ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 57 за 2018 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Беклемишев Н.Д.

Алгоритм межкадрового прослеживания подвижного объекта

Рекомендуемая форма библиографической ссылки: Беклемишев Н.Д. Алгоритм межкадрового прослеживания подвижного объекта // Препринты ИПМ им. М.В.Келдыша. 2018. № 57. 15 с. doi:10.20948/prepr-2018-57
URL: http://library.keldysh.ru/preprint.asp?id=2018-57

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

Н.Д.Беклемишев

Алгоритм межкадрового прослеживания подвижного объекта

Москва — 2018

Беклемишев Н.Д.

Алгоритм межкадрового прослеживания подвижного объекта

В работе предлагается алгоритм межкадрового прослеживания подвижного объекта на видеопоследовательности, использующий вычисления на основе быстрого преобразования Фурье.

Ключевые слова: видеотрекинг, удержание объектов, БПФ

Nikolai Dmitrievich Beklemishev

The algorithm for the video tracking of the moving object

We offer an algorithm for the video tracking of the moving object with the use of calculations on the basis of the fast Fourier transform.

Key words: video tracking, FFT

Работа выполнена в рамках реализации Государственного задания на 2018 г. - № 0017-2018-0008.

Оглавление

Постановка задачи межкадрового прослеживания объекта

Межкадровое прослеживание (удержание) объектов, или видеотрекинг – процесс оценки положения одного или нескольких объектов в течение времени с помощью видеокамеры. Ускоренное развитие вычислительных средств привело к созданию новых алгоритмов удержания объектов и к появлению новых приложений таких алгоритмов [1 – 3].

Мы рассматриваем ситуацию, когда в кадре имеется, вообще говоря, подвижный «фон», занимающий большую часть кадра, и заранее выделенный объект, подвижный относительно фона. Рассматривается задача межкадрового прослеживания данного объекта. При этом в настоящей работе мы не касаемся таких проблем как клаттер (толчея объектов), когда требуется выделить данный объект среди нескольких близко расположенных или частично перекрывающихся объектов [2].

Мы рассматриваем два последовательных кадра видеоряда. При этом мы предполагаем, что преобразование, задающее совмещение фона двух последовательных кадров, нам известно. Преобразование с первого кадра на второй кадр, совмещающее фон между кадрами, предполагается проективным и задается матрицей преобразования M размера 3x3. На изображениях имеется подвижный относительно фона объект, занимающий на первом кадре некоторую область D. Задача удержания объекта сводится к тому, чтобы выделить на втором кадре область D', которую занимает сместившийся объект.

На первом кадре мы рассматриваем окрестность Ω области D. Это квадратный участок растра, с центром в центре тяжести области D и целиком содержащий эту область. Размер окрестности определяется скоростью перемещения объекта относительно фона и задается таким образом, чтобы объект на втором кадре не выходил за пределы области $M(\Omega)$, где M – преобразование, совмещающее фон.

Пусть $\boldsymbol{x} \in \Omega$ – точка окрестности, $f_1(\boldsymbol{x})$ – уровень яркости первого кадра в точке $\boldsymbol{x}$. Определим также в точках окрестности Ω функцию $f_2(\boldsymbol{x})$ как уровень яркости второго кадра в точке $M(\boldsymbol{x})$. При этом для точек фона должно выполняться приближенное равенство $f_1(\boldsymbol{x}) \approx f_2(\boldsymbol{x})$, поскольку $M(\boldsymbol{x})$ – преобразование, совмещающее фон.

Уровень яркости точек объекта на первом кадре задается значениями $f_1(\boldsymbol{x})$, $\boldsymbol{x} \in D$. Уровень яркости точек объекта на втором кадре задается значениями $f_2(\boldsymbol{x})$, $\boldsymbol{x} \in M^{-1}(D')$, где D' – область, занимаемая сместившимся объектом на втором кадре. Будем решать задачу нахождения области D' со следующим дополнительным ограничением: $M^{-1}(D') = D + \boldsymbol{h}$, где $\boldsymbol{h}$ – некоторый неизвестный вектор смещения, который требуется определить. Таким образом, мы предполагаем, что от кадра к кадру объект сдвигается относительно неподвижного фона на некоторый вектор, но что поворотом

объекта, изменением его размера и формы можно пренебречь, как показано на рис. 1. При таких предположениях задача удержания объекта сводится к определению вектора смещения $\boldsymbol{h}$.

