

УСОВЕРШЕНСТВОВАННЫЙ АЛГОРИТМ МЕДИАННОГО ПОТОКА ДЛЯ ВИЗУАЛЬНОГО ОТСЛЕЖИВАНИЯ ОБЪЕКТОВ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЯХ

В статье рассмотрено ряд усовершенствований алгоритма медианного потока, предназначенного для визуального отслеживания объектов. Суть усовершенствований сводится к адаптивному выбору размера окна апертуры, количества уровней пирамиды и положения анализируемых точек при оценивании разреженного оптического потока. Это позволяет повысить эффективность отслеживания по сравнению с базовым алгоритмом, особенно при работе с малыми объектами. На реальных видеопоследовательностях проведен анализ улучшенного медианного потока. Полученные результаты показали универсальность и невысокую вычислительную требовательность алгоритма, что делает его перспективным для применения во встраиваемых системах.

The improvements of a median flow algorithm, which meant for visual object tracking, are considered. The improvements consist in adaptive selection of aperture size, number of pyramid levels and locations of points used during the sparse optical flow estimation. This permits to increase the efficiency of tracking as compared with the basic algorithm, especially when dealing with small objects. Analysis of improved median flow was performed on real video sequences. The obtained results showed the versatility, robustness and moderate computational demands of the algorithm, which makes it promising for application in embedded systems.

1. Введение

На сегодняшний день алгоритмы визуального отслеживания находят широкое применение в медицине, человеко-машинных интерфейсах, системах слежения и видеоаналитике, робототехнике, что собственно и обуславливает их высокую актуальность.

С точки зрения реализации алгоритмов отслеживания, большой интерес представляют недорогие и надежные встраиваемые процессоры, а также системы на кристалле (SoC). При этом, однако, их вычислительных возможностей зачастую недостаточно для реализации большинства сложных алгоритмов. Таким образом, необходимость в поиске, надежных и одновременно быстродействующих алгоритмов отслеживания по-прежнему сохраняется.

Даная работа направлена на решение обозначенной задачи. Здесь анализируется алгоритм, называемый медианным потоком, впервые описанный в работе [1]. В ходе испытаний было установлено, что этот алгоритм обладает компромиссными возможностями в отношении быстродействия и надежности, однако последняя может быть несколько увеличена. Таким образом, цель данной работы как раз и заключается в исследовании подходов позволяющих повысить надежность алгоритма медианного потока.

В последующих разделах рассматривается: базовый алгоритм, его преимущества и недо-

статки (раздел 2); простые эвристические подходы, направленные на улучшения надежности медианного потока (раздел 3); усовершенствованные способы повышения надежности алгоритма (раздел 4); экспериментальная апробация и сравнительный анализ базового, простого эвристического и усовершенствованного подходов (разделы 5, 6).

2. Исходный алгоритм медианного потока и его недостатки

Суть алгоритма медианного потока заключается в оценивании на смежных кадрах видеопоследовательности положения объекта с помощью разреженного оптического потока [1]. При этом считается, что объект состоит из небольших жестко связанных областей, которые при его перемещении движутся синхронно (согласованно).

Инициализация алгоритма осуществляется следующим образом: задается прямоугольник, в котором находится объект слежения и в пределах этого прямоугольника генерируется разреженная координатная сетка (обычно размера 10×10 обычно достаточно). Далее точки сетки отслеживаются между предыдущим и текущим кадром, для чего используется пирамидальный алгоритм оптического потока Лукаса-Канаде [4, 5].

С целью повышения надежности алгоритма в нем предусмотрено отсеивание неточных ре-

зультатов. Для этого применяется двухуровневая фильтрация: во-первых, отбрасываются соответствующие точки смежных кадров, для областей которых значение нормализованной корреляции (NCC) не меньше заданного порога, а во-вторых, те точки – чьи ошибки прямого-обратного распространения (FB-ошибки) выше заданного значения. При этом под ошибкой прямого-обратного распространения подразумевается евклидово расстояние между соответствующими точками, отслеженными сначала в прямом направлении – от предыдущего к текущему кадру, а потом в обратном – от текущего кадра к предыдущему.

Сдвиг объекта оценивается как медианное значение разностей между координатами соответствующих точек текущего и предыдущего кадров. При этом в рассмотрение принимаются только те точки, которые были признаны надежными во время фильтрации. Кроме того существует возможность оценить изменение масштаба объекта. Для этого необходимо найти расстояния между парами точек объекта на предыдущем и текущем кадрах, рассчитать их медианное значение в пределах каждого из кадров, а далее определить соотношение между полученными медианами. Более детально см. [1].

