

УДК 004.932.2

## АВТОМАТИЧЕСКОЕ ТЕСТИРОВАНИЕ АЛГОРИТМОВ СОПРОВОЖДЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ

К.А. ВОЛКОВ, В.О. ЛАНДЕНОК, С.С. ЗАЙЦЕВ,  
Х.М. КАРБАЛАИ САЛЕХ, М.Б.М. МАХММУД

*Белорусский государственный университет информатики и радиоэлектроники  
П. Бровки, 6, Минск, 220013, Беларусь*

*Поступила в редакцию 30 ноября 2016*

Рассмотрены вопросы тестирования алгоритмов сопровождения объектов на видеопоследовательности. Предложен метод автоматической оценки результатов работы алгоритма сопровождения. Разработано программное обеспечение для автоматического тестирования алгоритмов сопровождения.

*Ключевые слова:* сопровождение подвижных объектов, алгоритм, тестирование.

### Введение

Алгоритмы сопровождения подвижных объектов (АСПО) находят широкое применение в различных областях, связанных с автоматической и автоматизированной обработкой видеоизображений [1], таких как видеонаблюдение, системы управления движением наземных и летательных аппаратов, техническое зрение промышленных роботов, киноиндустрия и др. Разнообразие решаемых прикладных задач обуславливает сильную вариацию исходных видеоданных по частоте кадров, цветности, динамическому диапазону, размеру изображения, наличию и типу помех, что не позволяет создать единый универсальный АСПО [2, 3]. В настоящее время разработано значительное число методов и алгоритмов сопровождения объектов, основанных на сегментации, выделении ключевых точек, нахождении контуров, построении яркостных и цветовых диаграмм, корреляции блоков изображения, формировании ковариационных матриц, вычислении оптического потока, а также использующих различные комбинации данных подходов [3-5]. Основная сложность выбора одного из существующих алгоритмов в качестве готового решения или прототипа для разработки конкретного прикладного программно-аппаратного продукта заключается в том, что визуальная оценка качества работы алгоритма субъективна, трудоемка и не всегда репрезентативна из-за значительных объемов тестовых видеозаписей. Аналогичные недостатки визуальной оценки алгоритма сопровождения проявляются и в итерационном процессе разработки и отладки, а также при приемо-сдаточных испытаниях. В работе предлагается метод автоматического тестирования программных реализаций АСПО с использованием набора тестовых видеозаписей.

### Оценка работы алгоритма сопровождения

Автоматическое сопровождение объекта используется для определения его координат на кадрах видеопоследовательности с последующим расчетом производных величин, таких как пространственные координаты, скорость, направление движения и др. Полученные результаты в дальнейшем могут быть использованы в системах управления или системах принятия решений более высокого уровня. Соответственно, идеальным случаем работы АСПО является достоверное обнаружение объекта на каждом кадре, что обычно недостижимо на практике в силу специфических недостатков используемых алгоритмов или наличия объективных причин, та-

ких как ограниченные видимость или обзорность наблюдаемого пространства, наличие помех и искажений при фиксации, кодировании или передаче видеопоследовательности [1, 5]. Вместе с тем, использование методов предсказания и фильтрации координат объекта при постобработке выходных данных АСПО позволяет частично нивелировать ложное нахождение объекта в кадре или невозможность его нахождения. В работе предлагается использовать количественную оценку подобных ошибок при работе алгоритма сопровождения на тестовых видеозаписях, что позволит судить о его применимости для использования в конкретной программно-аппаратной системе [6-8].

Ошибка первого рода [8], или ложное определение координат, детектируется в кадре в том случае, если расхождение между тестируемыми и проверочными координатами превышает величины  $\sigma$ , или если найдены тестируемые координаты при неопределенных проверочных. Данный тип ошибки на  $i$ -м ( $i \in \{1, N\}$ ) кадре может быть определен по формуле

$$E_i^1 = \begin{cases} 1: (\|T_i^M - T_i^A\| \geq \sigma \cap T_i^M \neq T^{MISS} \cap T_i^A \neq T^{MISS}) \cup (T_i^M = T^{MISS} \cap T_i^A \neq T^{MISS}) \\ 0: (\|T_i^M - T_i^A\| < \sigma \cap T_i^M \neq T^{MISS} \cap T_i^A \neq T^{MISS}) \cup (T_i^M = T^{MISS} \cap T_i^A = T^{MISS}) \end{cases},$$

