

Ю.В. Грушко, Р.И. Паровик
**БЫСТРЫЙ ТРЕКИНГ ЗРАЧКА, ОСНОВАННЫЙ НА
ИССЛЕДОВАНИИ ГРАНИЧНО-СТУПЕНЧАТОЙ МОДЕЛИ
ИЗОБРАЖЕНИЯ И МНОГОМЕРНОЙ ОПТИМИЗАЦИИ
МЕТОДОМ ХУКА-ДЖИВСА**

Грушко Ю.В., Паровик Р.И. Быстрый трекинг зрачка, основанный на исследовании гранично-ступенчатой модели изображения и многомерной оптимизации методом Хука-Дживса.

Аннотация. Предлагается новый быстрый метод регистрации движения глаз в режиме реального времени на основе исследования гранично-ступенчатой модели полутонового изображения оператором Лапласиан – Гауссиана и нахождения нового предложенного дескриптора накопленных разностей (идентификатора точек), который отображает меру равноудаленности каждой точки от границ некоторой относительно монотонной области (например, зрачок глаза). Работа данного дескриптора исходит из предположения о том, что зрачок в кадре является наиболее округлой монотонной областью, имеющей высокий перепад яркости на границе, пиксели области должны иметь интенсивность меньше заранее заданного порога (но при этом зрачок может не являться самой темной областью на изображении). Учитывая все вышеперечисленные характеристики зрачка, дескриптор позволяет достичь высокой точности детектирования его центра и размера в отличии от методов, основанных на пороговой сегментации изображения, опирающихся на предположение о зрачке как наиболее темной области, морфологических методов (рекурсивная морфологическая эрозия), корреляционных или методов, исследующих только граничную модель изображения (преобразование Хафа и его вариации с двумерным и трехмерным пространствами параметров, алгоритм Starburst, Swirski, RANSAC, ElSe).

Исследована возможность представления задачи трекинга зрачка как задачи многомерной оптимизации и ее решение неградиентным методом Хука –Дживса, где в качестве целевой функции выступает функция, выражающая дескриптор. При этом отпадает необходимость в вычислении дескриптора каждой точки изображения (составления специальной аккумуляторной функции), что значительно ускоряет работу метода.

Проведен анализ предложенных дескриптора и метода, а также разработан программный комплекс на языке Python 3 (визуализация) и C++ (ядро трекинга) в лаборатории физико-математического факультета Камчатского государственного университета им. Витуса Беринга, позволяющий иллюстрировать работу метода и осуществлять трекинг зрачка в режиме реального времени.

Ключевые слова: математические модели, детекторы, дескрипторы, окулография, айтрекинг, амиотрофический склероз, зрачок, Лапласиан-Гауссиана, метод Хука-Дживса, численная оптимизация, преобразование Хафа

1. Введение. Окулография играет ключевую роль в создании устройств типа человек-компьютер для осуществления альтернативной аугментативной коммуникации (ААК) людей, страдающих невропаралитическими синдромами и другими болезнями центральной нервной системы (ЦНС). Доказано, что в случае синдрома «запертого человека»

по международной классификации болезней (МКБ-10) код G83.5 или синдрома бокового амиотрофического склероза (БАС) МКБ-10 код G12 у больных наблюдается полная потеря речи (афазия), паралич, за исключением мышц глаз, при полной сохранности сознания и чувствительности [1]. Общение с такими больными возможна только с помощью движения глаз, используя технику видеоокулографии (ВОГ) и электроокулографии (ЭОГ). По данным National Amyotrophic Lateral Sclerosis Registry, больных с синдромом БАС в мире насчитывается около 450 000 человек. В среднем БАС диагностируют у 3-5 человек на 100 тысяч населения.