Следует выделить следующие недостатки базового алгоритма медианного потока:

- оптический поток рассчитывается для точек с неизменным размером апертуры (окна) равным 4×4 пикселя и количеством уровней пирамиды равным 5 (см. реализацию метода TLD в состав которого входит алгоритм медианного потока [2, 11]). Однако такая комбинация не является универсальной и не позволяет надежно отслеживать, к примеру, малые объекты;
- на каждом кадре в области объекта генерируется разреженная сетка точек, которые далее подлежат отслеживанию. В то же время авторы алгоритма [1] отказались от механизмов предотвращения попадания точек сетки на непригодные для работы области (к примеру, с однородной яркостью) в пользу оценивания FB-ошибки [1, 2, 11]. В то же время такое решение не является оптимальным как с точки зрения количества вычислений, так и в отношении надежно-

сти: зачем отслеживать точки которые заведомо дадут неверный результат.

3. Первоначальные попытки улучшения алгоритма медианного потока

Экспериментальные исследования показали, что изменение параметров описанного выше алгоритма, в некоторых случаях, позволяет обеспечить более надежное отслеживание. Исходя из этого, первоначально, были предложены эвристические подходы, которые позволяют повысить эффективность работы базового алгоритма. Коротко приведем основные их положения.

В пирамидальном алгоритме Лукаса-Канаде объект на больших уровнях пирамиды, ввиду сильного уменьшения, может вырождаться в точку или вообще исчезнуть, что, очевидно, приведет к ошибочному определению для него оптического потока. Не надежный результат отслеживания также имеет место и при недостаточном количестве уровней пирамиды. Это особенно характерно для быстро перемещающихся, содержащих плавные переходы яркости объектов. Аналогичным образом на качестве отслеживания сказывается размер апертуры (размер окна, применяемый для оценивания перемещения каждой точки).

Таким образом, количество уровней пирамиды и размер окна было решено сделать адаптивными. В частности количество уровней пирамиды было ограничено так, что при масштабировании объект не становился меньше размера окна (апертуры). Учитывая, что на смежных уровнях пирамиды масштаб изображений отличается в два раза [3], было предложено следующее простое условие:

$$\frac{\min(w, h)}{2^L} \geq 2\omega + 1 \quad (1)$$

$$L \leq \log_2 \frac{\min(w, h)}{2\omega + 1}$$

где w и h – соответственно ширина и высота описывающего объект прямоугольника; L – количество уровней пирамиды; $2\omega + 1$ – размер окна (апертуры). В соответствии с рекомендациями, приведенными в работе [3], имеет смысл брать $L \leq 4$.

Выражение (1) применимо, если известна величина ω . Изначально предложено выбирать ω непосредственно на основе размеров объекта. Экспериментально было установлено, что при-

емлемый результат достигается в случае использования квадратных окон со значением ω от 3×3 до 7×7 пикселей, причем выбор осуществлять так, чтоб не было значительного перекрытия апертур. При этом использовалась следующая эмпирическая формула:

$$2\omega + 1 = \frac{h + w}{2D}, \quad (2)$$

где $D \times D$ – размер разреженной сетки точек, подлежащих отслеживанию; $(h + w) / 2$ – средний габаритный размер объекта. Здесь размер апертуры представлен в виде выражения $2\omega + 1$ в связи с удобством использования параметра ω – именно он передается в функцию вычисления оптического потока библиотеки OpenCV версии до 2.3 включительно.

Хотя выражение (2) дает лучшие результаты в сравнении с исходным подходом, оно остается не полностью оптимальным, поскольку никоим образом не учитывает характер и размер перепадов яркости присущих объекту. Поэтому, в следующем разделе предлагается новый способ оценивания размера апертуры.

Для оптического потока характерна еще одна сложность, связанная с апертурой. В литературе ее так и называют «проблемой апертуры» [4, 5], и заключается она в невозможности надежного определения оптического потока для всех точек изображения. Для решения этой проблемы в исходной работе [1] рассматривалась возможность использовать процедур отбора областей удачных для отслеживания [5, 6]. Однако авторы впоследствии от них отказались [2, 11], вероятно ввиду неудобства прямого их использования: оба подхода основаны на пороговой обработке и неверный выбор порога зачастую не обеспечивает достаточное для надежного отслеживания объекта количество точек. Более того подход описанный в [5] ресурсоемок и не может быть использован для работы во встраиваемых системах. В связи с этим предлагается простой способ отбора признаков. Он основан на разбиении области объекта слежения на небольшие неперекрывающиеся фрагменты, в каждом из которых производится поиск границ и выбирается по одной точке, которая в данном фрагменте соответствует границе с наибольшей амплитудой. Детектирование границ можно эффективно выполнять, с помощью операторов Собела или Превита. Подобную схему, также несложно адаптировать к быстрому способу определения углов «FAST»

[6]. Для этого, углы следует искать, используя небольшой порог (к примеру, равный единице) и среди найденных ключевых точек в каждом фрагменте выбрать по одной – с максимальной оценкой (score). При этом указанная схема на практике гарантирует определение в каждом из фрагментов по одной точке подлежащей отслеживанию. Отметим, что в следующем разделе данной работы предлагается еще один быстрый способ поиска оптимальных для отслеживания точек.

