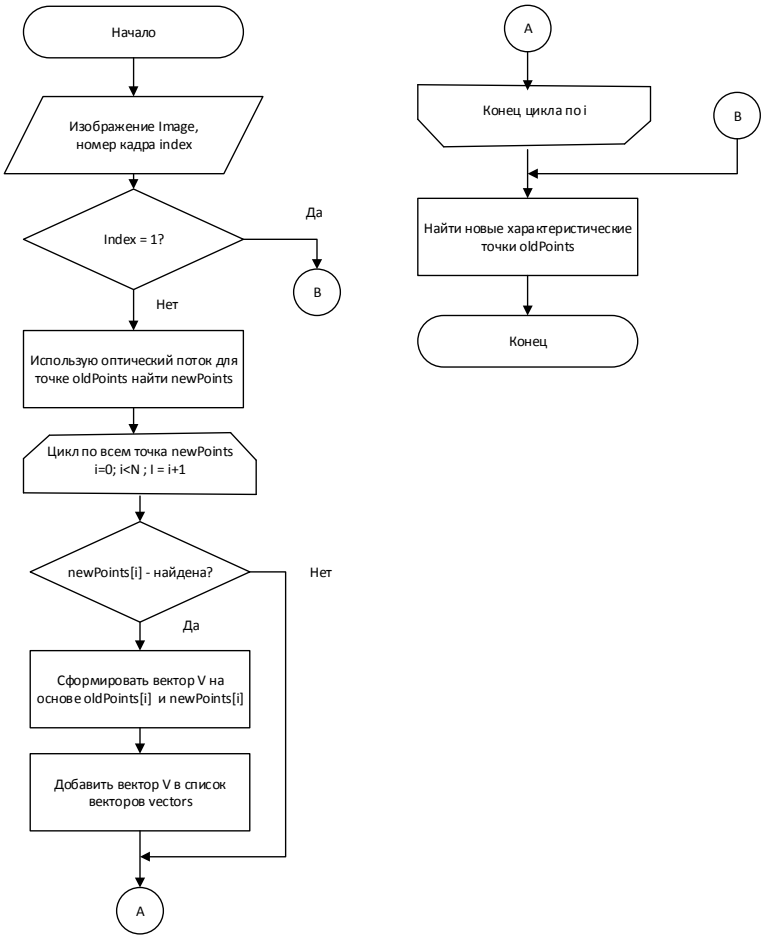


Рис. 3. Схема алгоритма нахождения векторов

Для вычисления оптического потока необходима два кадра, поэтому на первом кадре находятся характеристические точки oldPoints. На последующих кадрах, используя метод Лукаса - Канаде, ищутся положения точек из списка oldPoints предыдущего кадра на текущем кадре. Результаты записываются в список newPoints. Каждая точка из newPoints содержит её новые координаты или состояние, что точка не найдена. Затем перебираются все точки из newPoints и oldPoints для формирования списка векторов. После этого ищутся новые характеристические точки на текущем кадре. Характеристические точки ищутся заново на каждом кадре по нескольким причинам:

- Точки могут теряться от кадра к кадру;
- объекты могу поворачиваться, заслоняя старые точки;
- на видео появляются новые объекты, на которых находятся новые точки.

После формирования списка векторов происходит его нормализация.

2.2. Кластеризация векторов

После получения списка векторов необходимо с их помощью выделить объекты. Для этого можно использовать кластеризацию с неизвестным числом кластеров. Среди рассмотренных алгоритмов кластеризации был выбран алгоритм Forel. В этом алгоритм используется понятие максимального размера кластера.

На рис. 4. представлена схема алгоритма кластеризации векторов для получения обрамляющих рамок обнаруженных объектов. Входными данными являются список векторов Vectors и параметр R - максимальный размер кластера. Выходным параметром является список кластеров Clusters с координатами обрамляющих рамок.

