

УДК 004

Б.А. Залесский, Э.Н. Середин, Н.В. Ядловский

**ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ НА ОСНОВЕ
СРАВНЕНИЯ ГИСТОГРАММ ЦВЕТА**

Предлагаются три версии гистограммного алгоритма отслеживания объектов на видеопоследовательностях, снятых нестабилизированной камерой. Версии основываются на сравнении с помощью критерия Баттачариа гистограмм цвета областей кадров либо гистограмм цвета пар ближайших пикселей областей кадров, которые могут рассматриваться как частный случай матриц встречаемости. Использование технологии программирования видеокарты CUDA позволяет добиться выполнения версий в режиме реального времени. Проводится сравнение модифицированного алгоритма с хорошо известным алгоритмом среднего сдвига, основанным на линейном приближении критерия Баттачариа. На основе экспериментов показывается, что модификации алгоритма являются более точными и надежными по сравнению с алгоритмом среднего сдвига, хотя и требуют большего объема вычислений. Проводится сравнение предложенных версий с корреляционными алгоритмами.

Введение

В последние годы алгоритмы отслеживания объектов широко используются для решения прикладных задач в различных областях народного хозяйства, например для обеспечения безопасности дорожного движения, сохранности имущества, в навигации и т. д.

Для отслеживания объектов на видеопоследовательностях, снятых в разных условиях, используются различные подходы. Рассмотрим случай, когда видеопоследовательность снята камерой, установленной над объектом, например на борту летательного аппарата. Задача заключается в отслеживании в реальном времени объекта интереса, наблюдаемого камерой стандартного разрешения. Особенность данной задачи состоит в относительно невысоком качестве получаемого видео, часто нестабилизированного, кадры которого обычно искажены шумами различных типов.

В настоящее время известны несколько подходов к решению сформулированной задачи, например основанные на вычислении среднего сдвига, сравнении корреляций, гистограмм яркостей и ориентированных градиентов, оптических потоков, текстур, активных контуров и др. [1–7]. Использование сложных алгоритмов, требующих большого объема вычислений, ограничено в данном случае малым допустимым временем (25–40 мс), необходимым для обработки каждого кадра.

Алгоритм среднего сдвига (mean shift) был предложен для преодоления упомянутых трудностей. Использование линейного приближения произведения гистограмм выделенных областей начального и текущего кадров, которое реализует критерий Баттачариа, дает возможность уменьшить время вычисления так называемого среднего сдвига текущего кадра [1] настолько, что процессорные версии алгоритма выполняются на современных персональных компьютерах всего за несколько миллисекунд. Однако проведенное тестирование алгоритма на видеопоследовательностях невысокого качества показало его меньшую точность и надежность по сравнению с другими алгоритмами, например корреляционными.

В статье предложены три версии алгоритма отслеживания объектов на видеопоследовательностях, в которых, как и в алгоритме среднего сдвига, используется критерий Баттачариа для сравнения многомерных гистограмм цвета (инвариантных или почти инвариантных к повороту изображения) при выполнении отслеживания объекта интереса. Однако при этом, в отличие от известного алгоритма, сам критерий не заменяется на его линейное приближение. Во второй и третьей версиях алгоритма используются гистограммы яркостей соседних пикселей, которые могут рассматриваться как один из вариантов матриц встречаемости [8]. Приведены результаты сравнения точности и надежности разработанных версий с известными алгоритмами среднего сдвига и корреляционного поиска.

Использование исходного представления критерия Баттачариа в виде произведения нормированных квантизованных гистограмм позволило повысить точность и надежность оценки координат отслеживаемого объекта, хотя и привело к увеличению объема выполняемых вычислений. Возникшие трудности с программированием работающих в режиме реального времени версий были преодолены с помощью технологии программирования видеокарты CUDA. Использование видеокарты позволило сократить время поиска объекта на каждом кадре до 10–15 мс.