Базовый алгоритм медианного потока переинициализируется на каждом кадре [2], то есть генерируется новая разреженная сетка точек подлежащих отслеживанию. Однако замечено, что более надежный результат достигается при повторном использовании тех точки, которые не были отфильтрованы с помощью вычисления НСС, FB-ошибки, и не оказались за пределами прямоугольника охватывающего объект. При этом отброшенные точки несложно восстановить при помощи процедур, описанных выше.

4. Дальнейшее совершенствование алгоритма медианного потока

4.1. Поиск оптимального размера апертуры

Очевидно, что оптический поток будет надежнее вычисляться тогда, когда в апертуры каждой из точек попадут достаточно большие перепады яркости. Это означает, что размеры апертур следует выбирать так, чтоб энергия перепадов яркости в их пределах была максимальной. При этом величину перепада можно оценить с помощью пространственных градиентов, а оптимальный размер – искать исходя из анализа их энергий на разных масштабах. Масштаб, на котором энергия градиентов окажется максимальной, вероятно и будет соответствовать апертуре с наибольшими перепадами яркости.

Наиболее простой и очевидный способ оценить энергии градиентов на разных масштабах – вычислять квадраты разностей между точками изображения, взятыми на различных расстояниях друг от друга, то есть:

$$\begin{aligned} E_{\partial I / \partial x}(d) &= [I(x, y) - I(x + d, y)]^2 \\ E_{\partial I / \partial y}(d) &= [I(x, y) - I(x, y + d)]^2 \end{aligned} \quad (3)$$

где $E_{\partial I / \partial x}(d)$ и $E_{\partial I / \partial y}(d)$ – энергии горизонтальных и вертикальных градиентов в точке (x, y) соответственно; d – расстояние между точками

изображения, соответствующие масштабу на котором вычисляется градиент; $I(x, y)$ – яркость изображения объекта в точке (x, y) .

Таким образом, значения d , для которых суммарные по всему изображению энергии градиентов максимальны, должны соответствовать оптимальным размерам апертуры. Это можно записать следующим образом:

$$2\omega_x + 1 = \arg \max_d \sum_{y=1}^{M-d} \sum_{x=1}^{N-d} [I(x, y) - I(x+d, y)]^2, \quad (4)$$

$$2\omega_y + 1 = \arg \max_d \sum_{y=1}^{M-d} \sum_{x=1}^{N-d} [I(x, y) - I(x, y+d)]^2$$

где $2\omega_x + 1$ и $2\omega_y + 1$ – ширина и высота апертуры соответственно; N и M – ширина и высота изображения соответственно.

Поскольку, замена квадрата на абсолютное значение в приведенных выше формулах для большинства случаев значительно не повлияет на поведение функции $\arg \max$ по отношению к d , то выражения (4) можно представить в более эффективном для вычисления виде:

$$2\omega_x + 1 = \arg \max_d \sum_{y=1}^{M-d} \sum_{x=1}^{N-d} |I(x, y) - I(x+d, y)| \quad (5)$$

$$2\omega_y + 1 = \arg \max_d \sum_{y=1}^{M-d} \sum_{x=1}^{N-d} |I(x, y) - I(x, y+d)|$$

Процедуру оценивания размера апертуры иллюстрирует рис. 1, на котором предложенный подход фактически определил средний размер пятна, который, очевидно, оптимален для отслеживания.

С целью повышения быстродействия, предлагается изменять расстояние d с шагом $\Delta > 1$. При этом, однако, важно знать: не будет ли при большом шаге Δ пропущено расстояние, соответствующее глобальному максимуму энергии градиентов. В общем случае такая ситуация не исключена и может иметь место тогда, когда на изображении максимумы яркостных перепадов разнесены так, что ни одно значение $d_i = d_0 + i\Delta$; $i = 1, 2, 3, \dots$; $\Delta > 1$ не будет соответствовать истинному расстоянию между ними (см. рис. 2 – для простоты приведен одномерный случай).

На самом деле пропуск максимума энергии градиентов для узких разнесенных перепадов не является критичным. Дело в том, что узкие перепады яркости, характерны для узких линий или точек, которые вполне могут отслеживать-

ся самостоятельно. Это существенно не ухудшит надежность, в то же время потребует меньших размеров апертур, а соответственно и – вычислительных затрат.