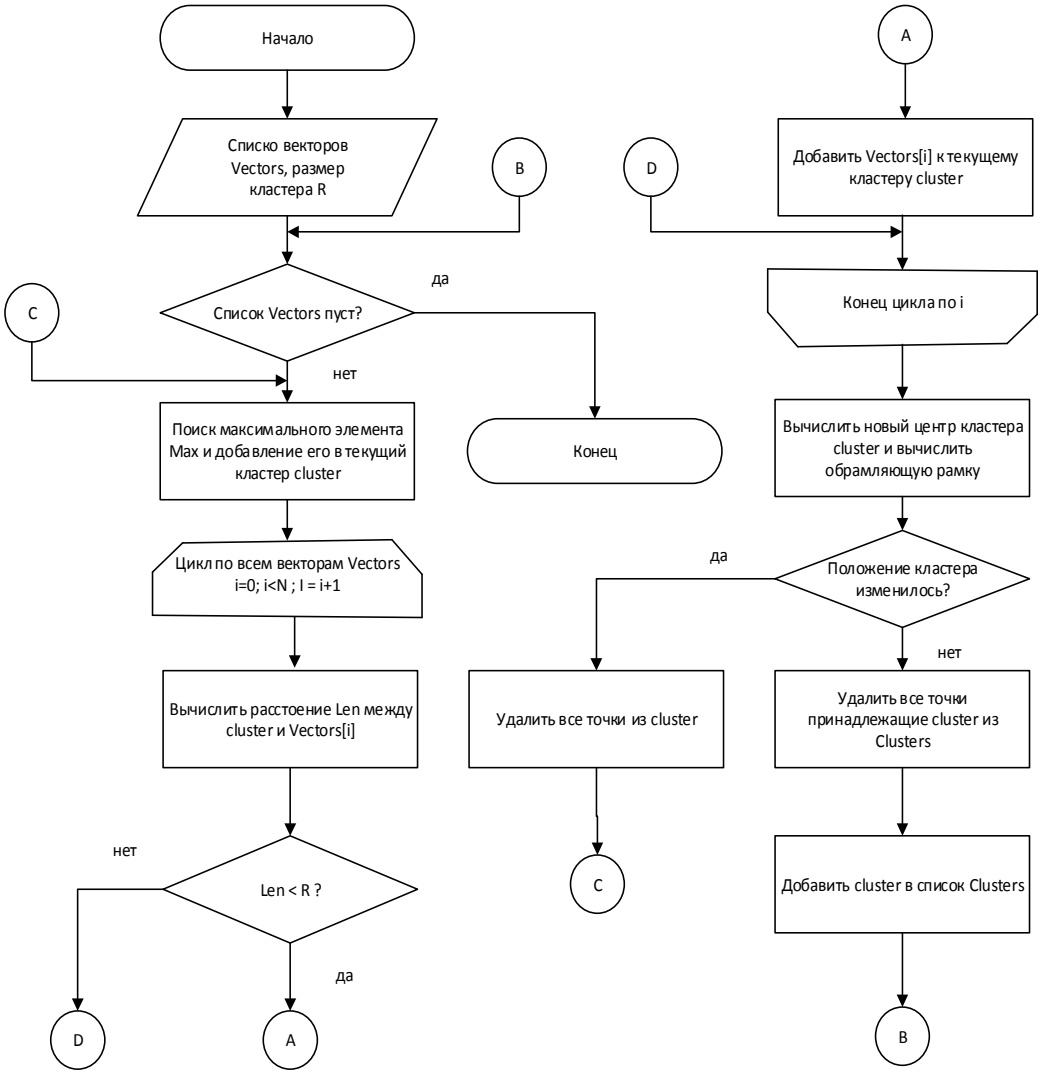


Рис. 4. Схема алгоритма кластеризации векторов

Согласно алгоритму кластеризация продолжается до тех пор пока все векторы не будут отнесены к какому либо кластеру. Кластеризация начинается с поиска максимального элемента в списке векторов. Максимальный элемент может вычисляться

как $\max = \max_i p_i$, где $i = \overline{1N}$. В качестве p_i можно взять длину вектора или его длину. Такой элемент помечается как центр кластера. При вычислении нового центра кластера так же вычисляется его обрамляющая рамка. Рамка задается как $S = \{minx, miny, maxx, maxy\}$,

где $minx = \min_i x_i$, $maxx = \max_i x_i$, $miny = \min_i y_i$, $maxy = \max_i y_i$, где $i = \overline{1N_S}$, N_S- количество точек в кластере, (x_i, y_i) - координаты вектора до нормализации. Все точки, попавшие в кластер удаляются из списка векторов, а сам кластер добавляется в список обнаруженных объектов. На рис. 5 представлен пример кластеризации векторов.

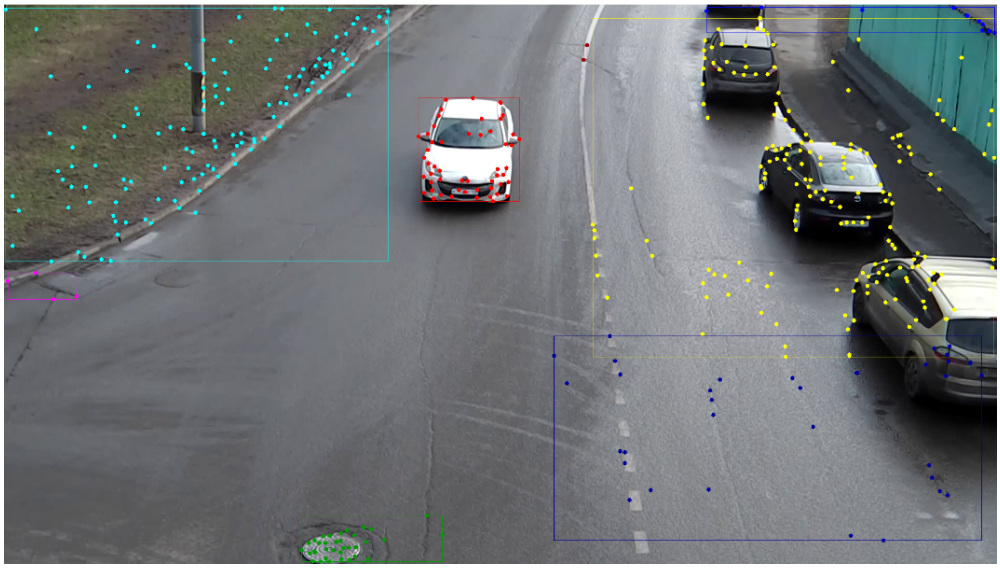


Рис. 5. Кластеризация векторов

2.3. Алгоритм отслеживания объектов

После получения списка рамок обнаруженных объектов необходимо провести их анализ для отслеживания объектов. Все рамки объектов делятся на три группы:

- Рамки, относящиеся к уже отслеживаемым объектам.
- Рамки, которые не относятся ни к одному отслеживаемому объекту.

Для отслеживания используется алгоритм, предложенный в [1]. В нём используется венгерский алгоритм для определения степени близости обнаруженных и отслеживаемых рамок.

На рис. 6 представлена схема алгоритма анализа рамок обнаруженных объектов. Входными данными являются список рамок обнаруженных объектов Detections и порог соответствия P. Выходным данным является список рамок отслеживаемых объектов.