Сам процесс ВОГ сводится к трем этапам: 1. Создание контраста инфракрасной подсветкой (ИК) между радужной оболочкой и зрачком. При этом используется метод «темного» или «яркого» зрачка. Так, при «темном» зрачке источник подсветки сдвинут относительно оптической оси камеры окулографа и вторичное отражение от сетчатки не поступает в камеру. 2. Движение глаза фиксируется камерой окулографа или иного устройства. 3. Выполняется препроцессинг кадра (сглаживание изображений, эквализация гистограммы и др.), детектирование зрачка в кадре, нахождение его координат и радиуса. Данный этап является достаточно сложным. Необходим поиск более эффективных алгоритмов трекинга как по вычислительной сложности, так и по точности детектирования при ограниченных вычислительных ресурсах (с использованием смартфонов, очков айтрекеров и др.).

В настоящее время разработаны различные методы и алгоритмы для поиска зрачка в кадре. Методы [2, 3, 4], использующие в своей основе операцию свертки с примитивами Хаара заранее заданных размеров, имеют большую вычислительную сложность (сама операция свертки достаточно ресурсоемкая операция), так как для корректного поиска зрачка в трехмерном пространстве параметров (x, y, R) необходимо иметь набор примитивов различного радиуса (необходимо учитывать миоз, т.е. сжатие зрачка при реагировании на свет, и мидриаз – его расширение), операция свертки повторяется итеративно для каждого примитива, после чего выбирается точка (x, y) , имеющая наибольший отклик. В найденной точке находится центр зрачка с соответствующим радиусом (в некоторых алгоритмах, например, Swirski [3], это лишь начальный этап поиска фактического центра зрачка с последующим его уточнением).

Большое количество методов основано на пороговой сегментации изображения и операциях математической морфологии [5, 6, 7, 8], опирающихся на предположение о зрачке как наиболее темной области.

Выбираются все точки изображения, интенсивности которых ниже некоторого порога т.е. $f(x, y) \leq T$. Проведя сегментацию изображения и его обработку с помощью морфологических операторов эрозии, можно вычислить центр масс, который и определяет его центр. Другой подход предусматривает рекурсивное применение морфологической эрозии к точкам, составляющим область зрачка, тогда оставшийся последний пиксель (самый удаленный от точек фона) и будет являться центром зрачка. Данные методы показывают точные результаты на изображениях с темным крупным зрачком, но неработоспособны на иных типах изображений, в которых темные ресницы, веки и тени на кожном эпителии могут сливаться со зрачком. В случае рекурсивной эрозии [8, 109 с.] необходимо, чтобы зрачок в кадре был самой большой темной областью на изображении, что существенно ограничивает применение данного метода в условиях недостаточной освещенности глаза, при съемке глаза с расстояния или во время миоза зрачка. Поскольку операция бинаризации, вычисление порога и применение морфологических операторов – вычислительно простые процедуры, то преимуществом методов является скорость их исполнения.

Еще один обширный класс – методы, исследующие граничную модель изображения, такие как метод оценки параметров модели на основе случайных выборок RANSAC [9], алгоритм Starburst [10], ElSe [11], Swirski [3], [12], метод обнаружения краевых сегментов [13]. Одной из часто используемых техник является преобразование Хафа (с двумерным [14] и трехмерным пространством параметров [15, 16]). Преобразование Хафа – вычислительный алгоритм, применяемый для параметрической идентификации множества кривых заданного семейства (прямая, эллипс) на растровом изображении. Так, для поиска зрачка в первую очередь выделяются точки-кандидаты, лежащие на границе между зрачком и радужной оболочкой. Такие точки будут иметь высокий градиент яркости $mag(\nabla f(x, y))$. Расположение граничных точек зрачка можно задать уравнением $(x - x_0)^2 + (y - y_0)^2 = R^2$. Очевидно, что множество центров всех возможных окружностей (центры, лежащие на граничных точках) радиуса R , проходящих через некоторую голосующую точку аккумуляторного пространства A , образует окружность радиуса R вокруг этой точки. Функция отклика в преобразовании Хафа для поиска окружностей заданного радиуса представляет собой окружность такого же радиуса с центром в голосующей точке. Точка аккумуляторной функции, получившая наибольшее количество

голосов, то есть $\arg \max_{x, y} A(x, y)$, отвечает наилучшему найденному по-

ложению фигуры заданного класса на изображении. Преобразование Хафа устойчиво к удалению части искомой фигуры. Например, центр зрачка можно найти, если его часть прикрыта веком или имеются блики на границе зрачка. Недостаток данных методов – их большая вычислительная сложность.