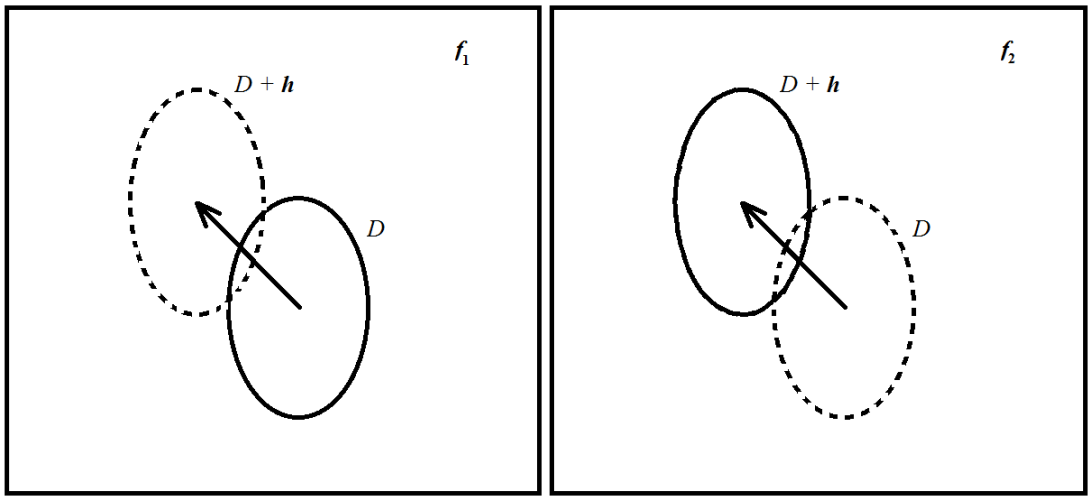

Рис. 1.

Построение целевого функционала

Мы будем искать вектор смещения $\boldsymbol{h}$ как точку минимума некоторого целевого функционала. Целевой функционал определяется следующими условиями. Во-первых, для точек подвижного объекта соответствующие значения уровня яркости на первом и на втором кадре должны мало отличаться: $f_1(\boldsymbol{x}) \approx f_2(\boldsymbol{x}+\boldsymbol{h})$, для точек $\boldsymbol{x} \in D$. Таким образом,

$$\int_D (f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}+\boldsymbol{h}))^2 d\boldsymbol{x} \to min \qquad (1)$$

Во-вторых, такие точки окрестности Ω, которые не принадлежат подвижному объекту ни на первом, ни на втором кадре, являются точками фона. В таких точках значения уровня яркости на первом и на втором кадре должны быть близки: $f_1(\boldsymbol{x}) \approx f_2(\boldsymbol{x})$ для точек $\boldsymbol{x} \in \Omega - D - (D+\boldsymbol{h})$. Отсюда

$$\int_{\Omega - D - (D+\boldsymbol{h})} (f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}))^2 d\boldsymbol{x} \to min \qquad (2)$$

Комбинируя условия (1) и (2), получаем следующий целевой функционал $F(\boldsymbol{h})$:

$$F = \int_D (f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}+\boldsymbol{h}))^2 d\boldsymbol{x} + \int_{\Omega - D - (D+\boldsymbol{h})} (f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}))^2 d\boldsymbol{x} \to min \quad (3)$$

Оказывается удобным вычислять все интегралы по окрестности Ω. Это позволяет использовать стандартный алгоритм быстрого преобразования Фурье (БПФ). Чтобы переписать функционал F в соответствующем виде, используем функцию-индикатор области D:

$$I_D(\boldsymbol{x}) = \begin{cases} 0, \boldsymbol{x} \notin D \\ 1, \boldsymbol{x} \in D \end{cases} \quad (4)$$

Тогда индикатор области $\Omega - D - (D + \boldsymbol{h})$ есть

$$I_{\Omega - D - (D+\boldsymbol{h})}(\boldsymbol{x}) = 1 - I_D(\boldsymbol{x}) - I_{D+\boldsymbol{h}}(\boldsymbol{x}) + I_D(\boldsymbol{x}) I_{D+\boldsymbol{h}}(\boldsymbol{x}) \quad (5)$$

С использованием (3), (4) и (5) мы получаем следующее выражение для целевого функционала:

$$\begin{aligned} F = \int_\Omega I_D(\boldsymbol{x})(f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}+\boldsymbol{h}))^2 d\boldsymbol{x} + \\ + \int_\Omega (1 - I_D(\boldsymbol{x}) - I_{D+\boldsymbol{h}}(\boldsymbol{x}) + I_D(\boldsymbol{x}) I_{D+\boldsymbol{h}}(\boldsymbol{x}))(f_1(\boldsymbol{x}) - f_2(\boldsymbol{x}))^2 d\boldsymbol{x} \end{aligned} \quad (6)$$