1. Описание алгоритма отслеживания объектов на основе критерия Баттачариа

Предложенные версии алгоритма отслеживания объектов могут быть использованы для полутоновых и *RGB*-изображений, но, в связи с тем что случай цветных изображений более труден для реализации из-за большего объема вычислений, здесь будут рассмотрены только цветные видеопоследовательности.

Обозначим через **I** *RGB*-изображение с пикселями $p = (x, y)$, образующими регулярную решетку $S = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$, и будем считать, что координаты цвета пикселей $I(p) = (I_R(p), I_G(p), I_B(p))$ изменяются в диапазоне $0, \dots, L-1$. Кадры видеопоследовательности в моменты времени $t = \{0, 1, \dots\}$ будем обозначать \mathbf{I}_t . Все три версии предложенного алгоритма отслеживания основаны на сравнении оконных цветовых гистограмм различных типов с помощью критерия Баттачариа. Гистограммы строятся по квантизованным изображениям **J**, яркости цветовых каналов которых изменяются в меньшем диапазоне $0, \dots, l-1$ ($0 < l \leq L$) и задаются равенствами

$$J_R(p) = \left\lfloor \frac{(l-1)I_R(p)}{(L-1)} \right\rfloor, \quad J_G(p) = \left\lfloor \frac{(l-1)I_G(p)}{(L-1)} \right\rfloor, \quad J_B(p) = \left\lfloor \frac{(l-1)I_B(p)}{(L-1)} \right\rfloor,$$

где $[a]$ – целая часть числа a . Квантизация изображений производится по двум причинам. Во-первых, использование меньшего диапазона яркостей уменьшает объем требуемых вычислений и, следовательно, сокращает время обработки кадров. Во-вторых, квантизация снижает влияние на результаты нелинейных изменений яркостей кадров, присущих в той или иной мере всем камерам. На реальных 24-битных *RGB*-видеопоследовательностях ($L = 256$) значения $l = 8, \dots, 32$ позволяют получать достаточно точные и надежные результаты. При $l > 32$ точность и надежность результатов практически не повышаются, но время вычислений становится недопустимо большим. При $l < 8$ ухудшается точность.

Пусть окно $O(p)$ – квадрат с центром в пикселе p , содержащий $m \times m$ пикселей, $O(p_0)$ – окно, содержащее объект интереса, выделенный на исходном кадре \mathbf{I}_0 . В первой версии алгоритма используются традиционные для метода средних сдвигов сглаженные оконные гистограммы $\mathbf{h}_{t,p}$ квантизованных кадров \mathbf{J}_t , которые представляют собой 3D-матрицы, индексированные квантизованным цветом:

$$\mathbf{h}_{t,p} = \{h_{t,p}(n_R, n_G, n_B)\}_{0 \leq n_R, n_G, n_B \leq l-1}$$

вида

$$h_{t,p}(n_R, n_G, n_B) = \frac{1}{C} \sum_{q \in O(p)} w(q-p) \mathbf{1}_{(J_t(q)=(n_R, n_G, n_B))}, \quad (1)$$

где нормирующая константа C выбирается так, чтобы сумма всех элементов гистограммы была равна 1; w – сглаживающая функция, а $\mathbf{1}_{(J_t(q)=(n_R, n_G, n_B))}$ – индикатор, равный 1, если $J(q) = (n_R, n_G, n_B)$, и равный 0 в противном случае. В качестве сглаживающей функции во всех трех версиях использовалась либо экспонента

$$w(p) = w_r(p) = \exp\left(-\frac{1}{r} \|p\|^2\right), \quad r > 0,$$

либо ядро Епанечникова

$$w(p) = w_r(p) = \begin{cases} 1 - \frac{1}{r} \|p\|^2, & \text{если } \|p\|^2 < r, \\ 0 & \text{в противном случае} \end{cases}$$

с параметром масштаба r .