В данной работе делается попытка разработать дескриптор (специальный идентификатор точек, выделяющий искомый объект) и метод на основе этого дескриптора, позволяющий с высокой точностью и скоростью определять координаты центра и размеры зрачка на изображении, учитывая все характеристики зрачка: круглая/эллиптическая форма области, ее монотонность, высокий перепад яркости на границе области и невысокая интенсивность точек, составляющих данную область. Учитывая одновременно все вышеперечисленные характеристики зрачка, дескриптор позволит достичь высокой точности детектирования.

Сама идея дескриптора позволяет определить еще одну важную задачу исследования – представление задачи поиска и трекинга зрачка, как задачи многомерной безусловной оптимизации целевой функции, выражающей дескриптор. При этом отпадает необходимость в вычислении дескриптора для каждой точки изображения, что позволит значительно ускорить работу предложенного метода и использовать его на низкопроизводительных вычислительных устройствах.

2. Описание дескриптора накопленных разностей – Accumulated Difference Descriptor. Работа дескриптора накопленных разностей, предложенного в этой работе, исходит из предположения о том, что зрачок в кадре является наиболее округлой монотонной областью, имеющей высокий перепад яркости на границе, пиксели которой удовлетворяют условию $f(x, y) \leq T$, то есть интенсивность пикселей меньше заранее заданного порога (при этом зрачок может не являться самой темной областью на изображении). Сам дескриптор отражает меру равноудаленности каждой точки от границ некоторой области изображения, обладающей вышеперечисленными свойствами. Причем если таких областей на изображении несколько, дескриптор будет иметь минимальный отклик в центре наиболее округлой области. Предложенный дескриптор производит поиск в трехмерном пространстве параметров (x, y, R) – координаты зрачка и радиус/множество граничных точек зрачка. Общая схема вычисления дескриптора представлена на рисунке 1 (используется нотация блок-схем и UML-диаграмм).