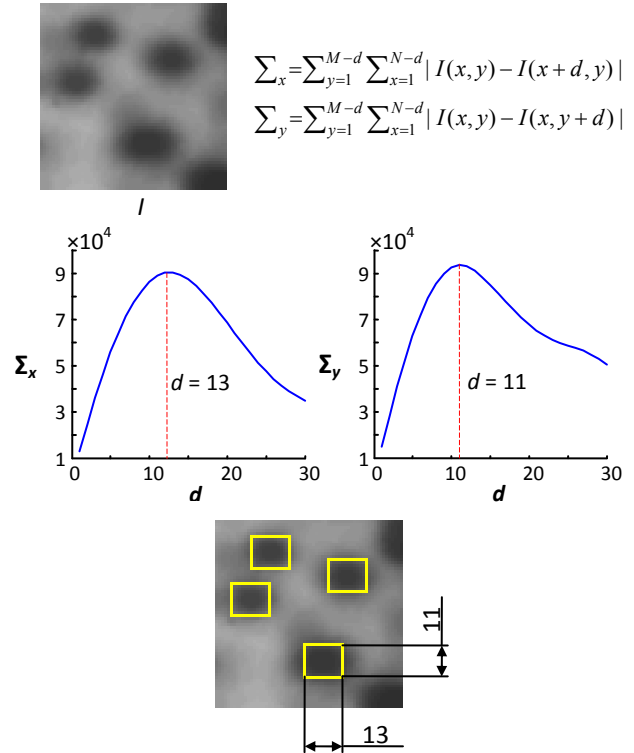


Рис. 1. Поиск оптимальной апертуры для изображения I. Кривые показывают изменение суммарной по изображению энергии горизонтальных (левая кривая) и вертикальных (правая кривая) градиентов. Пунктирной линией обозначены расстояния d , на которых достигаются максимумы энергий. Для наглядности под кривыми приведено изображение с нанесенными на нем прямоугольниками вокруг темных пятен. Как видно размеры этих прямоугольников равны максимумам d и приблизительно соответствуют действительным ширине и высоте пятен.

Широкие, расположенные близко друг к другу перепады, наиболее часто принадлежат нечеткими границами объектов, поэтому их целесообразнее отслеживать как единое целое. Примечательно, что функции $E_{\text{all}\partial x}(d)$ и $E_{\text{all}\partial y}(d)$ (в виду большой ширины перепадов яркости) будут изменяться плавно вблизи точек максимума. Таким образом, для их уточнения ниже предлагается простая процедура последовательного приближения.

Суть данной процедуры сводится к следующему: проходя по изображению с большим шагом d и найдя точку максимума энергии гради-

ентов d_{\max} , произвести более детальный анализ в ее окрестности. Для этого убедиться, что в точке d_{\max} энергия градиентов выше чем, точках $d_{\max} - \Delta/2$ и $d_{\max} + \Delta/2$ (для простоты можно просто анализировать ближайшие соседние точки: $d_{\max} - 1$ и $d_{\max} + 1$). Если это окажется не так, то продолжить поиск максимума в окрестности той из точек, которой будет соответствовать наибольшее значение энергии. При этом, если $E(d_{\max} - \Delta/2) = E(d_{\max} + \Delta/2)$, но превышают $E(d_{\max})$, то дальнейший анализ предпочтительнее проводить в окрестности $d_{\max} - \Delta/2$, так как это даст меньшую апертуру и соответственно большее быстродействие при вычислении оптического потока. Приведенная схема проиллюстрирована на рис. 3.

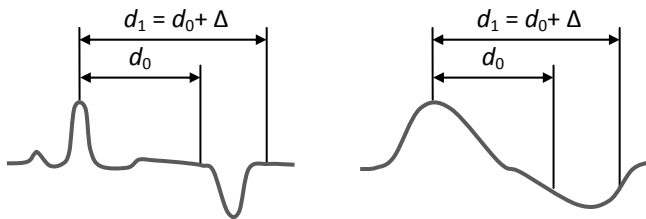


Рис. 2. Условия пропуска максимума энергии градиента при $\Delta > 1$. Рассмотрено два случая: слева – резкие небольшие по ширине перепады, характерны для мелких точек или узких линий, далеко отстоящих друг от друга; справа – широкие перепады, которые часто свойственны нечетким границам объектов.

Как говорилось ранее, значения ω целесообразно брать в пределах от 3 до 7. Нижняя граница здесь определяется требованиями алгоритма оценивания оптического потока [3] (см. реализацию OpenCV), а верхняя выбрана исходя из соображений быстродействия на встраиваемых процессорах. В целом на более мощных компьютерах верхняя граница может быть несколько увеличена, но ее все же не имеет смысла брать больше 15 пикселей. В связи с указанными ограничениями, а также с тем, что апертура представляется как $2\omega + 1$, поиск оптимального ее размера с помощью (5) необходимо проводить только в пределах d от 7 до 31. Учитывая, что для вычисления ω потребуется деление на 2 и округление, d можно изменять с шагом $\Delta = 2$. Наличие процедуры последовательного приближения, также позволяет незначительно увеличить шаг. Таким образом, поиск максимумов энергий градиентов можно выпол-

нить по d с шагом $\Delta = 4 \div 5$, а это потребует в наихудшем случае порядка $(31 - 7) / 4 + 4 = 10$ (последнее слагаемое определяет расходы на приближение) вычислений сумм в формулах (5).