где  $N$  – количество кадров на тестовой видеозаписи;  $T_i^M$  – проверочные координаты центра объекта на  $i$ -ом кадре (в пикселях), введенные вручную;  $T_i^A$  – тестируемые координаты центра объекта на  $i$ -ом кадре (в пикселях), полученные в результате работы алгоритма сопровождения;  $\sigma$  – допустимое расхождение тестируемых и проверочных координат центра объекта (в пикселях);  $T^{MISS} = (-1, -1)$  – условное обозначение координат объекта, который отсутствует в кадре.

Ошибка второго рода [8], или потеря объекта, детектируется в кадре в том случае, если тестируемые координаты не найдены при определенных проверочных. Данный тип ошибки на  $i$ -ом кадре может быть определен по формуле

$$E_i^2 = \begin{cases} 1: (T_i^M \neq T^{MISS} \cap T_i^A = T^{MISS}) \\ 0: (T_i^M = T^{MISS} \cup T_i^A \neq T^{MISS}) \end{cases}.$$

Для оценки качества работы алгоритма сопровождения на видеопоследовательности предлагается использовать такие практически значимые параметры, как количество идущих подряд ошибок (первого и второго рода) и количество ошибок за интервал времени [9].

Присутствие объекта на видеопоследовательности из  $m$  кадров, в каждом из которых содержится ошибка первого или второго рода, может быть определено по формулам

$$E_m^{SEQ1} = \begin{cases} 1: \max_{k=1, N-m+1} \left( \sum_{i=k}^{k+m} E_i^1 \right) = m \\ 0: \max_{k=1, N-m+1} \left( \sum_{i=k}^{k+m} E_i^1 \right) < m \end{cases}, \quad E_m^{SEQ2} = \begin{cases} 1: \max_{k=1, N-m+1} \left( \sum_{i=k}^{k+m} E_i^2 \right) = m \\ 0: \max_{k=1, N-m+1} \left( \sum_{i=k}^{k+m} E_i^2 \right) < m \end{cases}.$$

Присутствие на тестовой видеозаписи последовательности из  $t$  кадров, такой что не менее чем на  $p$  кадрах содержится ошибка первого или второго рода, может быть определено по формулам

$$E_{t,p}^{CNT1} = \begin{cases} \max_{k=1, N-t+1} \left( \sum_{i=k}^{k+t} E_i^1 \right) \geq p \\ \max_{k=1, N-t+1} \left( \sum_{i=k}^{k+t} E_i^1 \right) < p \end{cases}, \quad E_{t,p}^{CNT2} = \begin{cases} \max_{k=1, N-t+1} \left( \sum_{i=k}^{k+t} E_i^2 \right) \geq p \\ \max_{k=1, N-t+1} \left( \sum_{i=k}^{k+t} E_i^2 \right) < p \end{cases}.$$

Заметим также, что при  $p = t = m$  справедливы равенства  $E_m^{SEQ1} = E_{t,p}^{CNT1}$  и  $E_m^{SEQ2} = E_{t,p}^{CNT2}$ .

В качестве метрики, определяющей возможность использования исследуемого алгоритма сопровождения для видеоданных, сходных с представленными на тестовой видеозаписи, предлагается использовать следующую формулу

$$R = \sum_{i=1}^N \sum_{p=1}^I R_{i,p}^{CNT1} E_{i,p}^{CNT1} + \sum_{i=1}^N \sum_{p=1}^I R_{i,p}^{CNT2} E_{i,p}^{CNT2},$$

где  $R_{i,p}^{CNT1}$  и  $R_{i,p}^{CNT2}$  – весовые коэффициенты, подобранные таким образом, что величина  $R < 1$ , если исследуемый алгоритм успешно прошел тестирование и может быть использован для практического применения, и  $R \geq 1$  – в противном случае.

### Автоматическое тестирование алгоритма сопровождения

В работе предлагается метод автоматического тестирования АСПО [7, 8], который включает в себя следующие шаги.