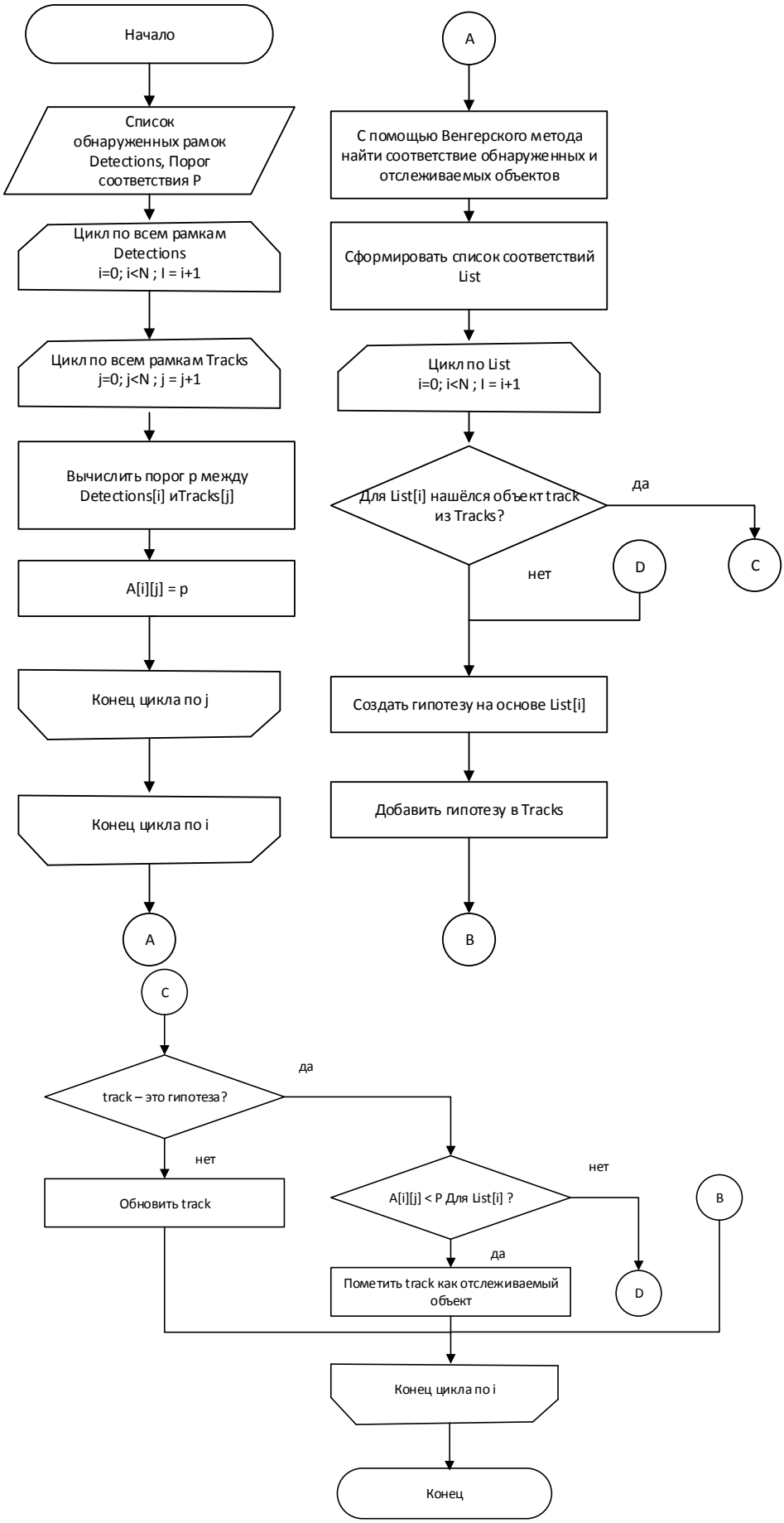


Рис. 6. Схема алгоритма отслеживания объектов

Список отслеживаемых объектов Tracks состоит из объектов 2х типов: отслеживаемые объекты и гипотезы. Гипотезой является первое появления любого обнаруженного объекта, для которого не найден отслеживаемый объект. Как видно из рисунка 2.6 сначала строится матрица стоимостей для Венгерского метода, где

$$A_{ij} = -\log \frac{\cap(Detection_i, Tracks_j)}{\cup(Detection_i, Tracks_j)}. \quad (2)$$

Если обнаруженный объект не пересекается с отслеживаемым объектом, то в A_{ij} записывается бесконечность. После этого решается задача о назначениях и формируется список List состоящий из элементов вида $\{d, t, A_{tp}\}$, где d - индекс обнаруженной рамки в Detection и t - индекс отслеживаемого объекта в Tracks (равен -1, если такой объект не найден). Все элементы из List, для которых $A_{tp} < P$, где P - порог соответствия, считаются обнаружениями отслеживаемых объектов на текущем кадре, если Tracks[t] является отслеживаемым объектом. Если Tracks[t] - гипотеза, то данная гипотеза помечается как новый отслеживаемый объект. Элементы, для который t =-1 или $A_{tp} \geq P$, помечаются как новые гипотезы. Все элементы Tracks, для которых не найдены новые обнаружения в течении нескольких кадров удаляются из списка. Выходом данного алгоритма являются все элементы Tracks, не являющиеся гипотезами.

В рамках данного раздела проводится ряд исследований, направленных на определение параметров алгоритмов обнаружения и отслеживания движущихся объектов, а также определение качественных показателей.