В качестве меры близости двух гистограмм применялся непосредственно критерий Баттачариа, который задается равенством

$$\alpha(\mathbf{h}', \mathbf{h}'') = \sum_{n_R=0, n_G=0, n_B=0}^{l-1, l-1, l-1} \sqrt{h'(n_R, n_G, n_B) h''(n_R, n_G, n_B)}. \quad (2)$$

(Напомним, что в классическом алгоритме среднего сдвига вместо $\alpha(\mathbf{h}', \mathbf{h}'')$ используется его линейное приближение.)

Были исследованы две модификации алгоритма поиска на основе критерия α . В одной из модификаций оценка \hat{p}_t положения окна, содержащего объект интереса на текущем кадре \mathbf{J}_t , выполнялась на основе глобального поиска по всему кадру:

$$\hat{p}_t = \operatorname{argmax}_{q \in S} \left\{ \alpha(\mathbf{h}_{0,p_0}, \mathbf{h}_{t,q}) \right\}.$$

Во второй модификации первой версии алгоритма использовался локальный поиск в окне $W(\hat{p}_{t-1})$ с центром \hat{p}_{t-1} , равным оценке положения объекта интереса на предыдущем кадре, и размером, зависящим от оценки максимальной скорости движения объекта относительно кадра:

$$\hat{p}_t = \operatorname{argmax}_{q \in W(\hat{p}_{t-1})} \left\{ \alpha(\mathbf{h}_{0,p_0}, \mathbf{h}_{t,q}) \right\}.$$

Каждая из модификаций имеет свои преимущества и недостатки, которые будут описаны в разд. 2.

Вторая версия предложенного алгоритма основана на сравнении сглаженных оконных гистограмм $\tilde{\mathbf{h}}_{t,p}$ совместных яркостей квантизованных кадров \mathbf{J}_t пар ближайших пикселей, которые представляют собой 6D-матрицы, индексированные парами квантизованных цветов ближайших пикселей:

$$\tilde{\mathbf{h}}_{t,p} = \left\{ \tilde{h}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) \right\}_{0 \leq n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B} \leq l-1}.$$

Сглаженная оконная гистограмма $\tilde{\mathbf{h}}_p$ для изображения \mathbf{J} может быть представлена в виде суммы, аналогичной сумме (1), однако представление получается весьма громоздким, поэтому приведем конструктивный способ ее описания. Для пиксела q обозначим через q' его ближайшего правого соседа, а через q'' его ближайшего верхнего соседа. Нетрудно заметить, что $q' = q + (1, 0)$, а $q'' = q - (0, 1)$. Алгоритм построения $\tilde{\mathbf{h}}_p$ состоит из следующих шагов:

Шаг 1. Полагаем все $\tilde{h}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) = 0$.

Шаг 2. Просматриваем все пиксели q окна $O(p)$. Для каждого пиксела q и переменной $F \in \{R, G, B\}$ вычисляем

$$n_{1,F} = \min(J_F(q), J_F(q')), \quad n_{2,F} = \max(J_F(q), J_F(q')),$$

а затем находим

$$\tilde{h}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) := \tilde{h}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) + w(q - p).$$

Шаг 3. Просматриваем все пиксели q окна $O(p)$. Для каждого пикселя q и переменной $F \in \{R, G, B\}$ вычисляем

$$n_{1,F} = \min(J_F(q), J_F(q')), \quad n_{2,F} = \max(J_F(q), J_F(q'')),$$

а затем находим

$$\tilde{\mathbf{h}}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) := \tilde{\mathbf{h}}_{t,p}(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) + w(q - p).$$

Шаг 4. Нормируем гистограмму $\tilde{\mathbf{h}}_{t,p}$ так, чтобы сумма ее элементов была равна единице.