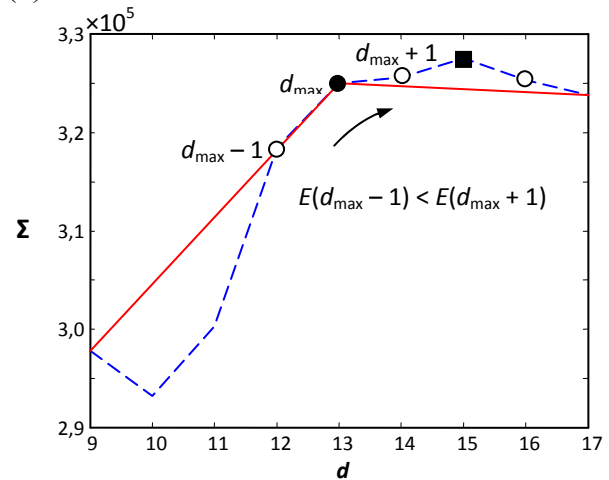


Рис. 3. Поиск максимума путем последовательного приближения, если энергия градиентов первоначально оценивалась с шагом изменения d большим единице. Кривая, показанная сплошной линией – значения энергий на расстояниях d взятых с шагом $\Delta = 4$. Пунктирная кривая – энергии градиентов для d взятых с шагом $\Delta = 1$. Черным круглым маркером обозначена точка соответствующая максимуму энергии на кривой, для которой d изменяется с $\Delta = 4$. Черным квадратным маркером обозначен глобальный максимум. Стрелка указывает направления поиска максимума.

Следует заметить, что предложенный подход оценки размера апертуры более чувствителен к перепадам яркости, обладающим большим пространственным размером. То есть, если на объекте присутствует перепад яркости, покрывающий значительную его площадь (это, к примеру, может быть тень), то в таком случае, очевидно, максимум энергии градиентов будет достигаться при d близком к размеру данного перепада, что нежелательно. Указанный эффект частично можно скомпенсировать, если разбить область объекта на фрагменты, далее в каждом из них произвести анализ размера апертуры, а полученный результат усреднить. При этом если имеется глобальный перепад, на однородных затемненных или осветленных им фрагментах будут оценены локальные перепады, а его влияние скажется только на участках попа-

дающих непосредственно на глобальную линию раздела яркости. После усреднения значение d будет частично скомпенсировано, как показано на рис. 4.

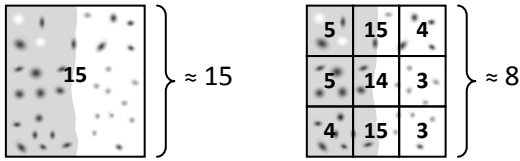


Рис. 4. Глобальный перепад яркости и его компенсация. На изображении слева максимум энергии из-за глобального перепада достигается при $d \approx 15$, в то же время оценивание энергии во фрагментах, показанных на рисунке справа дает результат более близкий к истинному: $d = 7,56 \approx 8$.

Таким образом, изложенный выше подход определения апертуры, основанный на выражении (5), призван заменить непосредственное оценивание по размеру, задаваемое уравнением (2). При этом, количество уровней пирамиды, как и прежде можно рассчитать по формуле (1).

4.2. Выбор точек подлежащих отслеживанию

Как уже отмечалось выше для решения проблемы апертуры в алгоритме медианного потока предложено два подхода, и оба они не лишены недостатков. Первый из них ищет максимум амплитуды границы, но не учитывает ее форму, что не устраняет проблему апертуры на ровных протяженных линиях. Второй подход призван анализировать оценки качества угловых ключевых точек FAST [6], однако как отмечают сами авторы, FAST все же может детектировать под некоторыми углами линии шириной в 1 пиксель, что опять же не позволяет полностью решить проблему апертуры. В связи с этим в данной работе предлагается еще один простой подход поиска ключевых точек, основанный на так называемом усовершенствованном цензурном преобразовании (MCT – Modified Census Transform) [7].

Его суть заключается в следующем: для областей изображения имеющих размер 3×3 пикселя и центр в точке с координатами x рассчитывается среднее значение $\bar{I}(x)$, которое далее сравнивается с каждым пикселем данной области, формируя в результате двоичную последовательность. Эта двоичная последовательность может быть представлена в виде 9-битного чис-

ла $\Gamma(x)$ и рассматривается как некоторый признак, кодирующий область изображения в точке x . Формально вычисление MCT признака определяется как:

$$\Gamma(x) = \otimes_{y \in \Omega(x)} \zeta(I(y) - \bar{I}(x), 0), \quad (6)$$

где $\Omega(x)$ – окно задающее область с центром в точке x ; символ \otimes – обозначает операцию конкатенации; $\zeta(a, b)$ – оператор сравнения, равный «1» если $a > b$ и «0» – в противном случае; $\bar{I}(x)$ – средняя яркость в области $\Omega(x)$; $I(y)$ – яркость пикселя в точке y . Процесс вычисления MCT иллюстрирует рис. 5 [8].

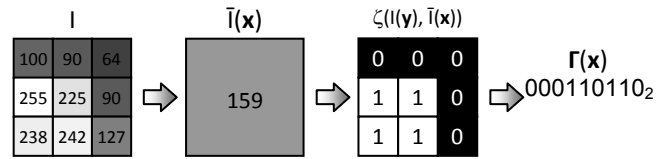


Рис. 5. – Процедура вычисления признака усовершенствованного цензурного преобразования (MCT)

Поскольку, надежному отслеживанию поддаются области, содержащие яркостные пятна, углы, края, пересечения и изгибы линий [4], именно их следовало бы использовать в качестве ключевых точек. Поиск таких областей несложно осуществить, используя специальный алфавит MCT-признаков, пример которого приведен на рис. 6.