Шаг 1. Для проведения тестирования выбираются видеозаписи, являющиеся наиболее репрезентативными для области применения АСПО.

Шаг 2. В каждом кадре видеозаписи вручную маркируется центр сопровождаемого объекта или указывается, что объект на изображении отсутствует.

Шаг 3. Для первого кадра видеозаписи осуществляется захват объекта (в указанных на предыдущем шаге координатах) с использованием тестируемого алгоритма.

Шаг 4. На следующем кадре видеозаписи осуществляется автоматический поиск объекта с использованием тестируемого алгоритма и обновляется метрика  $R$ .

Шаг 5. Если  $R < 1$  и не достигнут конец видеозаписи – переход к шагу 4, иначе выводятся результаты тестирования  $\{E_i^1\}$ ,  $\{E_i^2\}$ ,  $E_m^{SEQ1}$ ,  $E_m^{SEQ2}$ ,  $E_{i,p}^{CNT1}$ ,  $E_{i,p}^{CNT2}$ ,  $R$ .

Для реализации предложенного метода тестирования разработано программное обеспечение, структурная схема которого представлена на рис. 1. В Блоке 1 тестовые видеозаписи хранятся в виде AVI-файлов длительностью 1000-3000 кадров, что позволяет получить достоверный результат за короткое время тестирования. В Блоке 2 осуществляется ручное маркирование координат центров сопровождаемых объектов на тестовых видеозаписях (рис. 2). Результаты маркирования сохраняются в виде XML файлов в Блоке 3. Блок 4 использует тестовые видеозаписи из Блока 2 и информацию о расположении объекта в кадре из Блока 3 для тестирования работы функций захвата и поиска Блока 5. Параметры тестирования для вычисления оценочной функции передаются в Блок 4 из Блока 6. Блок 5 содержит тестируемый алгоритм и реализуется в виде внешней подключаемой библиотеки, в которой доступны функции, представленные в таблице. Блок 4 покадрово передает видеопоследовательность в Блок 5, анализирует получаемый результат, сравнивая его с эталонным, вычисляет метрики и передает их в Блок 6 для визуализации. Для ускорения процесса тестирования Блоки 2, 3, 4, 5 могут быть представлены в нескольких экземплярах, работающих параллельно с различными тестовыми видеозаписями. Это позволяет сократить время тестирования для одной ЭВМ в 3-7 раз (в зависимости от размера видеозаписи и быстродействия дисковой системы и процессора), а для кластера – еще более, практически пропорционально числу используемых ЭВМ.