Следующее тождество понадобится для дальнейшего преобразования (6):

$$\int_\Omega I_{D+\boldsymbol{h}}(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} = \int_\Omega I_D(\boldsymbol{x}) f(\boldsymbol{x}+\boldsymbol{h}) d\boldsymbol{x} \quad (7)$$

Раскрывая скобки в (6) и обозначая сумму членов, не зависящих от $\boldsymbol{h}$, через C, получаем, с использованием (7):

$$\begin{aligned} F = \int_\Omega -2\, I_D(\boldsymbol{x}) f_1(\boldsymbol{x}) f_2(\boldsymbol{x}+\boldsymbol{h}) + I_D(\boldsymbol{x})(f_2(\boldsymbol{x}+\boldsymbol{h}))^2 d\boldsymbol{x} + \\ + \int_\Omega I_D(\boldsymbol{x})(-1 + I_D(\boldsymbol{x}+\boldsymbol{h}))(f_1(\boldsymbol{x}+\boldsymbol{h}) - f_2(\boldsymbol{x}+\boldsymbol{h}))^2 d\boldsymbol{x} + C \end{aligned} \quad (8)$$

Использование БПФ

Все интегралы в (8) имеют вид

$$w(\boldsymbol{h}) = \int_{\Omega} u(\boldsymbol{x})v(\boldsymbol{x}+\boldsymbol{h})d\boldsymbol{x} \tag{9}$$

Такой интеграл может быть вычислен с использованием преобразования Фурье следующим образом. Имеем

$$w(-\boldsymbol{h}) = \int_{\Omega} u(\boldsymbol{x})v(\boldsymbol{x}-\boldsymbol{h})d\boldsymbol{x} = \int_{\Omega} u(\boldsymbol{x})v_1(\boldsymbol{h}-\boldsymbol{x})d\boldsymbol{x} \tag{10}$$

Из (10) получаем, что $w(-\boldsymbol{h})$ является сверткой функций $u(\boldsymbol{x})$ и $v_1(\boldsymbol{x})$, где $v_1(\boldsymbol{x}) = v(-\boldsymbol{x})$. Обозначим через $\hat{u}(\boldsymbol{\omega}) = \mathcal{F}[u(\boldsymbol{x})](\boldsymbol{\omega})$ преобразование Фурье функции $u(\boldsymbol{x})$:

$$\hat{u}(\boldsymbol{\omega}) = (2\pi)^{-1} \int_{\Omega} \exp(-i\boldsymbol{\omega}\boldsymbol{x})u(\boldsymbol{x})d\boldsymbol{x} \tag{11}$$

Соответственно, $u(\boldsymbol{x}) = \mathcal{F}^{-1}[\hat{u}(\boldsymbol{\omega})](\boldsymbol{x})$ – обратное преобразование Фурье. Аналогично, пусть $\hat{v}(\boldsymbol{\omega})$ и $\hat{w}(\boldsymbol{\omega})$ – преобразования Фурье функций $v(\boldsymbol{x})$ и $w(\boldsymbol{x})$.

Тогда $\hat{u}(\boldsymbol{\omega})^* = \mathcal{F}[u(-\boldsymbol{x})](\boldsymbol{\omega})$ – преобразование Фурье функции $u(-\boldsymbol{x})$, где звездочкой обозначено комплексное сопряжение. Преобразование Фурье свертки есть произведение преобразований Фурье сворачиваемых функций. Таким образом, из (10) получаем:

$$\hat{w}(\boldsymbol{\omega}) = \hat{u}(\boldsymbol{\omega})^* \cdot \hat{v}(\boldsymbol{\omega}) \tag{12}$$

Из (12) получаем следующую формулу для вычисления интеграла (9) с использованием преобразования Фурье:

$$w(\boldsymbol{h}) = \mathcal{F}^{-1}[\hat{u}(\boldsymbol{\omega})^* \cdot \hat{v}(\boldsymbol{\omega})] \tag{13}$$

Теперь, преобразуя (8) по образцу (9) и (13), и отбрасывая постоянную C, получаем:

$$F(\boldsymbol{h}) = \mathcal{F}^{-1}[\,-2\,\widehat{I_D f_1}^{\,*} \cdot \hat{f}_2 \; + \widehat{I_D}^{\,*} \cdot \mathcal{F}[f_2^2 + (-1+I_D)(f_1-f_2)^2]] \tag{14}$$

Далее мы предполагаем, что окрестность Ω – квадратный массив с длиной строки и длиной столбца, равной N, причем N является степенью двойки, $N = 2^m$. Дискретное преобразование Фурье двумерного массива $f(\boldsymbol{k})$ размера $N \times N$ находится по следующей формуле (11а).

