

Шишкина А. Д.

*ФГАОУ ВО «Южный федеральный университет»,
Институт математики, механики и компьютерных наук
им. И. И. Воровича,
г. Ростов-на-Дону*

E-mail: shis@sfedu.ru

Задача обнаружения движущихся объектов – одна из фундаментальных тем в области компьютерного зрения. Для решения подобных задач в настоящее время принято использовать различные модели глубокого обучения и нейросетевые алгоритмы, например YOLO или Faster-CNN, описанные в работах [1,2], так как они демонстрируют высокий уровень точности детекции целей. Однако использование подобных методов также сопряжено с рядом недочетов, указанных в научном обзоре [3]: серьезная вычислительная сложность, продолжительное время обработки кадра, необходимость в подготовке специализированных обучающих выборок, адаптированных под конкретный тип объекта, например, животное, человек или транспортное средство и уникальный сценарий наблюдения. Все перечисленные ограничения затрудняют использование подобных моделей в оптико-электронных системах реального времени, где требуются высокая скорость анализа в сочетании с ограниченными вычислительными ресурсами.

Наиболее распространенным методом для решения выделенной проблемы, согласно статье [4], считается вычисление разности между идущими подряд фреймами видеопотока. Однако он демонстрирует устойчивые и корректные результаты в условиях идеально неподвижной сцены и статичного фона, в более случаях панорамирования и тряски камеры разность наблюдаются во всех пикселях изображения, что приводит к многочисленным ложным срабатываниям. В данной работе рассматривается алгоритм выявления движущихся объектов путем вычисления разности кадров с компенсацией глобального движения сцены. Реализация алгоритма выполнена на языке программирования C++(17) с использованием библиотеки компьютерного зрения OpenCV и фреймворка Qt5 для создания адаптивных экранов приложения.

На первом этапе алгоритма производится непосредственная оценка глобального движения в кадре. Для этого на изображении выделяются устойчивые, различимые характерные точки с использованием алгоритма Good Features to Track (Shi–Tomasi), описанного в работе [5]. Благодаря использованию матрицы градиентов и минимального порога он обеспечивает выделение локальных областей с выраженной структурой интенсивности, необходимых для дальнейшего сравнения кадров видеопотока. Затем используется метод Lucas–Kanade [6], который позволяет определить смещение выбранных характерных точек в последовательных фреймах и установить соответствия между их положениями. Таким образом, формируется набор пар характерных точек, описывающих смещение отдельных участков изображения.

На основе полученного набора производится оценка аффинного преобразования, описывающего глобальное изменение сцены между фреймами:

$$T = \begin{pmatrix} a & b & t_x \\ c & d & t_y \end{pmatrix}, \quad (1)$$

где t_x, t_y – смещение изображения,

a, b, c, d – параметры вращения и масштабирования.

Поскольку среди выведенных соответствий могут присутствовать ошибочные точки, для повышения устойчивости оценки параметров преобразования применяется метод RANSAC (Random Sample Consensus), предложенный в работе [7].

После оценки параметров межкадрового смещения выполняется стабилизация изображения. Для этого текущий кадр преобразуется в систему координат предыдущего, что позволяет компенсировать влияние движения камеры на анализ и выявление целей.

Листинг 1. C++, Преобразование кадра в с.к. предыдущего

```
cv::Mat A_inv;  
cv::invertAffineTransform(A, A_inv);  
  
cv::warpAffine(currImg, currAligned, A_inv, prevImg.size(),  
cv::INTER_LINEAR, cv::BORDER_REPLICATE);
```

На следующем этапе выполняется вычисление разности между стабилизированным текущим кадром и предыдущим:

$$D = |I_t^{aligned} - I_{t-1}|, \quad (2)$$

где I_{t-1} – предыдущий кадр,
 I_t – текущий кадр.

В результате операции формируется карта различий, отражающая потенциальные области движения в сцене. Впоследствии она преобразуется в бинарное изображение с использованием пороговой функции:

$$B(x, y) = \begin{cases} 1, & D(x, y) > T \\ 0, & D(x, y) \leq T \end{cases}, \quad (3)$$

где T – пороговое значение.

Пороговая функция позволяет отделить области значимых изменений в полученном изображении от несущественных фоновых шумовых движений. Также для дополнительного подавления шумовых артефактов и объединения близких пикселей применяется морфологическая операция дилатации. Далее производится поиск связных компонент и формирование областей, соответствующих движущимся целям, приведенный в листинге 2. Для каждой пространственной области производится поиск ограничивающего её прямоугольника. Полученные прямоугольные области рассматриваются как кандидаты на движущиеся объекты и выводятся на экран. Образец возможного выделения целей показан на рисунке 1.



Рис 1. Образец выделения цели в приложении

Листинг 2. C++, Поиск контуров

```
findContours(thresh, contours, RETR_EXTERNAL,
CHAIN_APPROX_SIMPLE);

std::vector<Rect> rects;
for (int i = 0; i < (int)contours.size(); i++)
{
    rects.push_back(boundingRect(contours[i]));
}

std::vector<Rect> unionContours = ContoursUnion(rects);

if (objects_.size() == 0)
{
    objects_ = unionContours;
    lifeTime_.resize(objects_.size());
}
else
{
    FramesUnion(unionContours);
    objects_ = SimpleContoursUnion(objects_);
    DeleteOldContours(output);
}
```

Описанный алгоритм детекции объектов использует разность кадров и компенсацию глобального движения сцены, позволяя существенно уменьшить количество ложных срабатываний при использовании камеры или тепловизионных устройств в подвижных, динамических системах. Он обладает высокой скоростью работы, поскольку основан на простых математических операциях: получение и преобразование оптического потока с медиаустройств, аффинных преобразованиях, разности кадров и морфологической обработки. А также в отличие от нейросетевых общеупотребимых методов, предложенный подход обладает низкими требованиями к аппаратному оборудованию, не требует предварительного процесса обучения и может быть эффективно и быстро внедрен в системы слежения реального времени.

Литература

1. Redmon J., Divvala S., Girshick R., Farhadi A. – You Only Look Once: Unified, Real-Time Object Detection [Электронный ресурс]: <https://arxiv.org/abs/1506.02640>

2. Girshick R. – Fast R-CNN / Proceedings of the IEEE International Conference on Computer Vision (ICCV) [Электронный ресурс]: <https://arxiv.org/abs/1504.08083>
3. Rawat W., Wang Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review // Neural Computation. – 2017. – Vol. 29, №9. – pp. 2352–2449
4. Bouwmans T. Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview // Computer Science Review. – 2014. – Vol. 11–12. – pp. 31–66
5. Shi J., Tomasi C. Good Features to Track. Proceedings of CVPR, 1994
6. Lucas B., Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. IJCAI, 1981
7. Fischler M., Bolles R. Random Sample Consensus (RANSAC). Communications of the ACM, 1981