Теоретически построенная нижняя треугольная 6D-матрица $\tilde{\mathbf{h}}$ является инвариантной только к поворотам на углы от $\omega \in \left(\frac{\pi}{2}j - \frac{\pi}{6}, \frac{\pi}{2}j + \frac{\pi}{6}\right)$, $j = 0, 1, 2, 3$, но практически она лишь незначительно меняется при других углах поворота, что позволяет применять ее для реального сравнения областей изображений, повернутых на произвольные углы, о чем свидетельствуют проведенные эксперименты.

Во второй версии алгоритма для поиска объекта интереса использовался критерий Батгачариа

$$\beta(\tilde{\mathbf{h}}, \tilde{\mathbf{h}}'') = \sum_{\substack{l-1, l-1, l-1, l-1, l-1, l-1 \\ n_{1,R}=0, n_{2,R}=0, n_{1,G}=0 \\ n_{2,G}=0, n_{1,B}=0, n_{2,B}=0}} \sqrt{\tilde{\mathbf{h}}'(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B}) \tilde{\mathbf{h}}''(n_{1,R}, n_{2,R}, n_{1,G}, n_{2,G}, n_{1,B}, n_{2,B})}. \quad (3)$$

В первой модификации второй версии алгоритма по аналогии с первой модификацией первой версии осуществляется поиск по всему текущему кадру:

$$\hat{p}_t = \operatorname{argmax}_{q \in S} \left\{ \beta(\tilde{\mathbf{h}}_{0,p_0}, \tilde{\mathbf{h}}_{t,q}) \right\}.$$

Во второй модификации второй версии алгоритма осуществляется локальный поиск на текущем кадре в окне $W(\hat{p}_{t-1})$, центр которого полагается равным оценке координат объекта на предыдущем кадре \hat{p}_{t-1} :

$$\hat{p}_t = \operatorname{argmax}_{q \in W(\hat{p}_{t-1})} \left\{ \beta(\tilde{\mathbf{h}}_{0,p_0}, \tilde{\mathbf{h}}_{t,q}) \right\}.$$

Вторая версия обеспечивает наиболее точное и надежное отслеживание объекта среди всех предложенных в данной статье, однако она требует наибольшего числа вычислений, которые достаточно трудно осуществить в режиме реального времени.

Для случая ограниченных вычислительных ресурсов предлагается третья версия алгоритма отслеживания объектов, в которой используются сглаженные оконные гистограммы ${}^R\bar{\mathbf{h}}_{t,p}$, ${}^G\bar{\mathbf{h}}_{t,p}$, ${}^B\bar{\mathbf{h}}_{t,p}$ совместных яркостей RGB-каналов квантизованных кадров \mathbf{J}_t , представляющие собой 3D-матрицы, индексированные парами квантизованных цветов ближайших пикселей:

$${}^A\bar{\mathbf{h}}_{t,p} = \left\{ {}^A\mathbf{h}_{t,p}(n_1, n_2) \right\}_{0 \leq n_1, n_2 \leq l-1}, \quad A \in \{R, G, B\}.$$

Напомним, что для пикселя q его ближайший правый сосед обозначен буквой q' , а ближайший верхний сосед – буквой q'' так, что $q' = q + (1, 0)$, а $q'' = q - (0, 1)$.

Конструктивный способ построения ${}^R\bar{\mathbf{h}}_{t,p}$, ${}^G\bar{\mathbf{h}}_{t,p}$, ${}^B\bar{\mathbf{h}}_{t,p}$ состоит из следующих шагов:

Шаг 1. Полагаем все ${}^A\bar{\mathbf{h}}_{t,p} = 0, A \in \{R, G, B\}$.