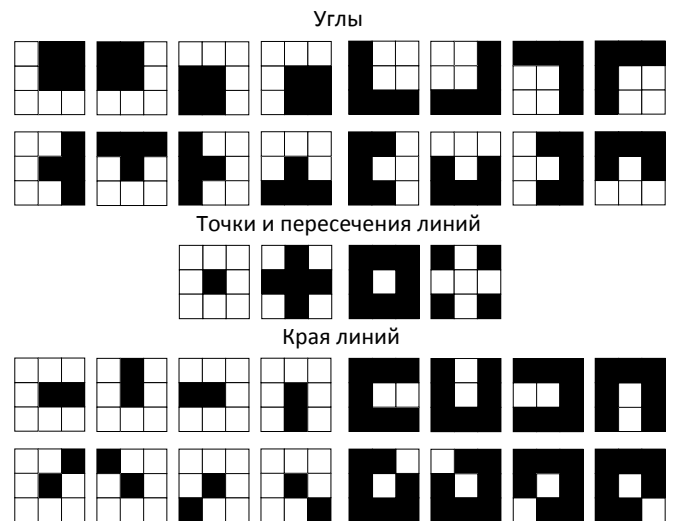


Рис. 6. Пример неполного алфавита MCT-признаков, определяющих отдельные точки, углы, края и пересечения линий.

Данным алфавитом заданы наиболее надежные для отслеживания области, однако он является неполным и при необходимости может быть расширен дополнительными признаками. Отметим, что отбор ключевых точек из

всего множества МСТ-признаков можно очень эффективно осуществлять с помощью таблиц поиска. Размер подобной таблицы для областей 3×3 пикселя составит $2^9 = 512$ элементов.

Детектор на основе МСТ-признаков, однако, чувствителен к шуму, поэтому для обнаруживаемых ключевых точек важно иметь определенную меру качества $S(\mathbf{x})$. Для ее вычисления, целесообразно использовать отклонение от средней яркости:

$$S(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega(\mathbf{x})} |I(\mathbf{y}) - \bar{I}(\mathbf{x})|. \quad (7)$$

Преимущество рассмотренного подхода перед детектором FAST заключается в том, что он в отличие от FAST анализирует область каждой из точек изображения полностью и соответственно не детектирует узкие протяженные линии.

Таким образом, в контексте рассматриваемого алгоритма медианного потока предлагается использовать описанный детектор ключевых точек совместно с подходом по оцениванию апертуры следующим образом:

- 1) Произвести поиск оптимального размера апертуры и определить размер преобладающих на объекте сильных перепадов яркости. Для этого пользуясь выражением (5) найти оптимальные значения $2\omega_x + 1$ и $2\omega_y + 1$. С целью экономии вычислительных ресурсов в уравнении (5) параметр изменять d с шагом $(4 \div 5)$, после чего выполнить процедуру последовательного приближения.
- 2) Путем масштабирования объекта привести его размер к такому, чтоб найденные на нем перепады по ширине и высоте оказались сравнимы с размером областей анализа ключевых точек (для рассмотренного подхода на основе МСТ-признаков этот размер составляет 3×3 , для ключевых точек FAST – 7×7).
- 3) Разбить масштабированный объект на фрагменты в соответствии с требуемой плотностью разреженной сетки, то есть на $D \times D$ ячеек (см. уравнение (2)). В каждой из ячеек произвести поиск ключевых точек и выбрать по одной с наибольшим значением оценки (для предложенного подхода на основе МСТ-признаков оценка вычисляется с помощью (7)).

5. Результаты тестирования алгоритма

Для того чтобы убедиться, что усовершенствованный алгоритм является более надежным чем его базовая версия [1], было проведено несколько простых тестов: на трех видеопоследовательностях обозначенных как «David», «Car4» и «Mirage» (показаны на рис. 7) были определены опорные положения объектов. Далее данные положения были сопоставлены с результатами, полученными с помощью каждого из алгоритмов. Реализация базового алгоритма, при этом, взята из оригинальной системы TLD, в которой был предварительно отключен компонент детектирования и обучения. Адаптация масштаба была отключена для всех алгоритмов.

В таблице 1 приведены данные относительно количества потерь объекта каждым из алгоритмов. Факт потери устанавливался на основании площади перекрытия прямоугольников, которые задают положения объекта на опорных данных и полученных с помощью тестируемых алгоритмов:

$$A = |BB \cap BB_{ref}|, \quad (8)$$

где BB – множество точек прямоугольника найденного системой слежения; BB_{ref} – множество точек опорного прямоугольника; $|\cdot|$ – оператор дающий количество точек множества. Объект считался потерянным, если площадь перекрытия меньше 50% площади опорного прямоугольника.