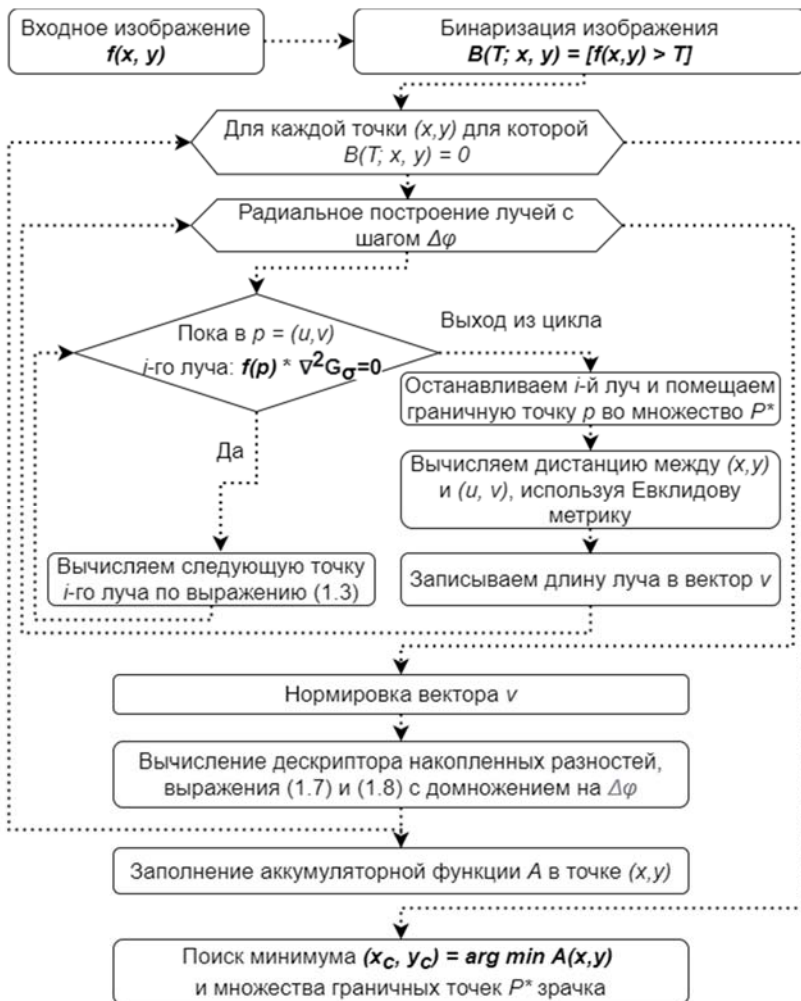


Рис. 1. Блок-схема вычисления дескриптора накопленных разностей

Опишем основные шаги вычисления дескриптора.

Шаг 1. Сегментация изображения по порогу. Зрачок здорового человека обладает формой близкой к кругу/эллипсу, и отражает мало света в том случае, если направление источника инфракрасной подсветки глаза не совпадает с направлением камеры. Это свойство позволяет радикально уменьшить количество обрабатываемой информации, а соответственно и ускорить работу метода.

Очевидный способ выделить темный зрачок из окружающего фона изображения $f(x, y)$ состоит в выборе значения порога T , разграничивающего моды распределения яркостей. Тогда любая точка (x, y) , в которой $f(x, y) \leq T$, называется точкой зрачка (в противном случае – точкой фона). Найдем сегментированное изображение $\mathcal{B}(T; x, y)$:

$$\mathcal{B}(T; x, y) = [f(x, y) > T] = \begin{cases} N, & \text{if } f(x, y) > T \\ 0, & \text{otherwise} \end{cases},$$

где N – число градаций полутонового изображения. Здесь и далее $[\bullet]$ обозначена нотация Айверсона – индикаторная функция, которая определена на множестве логических выражений $[\bullet]: \mathbb{B} \rightarrow \{0, 1\}$, возвращающая 1 на истинных и 0 на ложных. В результате сегментации образуются две группы пикселей: $Ob \leq T < Bg$. Вычислим значения m_1 и m_2 средних яркостей пикселей по областям Ob и Bg соответственно.

Тогда новое значение порога примет вид:

$$T = \frac{1}{2}(m_1 + m_2), \forall T \in \mathbb{N}_0, T \leq N.$$

Будем выполнять сегментацию по порогу, делать перерасчет m_1 и m_2 по областям и обновлять значение порога T , пока разница значений T между соседними итерациями не окажется меньше значения заданного параметра ΔT . Тем самым мы максимизируем межклассовую дисперсию между Ob и Bg . Стоит отметить, что для точной работы метода идеальной сегментации зрачка не требуется (объясняется далее). Основная задача данного этапа – уменьшение количества информации для ускорения метода.

Пороговая сегментация изображения f представлена на рисунке 26. Последующее уменьшение T истончает область зрачка, в этом случае использование метода, основанного на поиске центра масс самых темных точек или метода проекций яркостей не позволяет корректно найти центр зрачка, на рисунке представлен белым крестиком.

Шаг 2. Исследование гранично-ступенчатой модели полутонного изображения. Для каждой точки (x_0, y_0) изображения $f(x, y)$ для которой выполняется условие $B(T; x_0, y_0) = 0$ строятся N лучей, причем угол φ_i между осью абсцисс OX и лучом i равняется $\varphi_i = \varphi_{i-1} + \Delta\varphi$, здесь $\varphi \in [0, 2\pi]$, $\Delta\varphi$ – угол между лучами, некоторое приращение, регулирующее частоту распределения лучей по кругу и равный $\Delta\varphi = 2\pi / N$. Для построения лучей можно использовать алгоритмы построения отрезка на растре (Брезенхэма [17], DDA-линии [18] и др.). В реализованном алгоритме лучи строились итеративно. По известному множеству x последовательно рассчитывались y для следующих точек. Использовалось уравнение прямой с угловым коэффициентом:

$$y = \frac{y^* - y_0}{x^* - x_0}(x - x_0) + y_0 = k(x - x_0) + y_0,$$

где (x_0, y_0) , (x^*, y^*) – начало и конец распространения луча, $k = \tan \varphi$.