Шаг 2. Просматриваем все пиксели q окна $O(p)$. Для каждого пикселя q и переменной $A \in \{R, G, B\}$ вычисляем

$$n_{1,A} = \min(J_A(q), J_A(q')), n_{2,A} = \max(J_A(q), J_A(q')),$$

а затем находим

$${}^A\bar{\mathbf{h}}_{t,p}(n_{1,A}, n_{2,A}) := {}^A\bar{\mathbf{h}}_{t,p}(n_{1,A}, n_{2,A}) + w(q - p).$$

Шаг 3. Просматриваем все пиксели q окна $O(p)$. Для каждого пикселя q и переменной $A \in \{R, G, B\}$ вычисляем

$$n_{1,A} = \min(J_A(q), J_A(q'')), n_{2,A} = \max(J_A(q), J_A(q'')),$$

а затем находим

$${}^A\bar{\mathbf{h}}_{t,p}(n_{1,A}, n_{2,A}) := {}^A\bar{\mathbf{h}}_{t,p}(n_{1,A}, n_{2,A}) + w(q - p).$$

Шаг 4. Нормируем гистограмму ${}^A\bar{\mathbf{h}}_{t,p}$ так, чтобы сумма ее элементов была равна 1.

В третьей версии алгоритма для поиска объекта интереса используется критерий Баттачариа

$$\begin{aligned} \gamma \left({}^R\bar{\mathbf{h}}, {}^G\bar{\mathbf{h}}, {}^B\bar{\mathbf{h}}, {}^R\bar{\mathbf{h}}'', {}^G\bar{\mathbf{h}}'', {}^B\bar{\mathbf{h}}'' \right) = & \sum_{\substack{l=1, l-1, l-1, l-1, l-1, l-1 \\ n_{1,R}=0, n_{2,R}=0, n_{1,G}=0, \\ n_{2,G}=0, n_{1,B}=0, n_{2,B}=0}} \sqrt{{}^R\bar{h}'(n_{1,R}, n_{2,R}) {}^R\bar{h}''(n_{1,R}, n_{2,R})} \times \\ & \times \sqrt{{}^G\bar{h}'(n_{1,G}, n_{2,G}) {}^G\bar{h}''(n_{1,G}, n_{2,G})} \times \sqrt{{}^B\bar{h}'(n_{1,B}, n_{2,B}) {}^B\bar{h}''(n_{1,B}, n_{2,B})}. \end{aligned} \quad (4)$$

В первой модификации третьей версии алгоритма по аналогии с первой модификацией первой версии осуществляется поиск по всему текущему кадру

$$\hat{p}_t = \operatorname{argmax}_{q \in S} \left\{ \gamma \left({}^R\bar{\mathbf{h}}_{0,p_0}, {}^G\bar{\mathbf{h}}_{0,p_0}, {}^B\bar{\mathbf{h}}_{0,p_0}, {}^R\bar{\mathbf{h}}_{t,p}, {}^G\bar{\mathbf{h}}_{t,p}, {}^B\bar{\mathbf{h}}_{t,p} \right) \right\}.$$

Во второй модификации третьей версии алгоритма осуществляется локальный поиск на текущем кадре в окне $W(\hat{p}_{t-1})$, центр которого полагается равным оценке координат объекта на предыдущем кадре \hat{p}_{t-1} :

$$\hat{p}_t = \operatorname{argmax}_{q \in W(\hat{p}_{t-1})} \left\{ \gamma \left({}^R\bar{\mathbf{h}}_{0,p_0}, {}^G\bar{\mathbf{h}}_{0,p_0}, {}^B\bar{\mathbf{h}}_{0,p_0}, {}^R\bar{\mathbf{h}}_{t,p}, {}^G\bar{\mathbf{h}}_{t,p}, {}^B\bar{\mathbf{h}}_{t,p} \right) \right\}.$$

Третья версия алгоритма требует меньших объемов памяти и вычислений по сравнению со второй, но больших по сравнению с первой. В разд. 2 приведены результаты вычислительных экспериментов с описанными версиями алгоритма.