3. Исследовательский раздел

3.1. Методика исследования

Для того чтобы начать проводить исследования, необходимо:

- определить размер тестовой выборки;
- провести оценку параметров. Для оценки работы метода необходимо провести исследования:
  - по определению размеров кластер движущегося объекта при обнаружении;
  - по определению порога близости двух рамок при отслеживании объекта;
  - по определению начальной точки для начала кластеризации;
  - по определению формы кластера при обнаружении движения;
- сравнить результаты параметров алгоритмов;

- получить окончательные параметры для предложенного метода.

3.2. Выборка

Для проведения исследований взята выборка, состоящая из раскадровки видео в графическом формате JPEG. На каждом изображении присутствует несколько движущихся средств. Изображения взяты из базы данных алгоритма TLD.

Размер выборки обуславливается точностью определения параметров метода, а также необходимой сложностью проведения исследования. Для оценки размера выборки возможно воспользоваться законом больших чисел, записанному в форме 2-го неравенства Чебышева:

$$P\{|X - MX| \geq \varepsilon\} \leq \frac{\sigma^2}{N\varepsilon^2} \tag{3}$$

где N – размер выборки;

X – случайная величина;

ε – требуемая точность;

σ^2– дисперсия случайной величины X;

MX – математическое ожидание случайной величины X.

Критерий качества лежит в диапазоне от 0 до 1, поэтому дисперсия качества оцениваемого параметра не превосходит 0,25, т.е. σ^2 = 0,25. Достоверность метода возьмём равным 0,90, т. е. P = 0,90. Примем точность метода равной 0,1. Значит, необходимый размер выборки равен:

$$N \geq \frac{\sigma^2}{P\{|X - MX| \geq \varepsilon\}\varepsilon^2},$$
$$N \geq \frac{0{,}25}{(1 - 0{,}90) * 0{,}1^2} >= 250. \tag{4}$$

Таким образом, необходимый размер выборки равен N = 250 с ошибкой не более 10% и вероятностью 90%.

3.3. Определение размера кластер с формой гиперсфера

Для правильного детектирования объектов проводится исследование размера кластера с формой гиперсфера. Размер гиперсферы варьируется в значения $R = \overline{0{,}1}$. Такой диапазон выбран из-за того, что вектора нормализованы.

Результаты исследования показаны на графике на рис. 7 и на рис. 8.

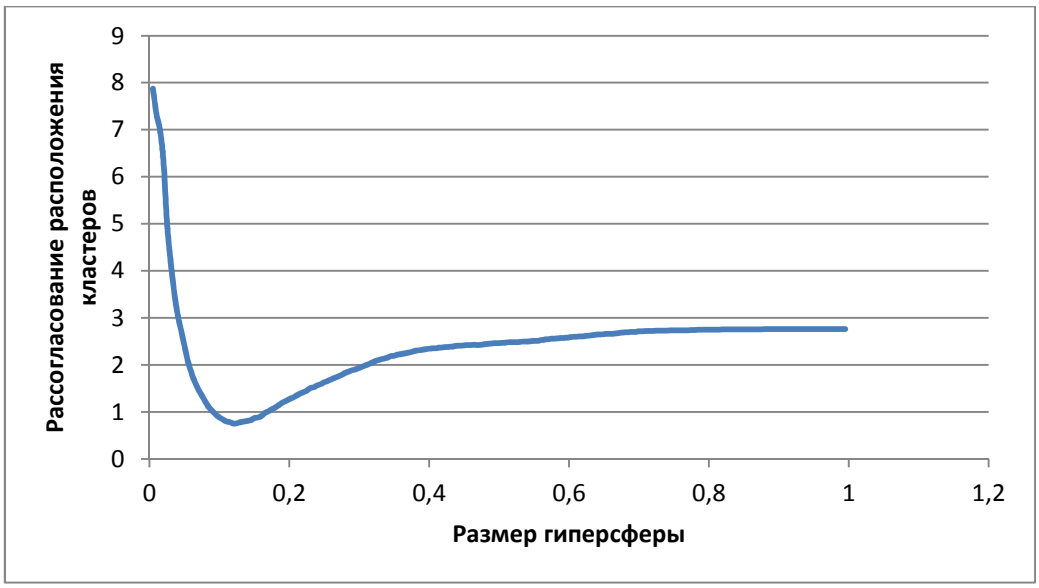


Рис. 7. Зависимость правильности расположения кластеров от размера гиперсферы

Характеристика оценки правильности расположения кластеров оценивается по формуле:

$$err(R) = \frac{\sum_t^N \frac{\sum_i^{N_t} \max_j(-log \frac{\cap (X_i Y_j)}{\cup (X_i Y_j)})}{N_t}}{N}, j = \overline{1K_t}, \quad (5)$$

где $err(R)$ - среднее максимальное рассогласование эталонных рамок и обнаруженных рамок;

N - количество кадров для исследования (250 изображений);

N_t- количество эталонных рамок на t - ом кадре;

K_t- количество обнаруженных рамок на t - ом кадре;

X_i - координаты i - ой эталонной рамки;

Y_j - координаты j - ой обнаруженной рамки.

Количество ложных срабатываний оценивается по формуле:

$$p_t = \frac{K_t - N_t}{N}, \quad (6)$$

где p_t - среднее количество ложных срабатываний на t - ом кадре.