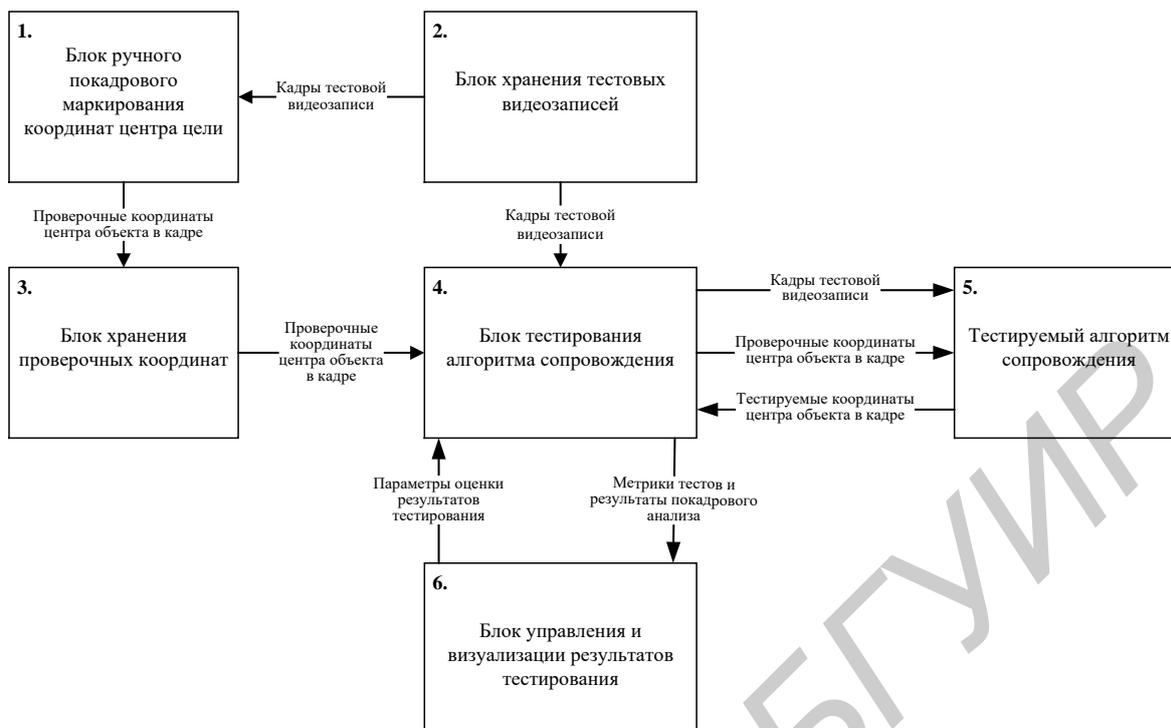


Рис. 1. Структурная схема программного обеспечения для тестирования алгоритмов сопровождения



Рис. 2. Пользовательский интерфейс блока ручного маркирования координат центра цели

Программное обеспечение для автоматического тестирования АСПО реализовано на языке C++, для организации пользовательского интерфейса и доступа к видеоданным используется фреймворк Qt 5.7. Основные функции программного интерфейса автоматического сопровождения объектов приведены в таблице.

## Программный интерфейс блока автоматического сопровождения объектов

Функция	Параметр	Тип	Описание
<b>SetFrame</b>		void	Инициализация очередного кадра видеопоследовательности
	frameData	char*	Указатель на начало растровых данных изображения
	frameWidth	int	Ширина изображения (в пикселях)
	frameHeight	int	Высота изображения (в пикселях)
<b>LockObject</b>		void	Захват объекта в кадре
	screenCoord	Point	Координаты объекта в кадре (в пикселях)
<b>DropObject</b>		void	Прекращение сопровождения объекта
<b>FindObject</b>		Point	Поиск объекта в кадре. Результат – координаты объекта в кадре (в пикселях) или (-1, -1), если объект не найден

### Заключение

Предложенный в работе метод автоматического тестирования алгоритмов сопровождения подвижных объектов показал свою высокую эффективность при использовании в процессе разработки программного обеспечения для систем видеонаблюдения и управления движением БЛА. Автоматический анализ результатов сопровождения позволил отказаться от визуального анализа и за счет этого значительно сократить время выполнения и повысить объективность итерационных процедур выбора прототипов и настройки внутренних параметров разрабатываемых алгоритмов.

## AUTOMATIC TESTING OF MOVING OBJECTS TRACKING ALGORITHMS ON THE VIDEO

K.A. VOLKOV, U.A. LANDZENOK, S.S. ZAITSEV,  
H.M. KARBALAEI SALEH, M.B.M MAHMOOD

### Abstract

The problems of object tracking algorithms testing on the video were considered. A method for automatic evaluation of the results of the tracking algorithm was offered. Software for automatic testing of tracking algorithm was developed.

*Keywords:* moving objects tracking, algorithm, testing.

### Список литературы

1. Amit Y. 2D Object Detection and Recognition: models, algorithms and networks. The MIT Press, 2002.
2. Bradski G., Kaehler A. Learning OpenCV Computer Vision with OpenCV Library. O' Reilly Media Publishers, 2008.
3. Baggio D., Escriva D., Mahmood N. et. al Mastering OpenCV with Practical Computer Vision Projects. Birmingham, 2012.
4. Gady A. Introduction to programming with OpenCV. Chicago, 2006.
5. Maggio E., Cavallaro A. Video Tracking: Theory and Practice. New York, 2011.
6. Канер К., Фолк Д., Нгуен Енг К. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. К., 2001.
7. Синуцын С., Налютин Н. Верификация программного обеспечения. М., 2008.
8. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб, 2004.
9. Золотых Н., Кустикова В., Мееров И. // Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. №5-2. С. 347-357.