Табл. 1. Количество потерь алгоритмом объекта

Алгоритм	Последовательность		
	David	Car4	Mirage
Базовый [1]	0	3	10
Простой эвристический	0	0	0
Улучшенный	0	0	0

Как видно из таблицы 1, для усовершенствованного алгоритма на тестовых видеопоследовательностях, в соответствии с выбранным критерием, не имели место потери объектов. При этом оценивание FB-ошибки в усовершенствованном алгоритме и в первоначальном эвристическом алгоритме не проводилось.

Табл. 2 содержит информацию о среднем отклонении от опорных координат каждым из алгоритмов.

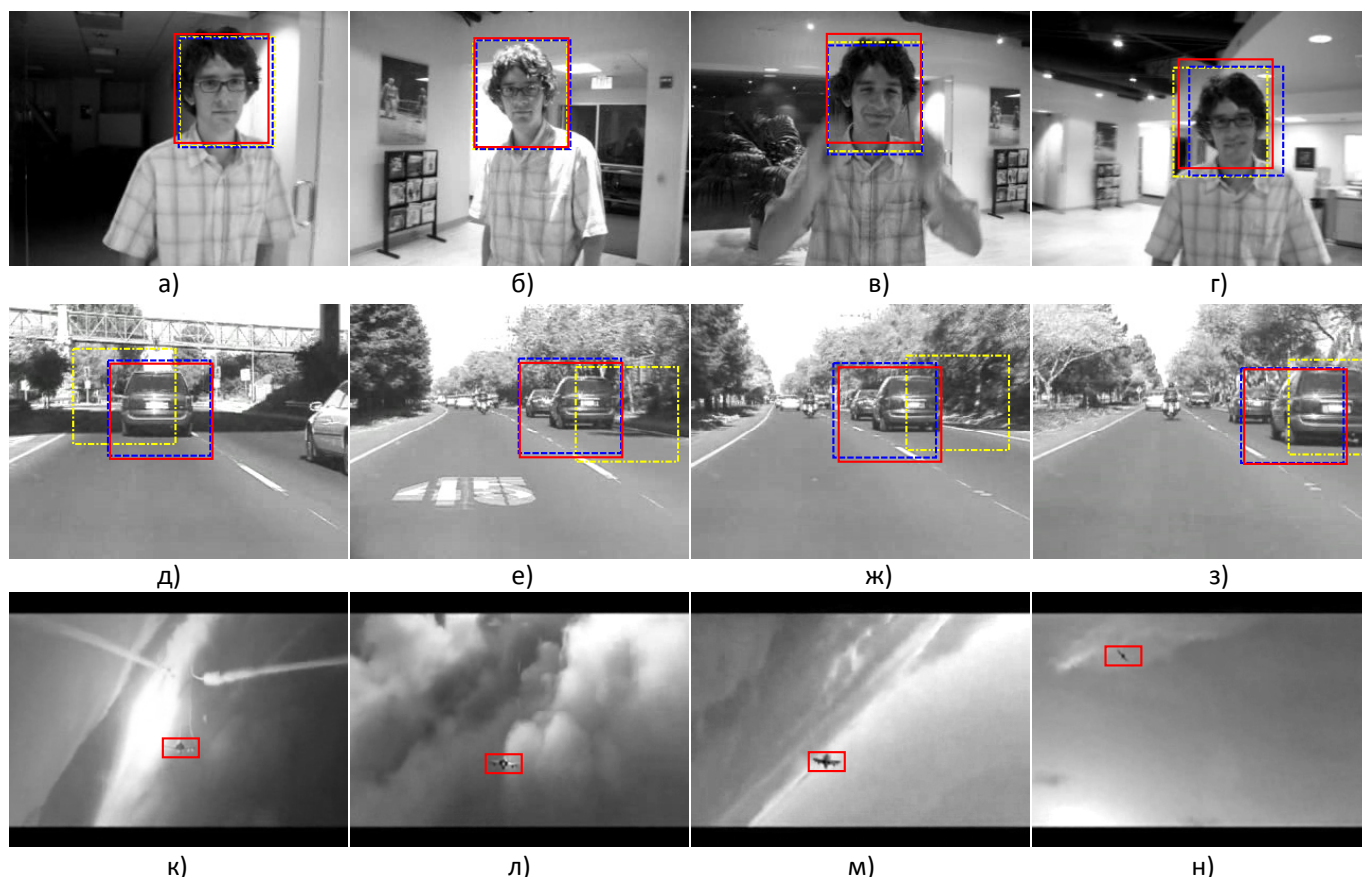


Рис. 7. Пример отслеживания различных объектов на видеопоследовательностях (показаны выбранные кадры): а)-г) – последовательность «David»; д)-з) – последовательность «Car4»; к)-н) – последовательность «Mirage». На рисунках прямоугольниками обозначены результаты: со штрихпунктирной линией – полученные с помощью базового алгоритма, с пунктирной линией – полученные простым эвристическим алгоритмом из раздела 3 и со сплошной линией – полученные усовершенствованным алгоритмом из этой работы.