$$\hat{f}(\boldsymbol{s}) = N^{-1} \sum \exp\left(-\frac{2\pi i \boldsymbol{k}\boldsymbol{s}}{N}\right) f(\boldsymbol{k}). \tag{11a}$$

Здесь $\boldsymbol{k}$ и $\boldsymbol{s}$ – целочисленные точки из квадрата $[0,N-1]\times[0,N-1]$. Суммирование ведется по всем целочисленным точкам этого квадрата. Мы будем использовать (11a) как приближение для (11) при $\boldsymbol{\omega}=2\pi\boldsymbol{s}/N$ (с точностью до постоянного множителя).

Алгоритм 1 удержания объекта

Таким образом, из (14) мы получаем следующий алгоритм для определения вектора смещения $\boldsymbol{h}$.

А) На вход подаются массивы f_1, f_2 и I_D размерности $N\times N$.

Б) С помощью поэлементного сложения, вычитания и умножения данных массивов вычисляются массивы $u=-2I_Df_1$ и $v=f_2^2+(-1+I_D)(f_1-f_2)^2$.

В) Вычисляются БПФ массивов: $\hat{u},\hat{v},\widehat{f_2},\widehat{I_D}$.

Г) С помощью поэлементного комплексного умножения вычисляются массивы $\hat{u}^*\cdot\widehat{f_2}$ и $\widehat{I_D}^*\cdot\hat{v}$.

Д) Вычисляется поэлементная сумма полученных в п. Г) массивов: $\hat{u}^*\cdot\widehat{f_2}+\widehat{I_D}^*\cdot\hat{v}$.

Е) Вычисляется обратное БПФ полученного в п. Д) массива. Это целевой функционал F.

Ж) Находится точка минимума массива F. Это искомый вектор смещения $\boldsymbol{h}$.

При дискретном преобразовании Фурье произведению Фурье-образов соответствует циклическая свертка массивов. Поэтому векторам смещения $\boldsymbol{h}$ с отрицательными координатами в интервале $[-N/2,-1]$ будут соответствовать точки минимума массива $F(\boldsymbol{h})$ с координатами в интервале $[N/2,N-1]$.

Оценим число операций алгоритма А) – Ж). Пункт Б) требует 5 поэлементных умножений и 3 поэлементных сложения/вычитания массивов $N\times N$. Пункт В) требует выполнения 4 БПФ массивов $N\times N$. Пункт Г) требует выполнения двух поэлементных комплексных умножений массивов $N\times N$, или 4 вещественных умножения и 2 вещественных сложения. Пункт Д) требует выполнения одного комплексного или 2 вещественных сложения массивов $N\times N$. В пункте Е) одно БПФ массива, в пункте Ж) нахождение точки минимума. Итого 7 операций поэлементного сложения/вычитания, 9 операций поэлементного умножения, 5 БПФ и одно нахождение точки минимума массивов $N\times N$.

Накопление погрешности

При многократном последовательном применении алгоритма 1 для серии кадров видеопоследовательности может накапливаться вычислительная погрешность. Если объект наблюдения смещается относительно малоконтрастного фона, область D может сместиться на край объекта, некоторое время двигаться совместно с ним, а затем объект будет полностью потерян, как показано на рис. 2 – 5. Марка в виде крестика находится в центре тяжести области D.