Поведение графика на рис. 7 объясняется тем, что при малых размерах гиперсферы происходит дробление кластера на маленькие. Ни один из кластеров по размеру не подходит для эталонной рамки. При увеличении размера гиперсферы, увеличивается и размер кластеров. Рассогласование эталонных и обнаруженных рамок уменьшается. При дальнейшем увеличении размера гиперсферы, размер обнаруженных рамок начинает превосходить размер эталонных рамок. При приближении размера гиперсферы к 1 не

происходит роста рассогласования, так как почти всё изображение занимает один кластер, с которым пересекается только одна эталонная рамка. Минимальное рассогласование достигается при значении 0,125, значение рассогласования 0,759.

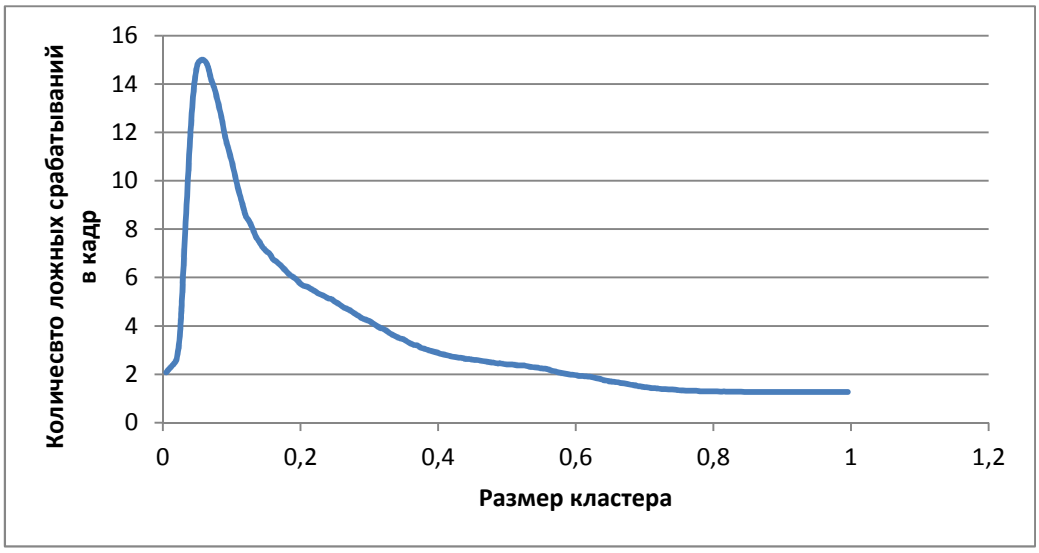


Рис. 8. Количество ложных обнаруженных объектов в один кадр

Пик на графике рис. 8 возникает из-за минимального количества векторов, из которых состоит кластер. При увеличении размера гиперсферы увеличиваются размеры кластеров, но не у всех кластеров количество векторов превосходит минимальное. При достижении размера кластера 0,45 достигается максимальное число ложных срабатываний за один кадр.

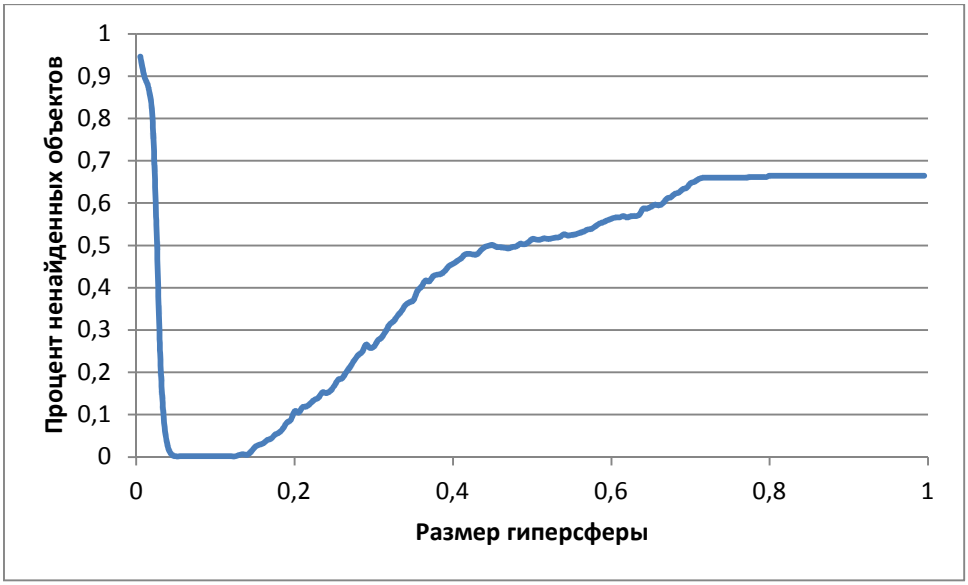


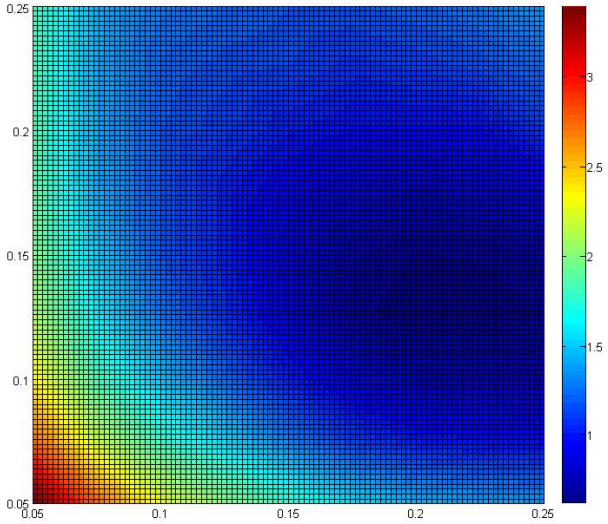
Рис. 9. Процент ненайденных эталонных объектов

В диапазоне 0,045 до 1.35 все эталонные объекты найдены. При уменьшении размера гиперсферы срабатывает ограничение на количество точек в кластере. А при увеличении объясняется поглощением одними кластерами других.