2. Особенности программной реализации алгоритмов

Как было отмечено выше, представленные версии алгоритма были предложены для повышения точности и надежности классической версии алгоритма среднего сдвига, в которой

вместо критерия Баттачариа использовалось его линейное приближение. Применение линеализации позволило Чену [1], а затем Команечи, Рамешу и Мейеру [9] разработать итерационный метод поиска приближенного максимума критерия Баттачариа, являющийся, по сути, разновидностью метода градиентного спуска. Разработанный метод обладает всеми преимуществами и недостатками градиентных методов. Он достаточно быстр, чтобы обеспечить отслеживание объектов на видеопоследовательностях в режиме реального времени с помощью бюджетного персонального компьютера, но в силу сходимости оценок к локальному экстремуму нередко приводит к потере сопровождаемого объекта.

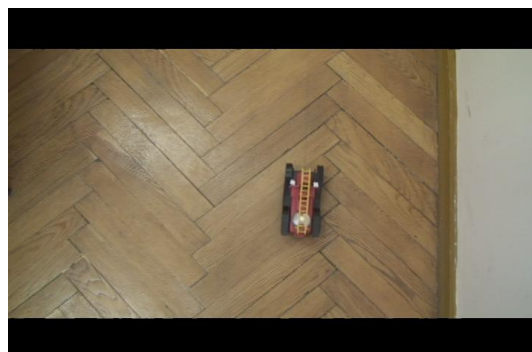
Для повышения точности и надежности сопровождения объектов на основе критерия Баттачариа вместо линеализованного приближения были использованы его исходные представления (2)–(4). Это дало возможность находить глобальный максимум критерия вместо одного из локальных и осуществлять поиск объекта по всему кадру (или по всей разрешенной области), однако значительно увеличило объем вычислений. Поэтому для обеспечения выполнения вычислений в режиме реального времени пришлось разработать параллельные программные реализации алгоритма, в том числе и с применением технологии программирования видеокарты CUDA.

Оптимизированные процессорные программные реализации первой версии алгоритма позволили выполнять отслеживание объектов в режиме реального времени, в то время как для достижения режима реального времени для второй и третьей версий алгоритма нужно было реализовать их на CUDA.

Проведенные исследования показали, что все три версии оказались более точными и надежными по сравнению с классическим алгоритмом среднего сдвига, а вторая версия оказалась на 5–10 % точнее первой из представленных в статье версии.

3. Сравнение характеристик предложенных алгоритмов на основе экспериментов

Для исследования точности и надежности версий алгоритма были использованы видеопоследовательности, полученные видеочамерами с разрешением 720×480 пикселей с борта беспилотного летательного аппарата (БЛА) (рисунок, а), а также снятые нестабилизированной камерой в помещении (рисунок, б). Каждая из последовательностей содержала примерно 500 кадров, представляющих собой 24-битные *RGB*-изображения (в данном случае $L = 256$).



а)

б)

Примеры кадров видеопоследовательностей:

а) полученных с борта БЛА; б) полученных в помещении

Сначала предложенные версии сравнивались с известным алгоритмом среднего сдвига [1]. Алгоритм среднего сдвига (основанный на локальном поиске) в отличие от предложенных версий нередко безвозвратно теряет сопровождаемый объект, что вызывает трудности с автоматическим тестированием, а на корректно обработанных кадрах дает примерно в 1,5 раза менее точный результат, поэтому ниже приведены результаты тестирования только предложенных модификаций.

В табл. 1 приведены результаты тестирования предложенных версий алгоритма на видеопоследовательности, снятой с борта БЛА. Тестирование проводилось на PC Intel® Core TM i5 CPU 750 2,67 GHz (реальная частота 3,6 GHz) с видеокартой NVIDIA GeForce GTX 650Ti. Для

выделения объекта был выбран размер окна 20×20 пикселей. В обоих случаях размер окна выбирался экспериментально таким образом, чтобы получить наилучшие точность и надежность результатов. Оказалось, что оптимальный размер окна совпадает с размером объекта. Экспериментально было установлено, что число градаций квантизованого цвета $l = 8, \dots, 32$ обеспечивает наилучшее соотношение точности и скорости вычислений.