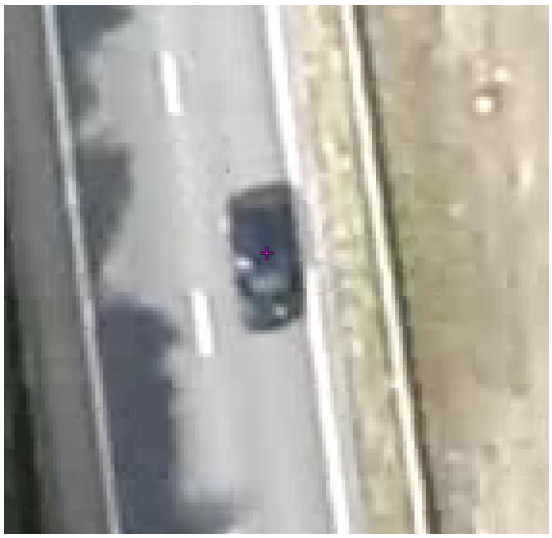

Рис. 2. Кадр 1

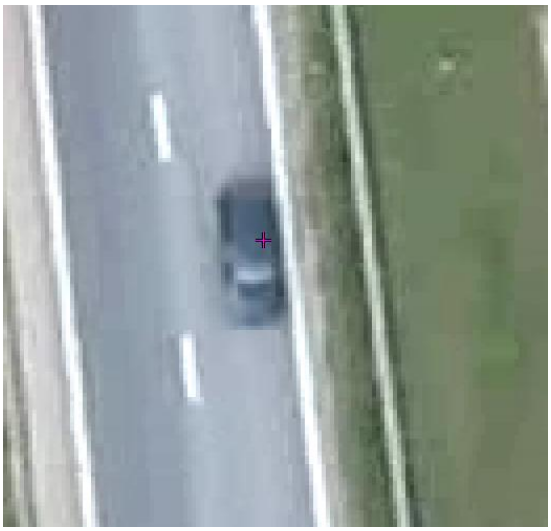

Рис. 3. Кадр 180

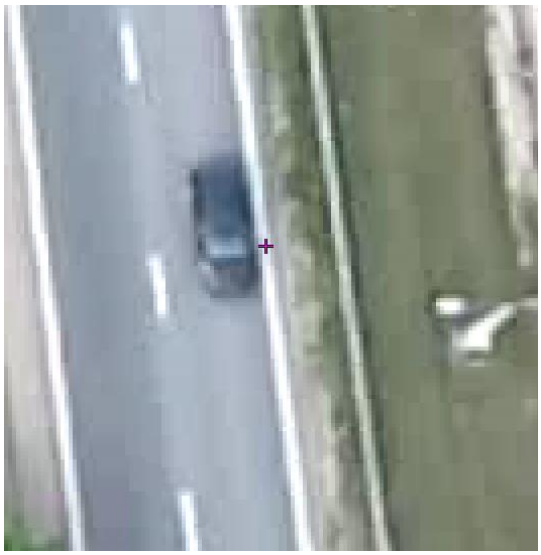

Рис. 4. Кадр 200

Рис. 5. Кадр 220

Для компенсации накопления вычислительной погрешности алгоритма 1 мы используем вспомогательный алгоритм привязки к пятну.

Привязка к пятну

Мы исходим из того, что объект наблюдения представляет собой светлое либо темное пятно на неподвижном фоне. Форма пятна определяется функцией-индикатором I_D. Фактическое положение пятна определяется из условия максимума либо минимума (для светлого и для темного пятна) коэффициента корреляции между функцией уровня яркости изображения и функцией-индикатором области D, смещенной на неизвестный вектор, определяющий искомое положение пятна. Таким образом, фактическое положение пятна определяется из условия максимума либо минимума следующего функционала:

$$G(\boldsymbol{h}) = \int_{\Omega} (I_D(\boldsymbol{x}+\boldsymbol{h}) - \overline{I_D})(f_1(\boldsymbol{x}) - \overline{f_1})d\boldsymbol{x} \to max \ \ (min) \qquad (15)$$

Здесь $\overline{I_D}$ и $\overline{f_1}$ – средние значения соответствующих функций. Оказывается удобным не делать разницы между темным и светлым пятном, и всегда искать максимум $G^2(\boldsymbol{h})$. Также практически нужно искать не абсолютный максимум $G^2(\boldsymbol{h})$ по всей области Ω, а ближайший к нулю локальный максимум этой функции. Это так поскольку алгоритм привязки к пятну используется только для компенсации вычислительной погрешности алгоритма 1, которая обычно незначительна при переходе на один кадр.

Выражение (15) также может быть вычислено с использованием БПФ. Преобразуем (15) к виду

$$G(\boldsymbol{h}) = \int_{\Omega} I_D(\boldsymbol{x}+\boldsymbol{h}) f_1(\boldsymbol{x})d\boldsymbol{x} - \overline{I_D}\,\overline{f_1}\,|\Omega| \qquad (16)$$

Здесь $|\Omega|$ – площадь области Ω (число элементов массива в дискретном случае). Мы получаем из (16), с использованием (9) и (13):

$$G(\boldsymbol{h}) = \mathcal{F}^{-1}\left[\widehat{I_D}(\boldsymbol{\omega}) \cdot \widehat{f_1}(\boldsymbol{\omega})^*\right] - \overline{I_D}\,\overline{f_1}\,|\Omega| \qquad (17)$$

Из (17) получается следующий алгоритм 2 корректировки положения объекта слежения с помощью привязки к пятну.

А) На вход подаются массивы f_1 и I_D размерности $N \times N$.