Двигаясь вдоль направления i -го луча, для каждой точки (x, y) будем исследовать гранично-ступенчатую модель полутонного изображения f . В этой модели предполагается, что границы определяются изменениями локальных производных. Так, наибольший перепад яркости в кадре при инфракрасной подсветке глаза присутствует на границе зрачка и радужной оболочки (относительно монотонные области).

Для вычисления значения границы изображения f в точке (x_0, y_0) воспользуемся оператором $\nabla^2 G_\sigma$, где ∇^2 – оператор

Лапласа $(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$, а $G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ – двумерная Гауссова

функция, которая описывает нормальное распределение значений интенсивностей пикселей в окрестности точки (x, y) . Среднеквадратичное отклонение σ отвечает за степень размытия [19].

Выбор оператора Лапласиан–Гауссиана (LoG) $\nabla^2 G_\sigma$ для данной задачи основывается на трех фундаментальных идеях: 1) LoG – дифференциальный оператор, вычисляющий значение второй произ-

водной в каждой точке изображения:2) Гауссова функция в этом операторе обеспечивает сглаживание изображения, снижая интенсивность структур (в т.ч. шумовых) с размерами меньше σ , делая некоторые области более однородными:3) ∇^2 – изотропен то есть инвариантен к повороту, что не только соответствует характеристикам зрительной системы человека [20], но и обеспечивает одинаковый отклик на изменение яркости в любом направлении, в отличие от градиентных операторов, требующих нескольких масок для вычисления максимального отклика в любой точке изображения (Sobel, Prewitt [21] и др.). Это свойство также позволяет подобрать наиболее корректный порог для остановки движения вдоль луча i .

Если не учитывать $\frac{1}{2\pi\sigma^2}$, то $\nabla^2 G_\sigma$ определяется как:

$$\begin{aligned} \nabla^2 G_\sigma(x, y) &= \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} = \frac{\partial}{\partial x} \left(-\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right) + \\ &+ \frac{\partial}{\partial y} \left(-\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}} + \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}. \end{aligned} \quad (1)$$

С учётом $\frac{1}{2\pi\sigma^2}$ (1) преобразуется к виду:

$$\nabla^2 G_\sigma(x, y) = \frac{1}{2\pi\sigma^4} \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Значение границы в точке формируется пространственной сверткой $\nabla^2 G_\sigma$ с изображением f в точке:

$$g(x, y) = \nabla^2 G_\sigma(x, y) * f(x, y).$$

Для наглядности была составлена карта границ для изображения и представлена на рисунке 2а.

В случае если $\nabla^2 G_\sigma(x, y) * f(x, y) > 0$, необходимо остановить движение вдоль луча i и запомнить координаты точки $q^i = (u, v)$, лежащей на границе, например, поместив в массив. Если луч достигает границы изображения – точка не найдена. Для каждой граничной точки необходимо найти расстояние $\xi((x_0, y_0), q^i)$ – длина луча, вычисленная по Евклидовой метрике.

Шаг 3. Вычисление дескриптора накопленных разностей и построение аккумуляторной функции. Непрерывную модель нахождения накопленных разностей в точке $\tau = (x_0, y_0)$ можно описать следующим образом:

$$A(\tau) = \int_{\varphi=0}^{2\pi} \left| \xi(\tau, P(\tau, \varphi)) - \xi(\tau, P(\tau, \varphi + d\varphi)) \right| d\varphi, \quad (2)$$

где P – некоторая функция, которая принимает на вход точку начала распространения луча τ , угол распространения φ относительно оси абсцисс, P возвращает точку $q = (u, v)$, лежащую на границе то есть $\nabla^2 G_\sigma(q) * f(q) > 0$, $d\varphi$ – бесконечно малое приращение угла между лучами.