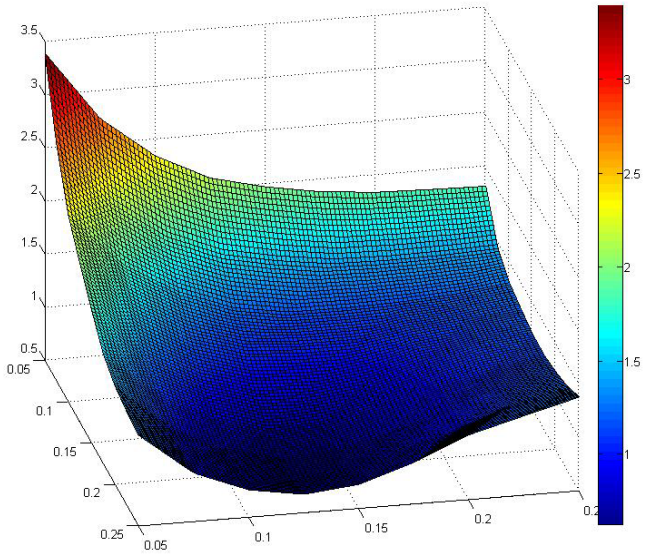
3.4. Определение размера кластер с формой прямоугольник

Для правильного детектирования объектов проводится исследование размера кластера с формой прямоугольник. Размер прямоугольника варьируется в значения $x = \overline{0, 0.5}$, $y = \overline{0, 0.5}, v = \overline{1, 5}$, $a = \overline{0, 45^{\circ}}$. Два вектора A и B находятся в одном кластере, если вектор a находится внутри прямоугольника с центром в B. При использовании такой формы кластера не надо нормализовывать вектор, только его составляющие.

Результаты одного из исследований показаны на графике на рис. 10 и 11.



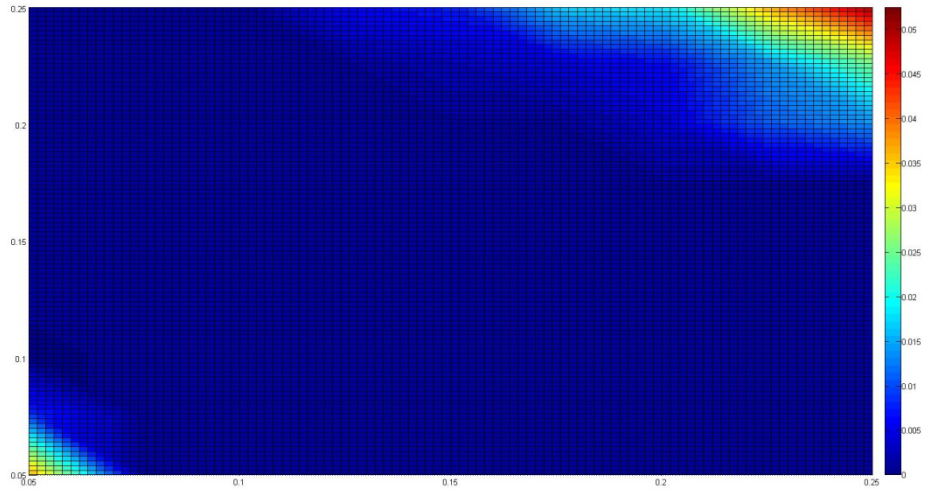
а)



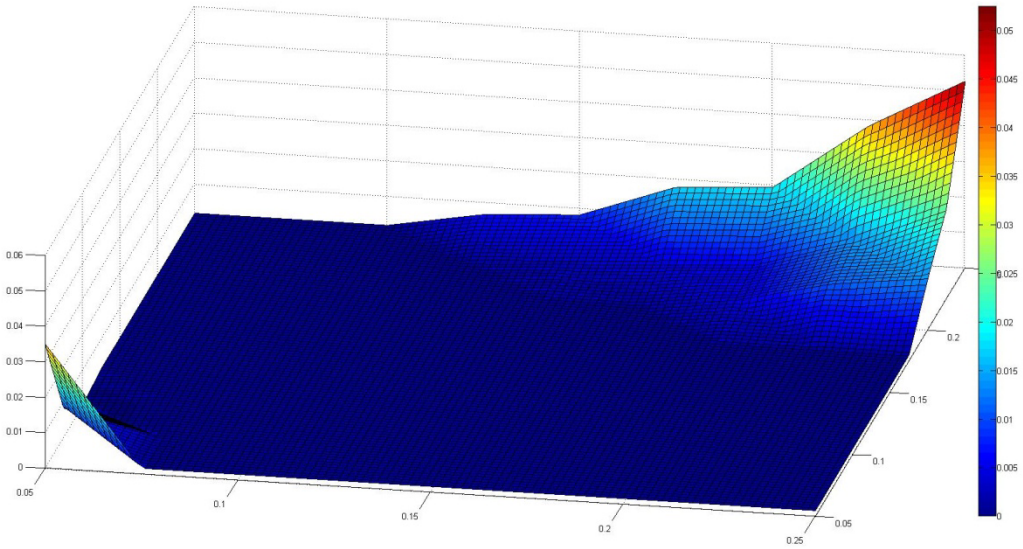
б)

Рис. 10 Зависимость правильности расположения кластеров от размеров прямоугольника

Так как результат исследования имеет пять измерений (x, y, v, a и значение рассогласования), то на графике представлен срез $v = 4,\ a = 25^{\circ}$, которым соответствует наименьшее значение рассогласования. При значения $x = 0{,}225, y = 0{,}13, v = 4,\ a = 25^{\circ}$ достигается наименьшее значение рассогласования, равное 0,62, при этом количество ненайденных объектов равно 0%. Видно. что использование в качестве формы кластера прямоугольника даёт лучший результат по сравнению с гиперсферой. Это объясняется характером движения объектов и их формой. В выборке отслеживались автомобили, которые двигались параллельно относительно горизонтальных рамок видео. Таким образом, задавая в прямоугольнике рамку для координат можно лучше выделить объект, чем в гиперсфере.