Таблица 1

Версии алгоритма	Критерий	Окно объекта	Кол-во потеря	Среднеквадр. ошибка, пикс.	t_{CPU} , с/кадр	t_{GPU} , с/кадр
<i>Версия 1.</i> Поиск по всему кадру	α	20×20	7	4,88	0,502	0,081
Поиск в окне 100×100	α	20×20	3	4,92	0,015	0,007
<i>Версия 2.</i> Поиск по всему кадру	β	20×20	2	4,36	4,536	1,364
Поиск в окне 100×100	β	20×20	1	4,36	0,141	0,043
<i>Версия 3.</i> Поиск по всему кадру	γ	20×20	14	4,67	1,732	0,258
Поиск в окне 100×100	γ	20×20	6	4,71	0,072	0,017

Объект считался потерянным, если ошибка в определении его координат была больше, чем половина размера окна, т. е. больше 10 пикселей (хотя при этом окно могло частично «накрывать» объект).

Результаты тестирования алгоритма на второй видеопоследовательности приведены в табл. 2.

Таблица 2

Версии алгоритма	Критерий	Окно объекта	Кол-во потеря	Среднеквадр. ошибка, пикс.	t_{CPU} , с/кадр	t_{GPU} , с/кадр
<i>Версия 1.</i> Поиск по всему кадру	α	50×50	7	4,45	2,28	0,358
Поиск в окне 100×100	α	50×50	0	7,12	0,083	0,022
<i>Версия 2.</i> Поиск по всему кадру	β	50×50	12	4,27	48,766	11,601
Поиск в окне 100×100	β	50×50	1	6,44	1,898	0,828
<i>Версия 3.</i> Поиск по всему кадру	γ	50×50	29	7,89	9,399	1,389
Поиск в окне 100×100	γ	50×50	3	10,27	0,526	0,119

При тестировании использовался тот же критерий потери объекта: он считался потерянным, если ошибка в определении его координат была больше, чем половина размера окна, т. е. больше 25 пикселей. Первая и вторая версии алгоритма обеспечили устойчивое сопровождение объектов размером 10×10 пикселей.

Было проведено сравнение точности и надежности разработанных версий алгоритмов с алгоритмами отслеживания объектов корреляционного типа, основанных на сравнении корреляции и ковариации Пирсона [10, 11] на исходных цветных видеопоследовательностях, а также на их полутоновых копиях (были протестированы четыре версии корреляционных алгоритмов).

Корреляционные алгоритмы в отличие от предложенных модификаций алгоритма среднего сдвига оказались неустойчивыми при отслеживании объектов (см. рисунок, *a*): все они теряли объект примерно на половине просмотренных кадров. В данном случае непосредственное использование корреляционных алгоритмов без дополнительных средств стабилизации решения дало бы очень плохие результаты.

На рисунке, *b* видно, что использование корреляции Пирсона на самих изображениях и их полутоновых копиях сделало возможным получение чуть более точных результатов по сравнению с предложенными версиями алгоритма (в среднем на 1,8 пикселя точнее при размере сопровождаемого объекта 100 пикселей), в то время как ковариация дала менее точные и значительно менее устойчивые решения.

Таким образом, при тестировании предложенные модификации оказались более устойчивыми к качеству видеопоследовательностей по сравнению с корреляционными алгоритмами, которые считаются одними из самых надежных при решении задач отслеживания. Кроме того, первая и третья модификации предложенного алгоритма превосходят по быстродействию корреляционные алгоритмы в несколько раз, поэтому легче реализовать их программные версии для работы на CPU и GPU.

Полученные результаты подтверждают возможность применения рассмотренных алгоритмов для решения задач отслеживания объектов на видеопоследовательностях.

Заключение

Предложены три версии алгоритма отслеживания объектов, снятых нестабилизированными видеокамерами, которые основаны на сравнении локальных оконных гистограмм яркостей областей изображений. В качестве меры близости гистограмм в предложенных версиях в отличие от известного алгоритма среднего сдвига использован критерий Баттачария в его исходном нелинейном виде. Вместе с глобальным поиском это позволяет находить вместо ближайшего локального экстремума критерия его гарантированный глобальный максимум, что ведет к повышению точности и надежности отслеживания.