Табл. 2. Средняя ошибка отклонения координат объекта в пикселях

Алгоритм	Последовательность		
	David	Car4	Mirage
Базовый [1]	7,96	11,9	2,47
Простой эвристический	6,27	16,36	1,63
Улучшенный	4,77	13,47	1,49

Из нее также видно, что последняя модификация алгоритма имеет наименьшую ошибку.

Приведенные выше результаты получены в среде MATLAB версии 7.14.0.739 (2012a) к которой были подключены библиотеки OpenCV 2.1, вычисляющие пирамидальный оптический поток. Тестирование проводилось на компьютере с процессором Intel Core Duo с тактовой частотой 1,86 ГГц, 2 Гб ОЗУ. Время обработки кадра с разрешением 320×240 пикселей составило в среднем 4,5 мс для объекта размером 97×90 пикселей. Слежение осуществлялось с использованием разреженной сетки 10×10 точек, параметр размера окна апертуры ω алгоритмом был определен равным 6, количе-

ство уровней пирамиды L – равным 2. Данные приведены для последовательности «Car4» (см. рис. 7), которая является наиболее сложной среди рассматриваемых в данной работе.

Учитывая высокие показатели быстродействия рассмотренного алгоритма (обработка кадра составляет менее 5 мс на ПК общего назначения), можно ожидать, что он сможет работать в режиме близком к реальному времени на современных встраиваемых ARM процессорах и системах на кристалле, таких как: TI OMAP, NVidia Tegra, Qualcomm Snapdragon и подобных.

6. Обсуждения результатов

Из приведенных выше результатов следует, что реализация усовершенствованного алгоритма имеет наименьшую среднюю ошибку отклонения координат от опорного положения объекта на двух тестовых роликах. При этом, однако, наилучший результат, полученный на основе базового алгоритма на последователь-

ности «Car4», следует считать корректным лишь формально. Причиной этого являются переинициализации, имевшие место при потере объекта (см. табл. 1), не позволившие накапливать погрешность при определении его координат.

Следует отметить также, что возможно использование более надежных и точных алгоритмов оценивания апертуры, основанных, например, на многомасштабном анализе качества ключевых точек [9] или на разности изображений размытых фильтрами Гаусса (DoG) [10]. Однако, при ознакомлении с ними детальнее несложно убедиться, что они уступают по сложности реализации и быстродействию рассмотренному выше алгоритму анализа энергий градиентов.

7. Выводы

В данной работе автором усовершенствован алгоритм медианного потока, предназначенный для автоматического отслеживания объектов на

видеопоследовательностях. В частности, предложены способы оптимального определения размера апертуры путем анализа энергий градиентов и быстрого поиска ключевых точек с помощью алфавита МСТ-признаков. Указанные решения позволили улучшить надежность и повысить точность вычисления оптического потока и, как следствие, эффективность рассматриваемого алгоритма слежения, особенно при работе с малыми объектами. Достоверность предложенных усовершенствований подтверждена экспериментальной апробацией на реальных видеопоследовательностях.

Учитывая, что усовершенствованный алгоритм обладает высоким быстродействием и достаточной надежностью, его можно считать перспективной процедурой отслеживания объектов для реализации во встраиваемых системах.

Список литературы

1. Kalal Z., Mikolajczyk K., Matas J. Forward-Backward Error: Automatic Detection of Tracking Failures // International Conference on Pattern Recognition. – 2010. – P. 23–26.
2. Kalal Z., Mikolajczyk K., Matas J. Tracking-Learning-Detection. // Transactions on Pattern Analysis and Machine Intelligence. – 2012– Vol. 34(7). – P. 1409–1422.
3. Boguet J.-Y. Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm. // Tech. report, Intel Corporation, Microprocessor Research Labs. – 2002. – 9 P.
4. Bradski G., Kaehler A. Learning OpenCV. – Sebastopol: O'Reilly Media, Inc. – 2008. – 557 P.
5. Shi J., Tomasi C. Good features to track. // Proc. Conference on Computer Vision and Pattern Recognition. – 1994. – P. 593–600.
6. Rosten E., T. Drummond. Machine learning for high-speed corner detection. // European Conference on Computer Vision. – 2006. – P. 430–443.
7. Küblbeck C., Ernst A. Face detection and tracking in video sequences using the modified // Image and Vision Computing. – 2006. – Vol. 24(6), P. 564–572.
8. Varfolomeiev A., Dzyuba V., Lysenko O. Implementation of a Real-Time Object Detection System on a TI TMS320DM6437 DSP // Mediterranean Conference on Embedded Computing. – 2012. – P. 108–111.
9. Lowe D. G. Object Recognition from Local Scale-Invariant Features // International Conference on Computer Vision. – 1999. – Vol. 2. – P. 1150–1157.
10. Mair E., Hager G. D., Burschka D., Suppa M., Hirzinger G. Adaptive and generic corner detection based on the accelerated segment test. // Proceedings of the European Conference on Computer Vision. – 2010. – P. 183–196.
11. Исходный код реализации метода TLD [электронный ресурс], режим доступа: <https://github.com/zk00006/OpenTLD> – проверен 25.09.2012.