Зададим аккумуляторную функцию A с размерностью изображения f . Для каждой $\tau = (x_0, y_0)$ зафиксируем сумму модулей разностей длин лучей, ограниченных контуром объекта на изображении (дискретное представление накопленных разностей):

$$\begin{aligned} A(\tau) &= \lim_{\Delta\varphi \rightarrow 0} \sum_{\varphi=0}^{2\pi - \Delta\varphi} \left| \xi(\tau, P(\tau, \varphi)) - \xi(\tau, P(\tau, \varphi + \Delta\varphi)) \right| \Delta\varphi = \\ &= \sum_{i=1}^{N-1} \left| \xi(\tau, q^i) - \xi(\tau, q^{i+1}) \right| \Delta\varphi, \end{aligned} \quad (3)$$

где $q^i = (u, v)$ – точка, находящаяся на границе зрачка, принадлежащая i -му лучу, q^{i+1} – точка, находящаяся на границе зрачка, принадлежащая $i+1$ лучу, $\xi(\tau, q^i) = \left[(x_0 - u)^2 + (y_0 - v)^2 \right]^{\frac{1}{2}}$ – Евклидово расстояние от точки-кандидата до граничной точки – длина луча.

Перед вычислением дескриптора накопленных разностей необходимо провести нормирование вектора, содержащего длины лучей сетки. Делается это для исключения прямого влияния длин лучей на расчет дескриптора, учитываться должны только отношения (разности) между самими лучами. В случае если на изображении существуют неровные темные области (блики, пигментированные образования кожного покрова), имеющие высокий перепад яркости на границе, и по размеру такие области гораздо меньше, чем область зрачка, без проведения нормализации центры таких областей примут наименьшие накопленные разности. Проведя нормализацию построенных лучей сетки из точки (x_0, y_0) , будем учитывать только отношения между смежными лучами сетки, и наименьшие накопленные разности будут находиться именно в центре области зрачка, как наиболее круглого объекта на изображении.

Так как трекинг ведется на полутоновом изображении, $A(x, y)$ будет представлять из себя 8-битную целочисленную карту, и область ее значений составит $A(x, y) \in [0, 2^8 - 1]$, в случае нормирования $A(x, y) \in [0, \Delta\varphi]$.

Чаще всего сегментация по порогу дает неудовлетворительный результат, это связано с индивидуальными особенностями человека, который проходит процесс ВОГ. Так, в случае гетерохромии глаза МКБ-10 H21.24 (аномальное затемнение радужной оболочки), сгущения ресниц или недостаточной инфракрасной подсветке глаза, в $\mathcal{B}(T; x, y)$ могут появиться дополнительные области, которые могут помешать корректному нахождению зрачка в кадре (рисунок 2б). Предполагается, что для *округлой монотонной области*, пиксели которой *удовлетворяют условию* $\mathcal{B}(T; x, y) = 0$ и имеющей *высокий перепад яркости* по всей границе (именно этим трем критериям чаще всего соответствует область зрачка на изображении), сумма накопленных разностей в центре такой области устремится к 0, в отличие от объектов, имеющих неправильную геометрическую форму – ресницы, тени на кожном эпителии или радужке, которые также, как и зрачок, являются достаточно темными областями но не имеют резких перепадов по профилю яркости. Данная особенность часто затрудняет поиск точных координат зрачка, используя сегментационные или морфологические методы, описанные в работах [5, 6, 7], [8, 66 с., 109 с.] и др., но игнорируется дескриптором накопленных разностей (т.е. одновременно учитываются все вышеперечисленные характеристики зрачка), увеличивая его точность. При

$\Delta\varphi \rightarrow 0^\circ$ точность определения дескриптора точек области будет увеличиваться, увеличивая и вычислительную сложность алгоритма. Оптимальный диапазон $\Delta\varphi$ для поиска окружностей в кадре составляет 14–40 градусов.