а)



б)

Рис. 11. Процент ненайденных эталонных объектов.

3.5. Определение порога соответствия при отслеживании

Для выбора порога соответствия, при котором достигается наименьший процент ненайденных объектов и значение порога соответствия будет наименьшим, проведено данное исследование. Для обнаружения объектов выбран кластеризатор с формой прямоугольника и размерами $x = 0{,}225, y = 0{,}13, v = 4,\ a = 25^{\circ}$, так как он даёт наилучший результат.

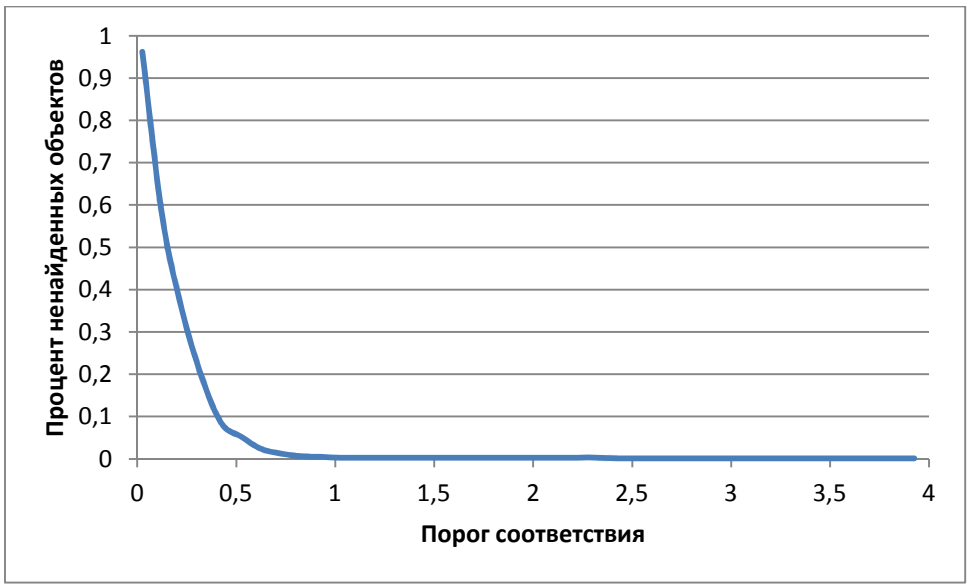


Рис. 12. Процент ненайденных эталонных объектов

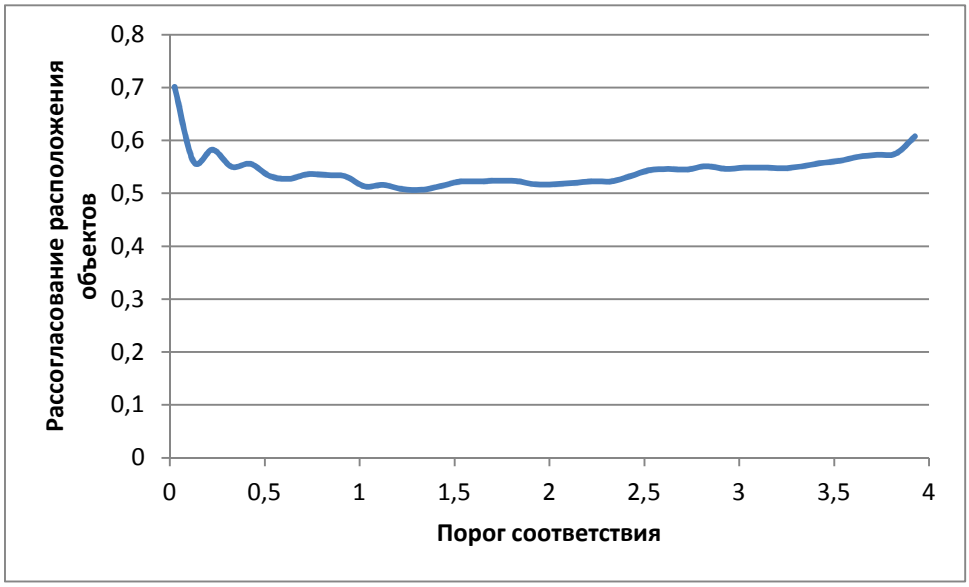


Рис. 13. Рассогласование расположения объектов в зависимости от порога соответствия

На рис. 12 видно, что существует значение порога (0,725), при котором находятся все эталонные отслеживаемые объекты. При уменьшении порога рамки детектированных объектов и отслеживаемых находятся не достаточно близко. Из рис. 13 следует, что порог соответствия не сильно влияет на рассогласование расположения объектов. Оказалось, что данных порог влияет только для тех объектов, для которых детектор не точно нашёл рамку и возникает двусмысленность, например, тень может определиться как часть объекта или найдётся только часть объекта. В качестве значения порога соответствия можно взять значение выше 1 и меньше 1,5.