Проведенные эксперименты показали большую точность и надежность предложенных версий по сравнению не только с алгоритмом среднего сдвига, но и с алгоритмами, основанными на применении корреляции и ковариации Пирсона.

Программные реализации версий, выполненные с помощью технологии программирования видеокарты CUDA, оказались настолько быстрыми, что благодаря им можно выполнять отслеживание объектов в режиме реального времени.

Список литературы

1. Cheng, Y. Mean shift, mode seeking, and clustering / Y. Cheng // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 1998. – № 17(8). – P. 790–799.
2. Yilmaz, A. Object tracking: A survey / A. Yilmaz, O. Javed, M. Shah // ACM Computing Surveys. – 2006. – Vol. 38, № 4. – 45 p.
3. Marimon, D. Orientation histogram-based matching for region tracking / D. Marimon, T. Ebrahimi // Proc. 8th Intern. Workshop on Image Analysis for Multimedia Interactive Services WIAMIS. – Santorini, 2007. – P. 8–12.
4. Lowe, D. Object recognition from local scale invariant features / D. Lowe // Proc. Intern. Conf. on Computer Vision ICCV. – Corfu, 1999. – P. 1150–1157.

5. Bay, H. Surf: Speeded up robust features / H. Bay, T. Tuytelaars, L. Van Gool // Proc. 9th Europ. Conf. on Computer Vision ECCV. – Graz, 2006. – P. 404–417.
6. Altmann, J. A Fast Correlation Method for Scale-and Translation-Invariant Pattern Recognition / J. Altmann, H.J. Reitböck // IEEE Trans. Pattern Anal. Mach. Intell. – 1984. – Vol. 6, № 1. – P. 46–57.
7. Object Tracking by Particle Filtering Techniques in Video Sequences / L. Mihaylova [et al.] // Advances and Challenges in Multisensor Data and Information. NATO Security Through Science Series. – Netherlands : IOS Press, 2007. – P. 260–268.
8. Haralick, R.M. Textural Features for Image Classification / R.M. Haralick, K. Shanmugam, I. Dinstein // IEEE Transactions on Systems, Man, and Cybernetics. – 1973. – № 6. – P. 610–621.
9. Comaniciu, D. Real-Time Tracking of Non-Rigid Objects using Mean Shift // D. Comaniciu, V. Ramesh, P. Meer // Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR. – Hilton Head Island, 2000. – Vol. 2. – P. 142–149.
10. Афифи, А. Статистический анализ / А. Афифи, С. Эйзен. – М. : Мир, 1982. – 488 с.
11. Zalesky, B.A. Real Time Object Tracking Algorithm / B.A. Zalesky, E.N. Seredin // Intern. Congress on Computer Science: Information Systems and Technologies, CSIST 2013. – Minsk, 2013. – P. 500–504.

Поступила 11.05.2015

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: zalesky@newman.bas-net.by
eduard.seredin@tut.by*

B.A. Zalesky, E.N. Seredin, M.V. Yadlouski

OBJECT TRACKING VIA COMPARISON OF COLOR HISTOGRAMS

Three versions of a histogram algorithm for tracking objects on video sequences made by an unstable camera are presented. Local color 1D-histograms of pixels and local color 2D-histograms of pairs of adjacent pixels are used in all versions as region features. The histograms are compared by the Bhattacharia criterion. A parallel computing platform CUDA, developed to program GPUs, allows creation of real time or near-real time program realizations of the offered versions. Results of comparison of the versions with the known mean shift algorithm and correlation type algorithms are also presented. It is shown by experiments that the versions are more accurate and reliable than the mean shift algorithm, which estimates similarity of linear approximations of local histograms, and more robust with respect to the video quality than the correlation algorithms.