Б) Вычисляются средние значения $\overline{f_1}$ и $\overline{I_D}$

В) Вычисляются БПФ массивов $\widehat{f_1}$ и $\widehat{I_D}$

Г) С помощью поэлементного комплексного умножения вычисляется массив $\widehat{I_D} \cdot \widehat{f_1}^*$

Д) Вычисляется обратное БПФ массива $\widehat{I_D} \cdot \widehat{f_1}^*$

Е) Из полученного в п. Д) массива поэлементно вычитается произведение $\overline{f_1} \cdot \overline{I_D} \cdot |\Omega|$. Получаем функционал $G(\boldsymbol{h})$. Далее $G(\boldsymbol{h})$ поэлементно возводится в квадрат, получаем массив $G^2(\boldsymbol{h})$

Ж) Ищется ближайшая к нулю (см. замечание ниже) точка локального максимума массива $G^2(\boldsymbol{h})$. Это искомый вектор смещения $\boldsymbol{h}$.

Как и в алгоритме 1, векторам смещения $\boldsymbol{h}$ с отрицательными координатами в интервале $[-N/2, -1]$ будут соответствовать точки локального максимума массива $G^2(\boldsymbol{h})$ с координатами в интервале $[N/2, N-1]$. В пункте Ж) алгоритма привязки к пятну ищется такой локальный максимум $G^2(\boldsymbol{h})$, которому соответствует вектор $\boldsymbol{h}$ наименьшей длины.

Оценим число операций алгоритма привязки к пятну. Пункт Б) требует два вычисления средних значений массивов $N \times N$. Пункт В) требует выполнения 2 БПФ массивов $N \times N$. Пункт Г) требует выполнения одного поэлементного комплексного умножения массивов $N \times N$, или 2 вещественных умножения и одно вещественное сложение. В пункте Д) одно БПФ массива. Пункт Е) требует выполнения одного поэлементного вычитания и одного поэлементного умножения. В пункте Ж) нахождение точки локального максимума. Итого 2 операции поэлементного сложения/вычитания, 3 операции поэлементного умножения, 3 БПФ, два вычисления среднего и одно нахождение точки максимума массивов $N \times N$.

Мы видим, что алгоритм привязки к пятну требует существенно меньше арифметических операций, чем основной алгоритм 1 удержания объекта. К тому же можно проводить привязку к пятну не на каждый кадр, а через несколько кадров. Практически в тестовой программе привязка к пятну производится через один кадр. В этом случае почти все вычислительные затраты относятся к алгоритму 1.

Рис. 6. Кадр 180, есть привязка к пятну

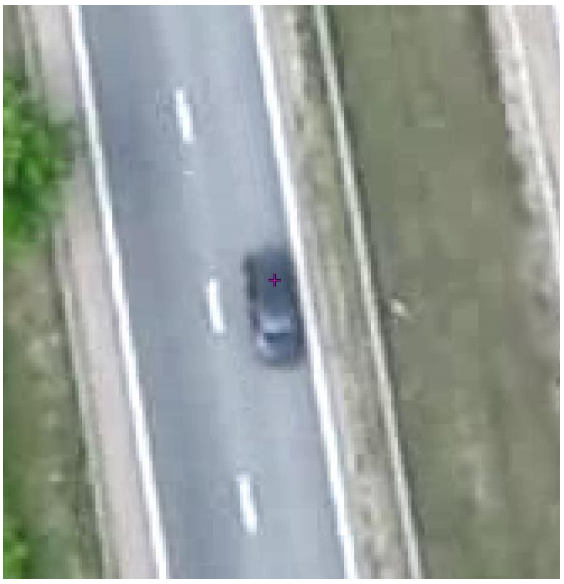

Рис. 7. Кадр 220, есть привязка к пятну

Применение алгоритма привязки к пятну совместно с алгоритмом 1 позволяет избежать накопления вычислительной погрешности и потери объекта слежения, что показано на рис. 6 и 7 выше.

Выбор индикатора области

Объект слежения может быть задан на первом кадре видеопоследовательности либо программно, например, при работе одного из алгоритмов по выделению подвижных объектов, либо оператором.

В первом случае может быть задано либо пятно, соответствующее данному объекту, либо задана точка на объекте. Во втором случае оператор, как правило, задает только одну точку на объекте. Если пятно, соответствующее объекту наблюдения, известно, то индикатор области задается непосредственно. При задании объекта одной точкой часто (если объект простой формы) бывает достаточно задать область в виде эллипса с осями, параллельными сторонам кадра и с центром в данной точке. Размеры эллипса выбираются соответствующими характерным размерам объектов данного вида. Примерами такой ситуации могут быть слежение за фигурами людей в поле или слежение за движущимися автомобилями на дороге, как на рис. 2–7.