Выводы

В результате проведённых исследований выявлены следующие результаты:

1) проведено исследование по определению размера гиперсферы при кластеризации объектов, в результате которого выяснилось, что минимальное рассогласование достигается при значении 0,125. При данном размере гиперсферы находятся все эталонные объекты;
2) проведено исследование по определению размера прямоугольника при кластеризации объектов, в результате которого выяснилось, что минимальное рассогласование достигается при $x = 0{,}225, y = 0{,}13, v = 4,\ a = 25^{\circ}$. Выяснено, что использование прямоугольника даёт меньшее рассогласование расположения эталонных объектов и обнаруженных рамок, чем при использовании гиперсферы, 0,62 и 0,759 соответственно;
3) проведено исследование зависимости результатов кластеризации от выбора начальной точки кластеризации и определено, что начальная точка кластеризации не влияет на результат обнаружения объектов;
4) проведено исследование зависимости порога соответствия при отслеживании объектов, в результате которого определено, что данный порог имеет минимальное значение 0,725, ниже которого начинают теряться объекты. Оказалось, что при увеличении порога не происходит сильного ухудшения результатов;
5) в результате исследований было выяснено, что метод имеет следующие ограничения:
    - при появлении объекта в кадре происходи задержка при его обнаружении, так как на нем нельзя сразу найти достаточное количество характеристических точек;
    - при неправильном подборе параметров кластеризации объекты, которые двигаются рядом и с похожей скоростью, могут попасть в один кластер;
    - нет возможности отслеживать неподвижные объекты;
    - необходимость подбора параметров кластеризации. Данный параметр зависит от размера объектов относительно изображения, может быть вычислен заранее и меняться в ходе работы.

Список литературы

1. Choi W., Savarese S. Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera // European Conference on Computer Vision. Heraklion. 2010. P. 553-567.
2. Choi W., Pantofaru C., Savarese S. Detecting and Tracking People using an RGB-D Camera via Multiple Detector Fusion // Workshop on Challenges and Opportunities in Robot Perception, at the International Conference on Computer Vision. Barcelona. 2011. P. 1076-1083.
3. Kalal Z., Mikolajczyk K., Matas J. Tracking - Learning - Detection // IEEE Transacions on pattern analysis and machine intelligence. 2010. Vol. 6. No. 1. P. 1409-1422;
4. Shi J., Tomasi C. Good features to track // Conference on Computer Vision and Pattern Recognition. Berlin, 1994. P.593-600.
5. Okuma K., Taleghani A., Freitas N.D., Freitas O.D., Little J.J., Lowe D.G. A boosted particle flter: Multitarget detection and tracking // European Conference on Computer Vision. Prague. 2004. P. 28-39.
6. Wu B., Nevatia R. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors // International Journal of Computer Vision. 2007. P. 247-266.
7. Tomasi C., Kanade T. Detection and tracking of point features // International Journal of Computer Vision, 1991. P. 132-149.
8. Kuhn H.W. The hungarian method for the assignment problem // Naval Research Logistics Quarterly. 1955. P. 29-47.
9. Khan Z., Balch T. Dellaert F. Mcmc-based particle filtering for tracking available number of interacting targets // Pattern Analysis and Machine Intelligence, IEEE Transactions. 2005. Vol. 27. No. 11. P. 279-290.
10. Harris C., Stephens M. A combined corner and edge detector // Plessey Research Roke Manor. 1988. P. 147-151.
11. Bibby C., Reid I. Real-time Tracking of Multiple Occluding Objects using Level Sets, Computer Vision and Pattern Recognition // Computer Vision and Pattern Recognition IEEE. 2010. Vol. 6. No. 1. P. 1307 – 1314.
12. Horn B. K. P., Schunck B. G. Determining optical flow //Artificial intelligence. 1981. Vol. 17. No. 1-3. P. 185–203.

13. Alhwarin F., Wang C., Risti-Durrant D., Gräser A. Improved SIFT-Features Matching for Object Recognition // Visions of Computer Science - BCS International Academic Conference. London. 2008. P. 179-190.
14. Lowe David G. Object Recognition from Local Scale-Invariant Features // International Conference on Computer Vision. Corfu. 1999. P. 1150 – 1157.
15. Friedman N., Russell S. Image Segmentation in Video Sequences / Computer Science Division University of California. Berkeley, 1997. P. 175-181.
16. Stauffer C., Grimson W.E.L. Adaptive background mixture models for real-time tracking / The Artificial Intelligence Laboratory Massachusetts Institute of Technology. 1999. P. 366 – 381.
17. Heikkila M., Pietikainen M. A Texture-Based Method for Modeling the Background and Detecting Moving Objects // Pattern Analysis and Machine Intelligence, IEEE Transactions. 2006. Vol. 28. No. 4. P. 657 – 662.
18. Elgammal A., Harwood D., Davis L. Non-parametric Model for Background Subtraction // Computer Vision Laboratory University of Maryland, College Park. 2000. P. 751-767.
19. Zach C., Pock T., Bischof H. A Duality Based Approach for Realtime Optical Flow // Research Center Institute for Computer Graphics and Vision. TU Graz. 2007. P. 214-223.
20. Craig R. Implementation of the Pure Pursuit Path Algorithm // The Robotics Institute Camegie Mellon University Pittsburgh, Pennsylvania. 1992. P. 34-50.
21. База данных с видео. Режим доступа: http://www.vision.ee.ethz.ch/~aess/dataset/ (дата обращения 22.05.2014).
22. Opencv. Режим доступа: http://opencv.org/ (дата обращения 22.05.2014).