На рис. 8 и 9 ниже приведены примеры массива уровней яркости f_1 с объектом слежения в центре и функции-индикатора области D в виде эллипса. При этом для конкретного объекта в этом примере размеры объекта и используемой модельной области в виде эллипса достаточно сильно различаются. Тем не менее, алгоритмы 1 и 2 совместно обеспечивают устойчивое удержание объекта.

Рис. 8. Массив f_1 128x128 пикселей

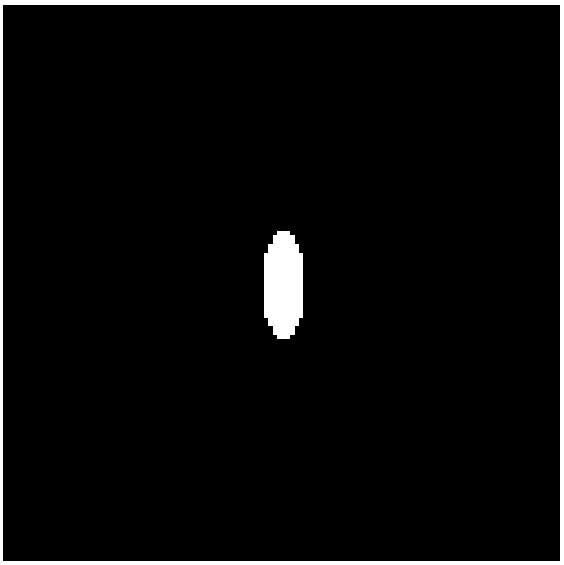

Рис. 9. Массив I_D 128x128 пикселей, эллипс с осями 9x25 пикселей

В ситуации, когда размеры выбранной области D не точно совпадают с фактическими размерами удерживаемого объекта, хорошие результаты получаются, если использовать в качестве функции-индикатора не ступенчатую функцию с двумя значениями 0 и 1, а непрерывную, например, гауссовскую функцию, как показано на рис. 10.

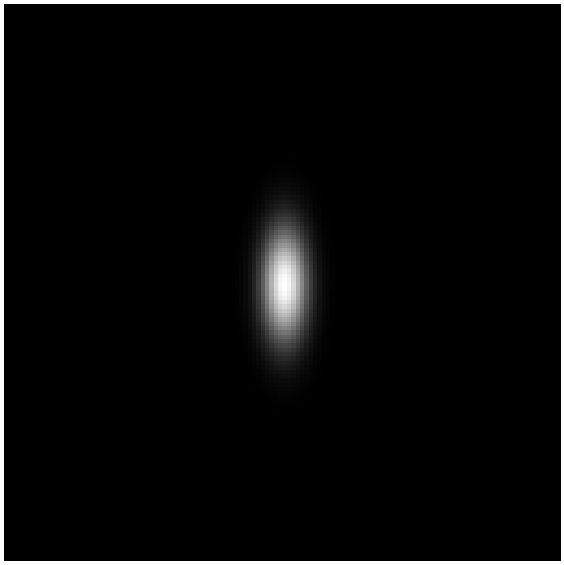

Рис. 10. Массив I_D 128x128 пикселей, гауссовская функция $(a = 9, b = 25)$

Пусть эллипс задается неравенством $r(x,y) < 1$, где

$$r(x,y) = \frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2}$$

Тогда соответствующую гауссовскую функцию можно выбрать в виде $I_D(x,y) = \exp(-r(x,y))$.

Можно использовать гауссовский индикатор области и в алгоритме 1. При этом формула (5) для индикатора области $\Omega - D - (D + \boldsymbol{h})$ перестает быть корректной. Практически хорошие результаты получаются, если заменить (5) на следующую формулу

$$I_{\Omega - D - (D+\boldsymbol{h})}(\boldsymbol{x}) = 1 - I_D(\boldsymbol{x}) - I_{D+\boldsymbol{h}}(\boldsymbol{x}) \qquad (18)$$

Формула (18) соответствует (5) при непересекающихся областях D и $D + \boldsymbol{h}$. Тогда формула (14) примет следующий вид

$$F(\boldsymbol{h}) = \mathcal{F}^{-1}[\,-2\,\widehat{I_D f_1}^{\,*} \cdot \widehat{f_2} \; + \widehat{I_D}^{\,*} \cdot \mathcal{F}[f_2^2 - (f_1 - f_2)^2]] \qquad (19)$$

В самом алгоритме 1 единственным отличием будет формула для вычисления функции v: $v = f_2^2 - (f_1 - f_2)^2$ для гауссовского индикатора вместо $v = f_2^2 + (-1 + I_D)(f_1 - f_2)^2$ для ступенчатого индикатора. Далее мы приводим временные характеристики описанного в данном разделе алгоритма по результатам тестирования.

Время работы алгоритма

Время совместной работы алгоритмов 1 и 2 тестировалось на малогабаритном вычислительном модуле IntencePC, при исполнении на одном ядре процессора. Процессор Intel Core i5 – 4300U CPU 1.90 GHz x64 RAM 16 Gb. Также тестирование было проведено на настольном ПК Pentium Dual-Core CPU 3,06 GHz при исполнении на одном ядре процессора. При выборе размера окрестности Ω 128x128 пикселей ($N = 128$) и при проведении привязки к пятну через один кадр среднее время работы составило 10,4 миллисекунды на кадр (ПК Pentium) без учета времени загрузки кадров в память и без учета времени расчета матрицы M связующего проективного преобразования. При этом фактически выбиралась область кадра 512x512 пикселей и уменьшалась (прореживалась) до размера 128x128 пикселей. Выбиралась ступенчатая функция-индикатор области объекта в виде эллипса, как показано на рис. 9. Оказалось, что в тех же условиях, но при выборе гауссовской функции-индикатора, на ПК Pentium время увеличивается почти вдвое до 19,4 миллисекунды на кадр за счет того, что функция-индикатор рассчитывалась каждый раз заново при поступлении нового кадра. Это означает, что как саму функцию-индикатор в виде массива, так и массив ее БПФ следует вычислять однократно перед началом работы алгоритма.

При выборе $N = 64$ (ПК Pentium) среднее время работы алгоритма составляет 7,6 миллисекунды на кадр со ступенчатым индикатором и 9,2 миллисекунды на кадр с гауссовским индикатором. Время сокращается менее чем вдвое по сравнению с $N = 128$. Время работы поэлементных операций над массивами в алгоритмах 1 и 2 должно уменьшаться в 4 раза, а время расчета БПФ даже больше, чем в 4 раза. Таким образом, при $N = 64$ основное время тратится не на сам алгоритм удержания, а на вспомогательные операции, такие как проективное преобразование окрестности.

При выборе $N = 256$ на ПК Pentium среднее время работы алгоритма составляет 77 миллисекунд на кадр со ступенчатым индикатором и 81 миллисекунду на кадр с гауссовским индикатором.

Результаты тестирования приводятся в таблицах 1 (IntencePC) и 2 (ПК Pentium). В таблицах N – размера окрестности Ω по строке, N_1 – фактический размер область кадра по строке с последующим прореживанием до N пикселей, t – среднее время работы алгоритма удержания объекта (миллисекунд на кадр) при выборе ступенчатого индикатора области, t_G – среднее время работы алгоритма удержания объекта (миллисекунд на кадр) при выборе гауссовского индикатора области.

Таблица 1

Время работы на IntencePC

N	64	128	256
N_1	512	512	512
t	2,44	5,10	16,2
t_G	2,83	6,3	22,6

Таблица 2

Время работы на ПК Pentium

N	64	128	256
N_1	512	512	512
t	7,6	10,4	77
t_G	9,2	19,4	81

Результаты показывают превосходство в производительности современного малогабаритного вычислительного модуля IntencePC над

настольным ПК Pentium от двух до почти четырех раз. Правда, нужно учитывать, что при тестировании на IntencePC не выполнялись параллельно с данной никакие другие задачи. Параллельное выполнение нескольких вычислительно интенсивных процессов, таких как работа алгоритмов совмещения изображений, стабилизации кадров и удержания объектов, приводит к перегреву процессора и последующему снижению тактовой частоты и производительности.

Общий вывод: разработанный алгоритм удержания подвижных объектов пригоден для выполнения на малогабаритном вычислительном модуле IntencePC в реальном масштабе времени при выборе размера обрабатываемой окрестности до 128 пикселей включительно.

* * *

Мы рассмотрели алгоритм межкадрового прослеживания подвижного объекта на видеопоследовательности, использующий вычисления на основе быстрого преобразования Фурье.

Библиографический список

1. Cavallaro A, Maggio E. Video tracking: theory and practice. London: Wiley, 2011. 266 p.

2. Wu B., Nevatia R. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors // Proceedings of International Conference on Computer Vision. Washington: IEEE Computer Society, 2005. P. 90–97.

3. Визильтер Ю. В., Лагутенков А. В., Ососков М. В., Выголов О. В., Блохинов Ю. Б. (2006) Выделение и межкадровое прослеживание движущихся объектов при регистрации изображений сложных пространственных сцен произвольно движущимися двумерными сенсорами // Вестн. компьютерных и информац. технологий. 2006. No 3. С. 34–39.