Таким образом, выражения (2) и (3) описывают ключевое свойство точек, составляющих некоторую монотонную темную область, обладающую высоким перепадом яркости на границах, который отражает меру равноудаленности каждой точки от границ этой области. Стоит отметить, что аккумулятор можно вычислить и для всего изображения (рисунок 2 д, ж), то есть игнорируя условие $\mathcal{B}(T; x_0, y_0) = 0$, однако в данном случае время вычисления функции многократно возрастет.

Рассмотрим процесс построения аккумуляторной функции на примере. На рисунке 2г представлена одна из точек кандидат, удовлетворяющая условию $\mathcal{B}(T; x_0, y_0) = 0$, располагающаяся недалеко от фактического центра зрачка (отмечен маленьким эллипсом). Из этой точки построим лучи с шагом $\Delta\varphi = 14,4^\circ$, что составляет $N \sim 25$ лучей.

Лучи останавливаются при выполнении условия $\nabla^2 G_\sigma(q) * f(q) > 0$, белыми крестиками отмечены граничные точки. Используем выражение (3) для вычисления значения дескриптора в данной точке и запишем значение в соответствующую точку аккумуляторной функции. Полученное значение составило $A(x_0, y_0) = 13.65$. На рисунке 2д представлены две точки-кандидаты (x_0, y_0) и (x_1, y_1) . Точка, лежащая вне границ зрачка принимает значение $A(x_1, y_1) = 31.15$, тогда как точка, располагающаяся в фактическом центре зрачка имеет наименьшее значение накопленных разностей и составляет $A(x_0, y_0) = 2.54$. Аккумуляторная функция, рассчитанная для точек с условием $\mathcal{B}(T; x_0, y_0) = 0$, представлена на рисунке 2е, центром зрачка является точка $\arg \min_{(x,y)} A(x, y)$.

Аккумулятор, вычисленный для всего изображения с нормированием длин лучей сетки, представлен на рисунке 2ж. Как видно из аккумулятора, блики от инфракрасной подсветки игнорируются, так как не являются достаточно темными и округлыми областями в сравнении со зрачком, хотя и имеют высокий перепад яркости на границе.

Шаг 4. Определение минимума аккумулятора и центра зрачка.
Положение минимума аккумуляторной функции соответствует цен-

тру наиболее округлого объекта, т.е. центру зрачка (\bar{x}, \bar{y}) . Если значение минимума больше порога T_c , будем считать, что на изображении нет глаза:

$$(\bar{x}, \bar{y}) = \begin{cases} \arg \min_{(x,y)} A(x,y), & \text{if } T_c \geq \min_{(x,y)} A(x,y) \\ \text{none}, & \text{otherwise} \end{cases}. \quad (4)$$

Если точек максимума несколько, они образуют область интереса F , найдем их геометрический центр, он и будет являться центром зрачка.

$$\bar{x} = \frac{1}{\sum A(x,y) \in F} \sum A(x,y)x, \quad \bar{y} = \frac{1}{\sum A(x,y) \in F} \sum A(x,y)y.$$

На рисунках 2 е, ж представлен минимум аккумуляторной функции. На рисунке 2з представлен найденный центр зрачка с использованием дескриптора накопленных разностей, центр отмечен ярким белым крестиком. Эллипс вокруг зрачка вписан в граничные точки множества \hat{P} (см. листинг 1), используя метод наименьших квадратов (МНК). Серый крестик слева – центр зрачка, рассчитанный методом проекций яркостей, серый крестик справа – преобразование Хафа. Исходя из профиля границы зрачка (рис. 3и), составленного лучами сетки в центрах зрачка, найденных различными методами, можно сказать, что дескриптор накопленных разностей находит точку, которая является наиболее равноудаленной от границ зрачка и максимально близко аппроксимирует профиль, составленный из фактического центра, отмеченного медицинским экспертом.

3. Постановка задачи регистрации зрачка как задачи многомерной оптимизации и ее решение методом Хука-Дживса. Метод деформируемой сетки (Deformable mesh method). Нахождение аккумуляторной функции для каждого кадра неоптимальная и ресурсоемкая задача даже при обходе пикселей, удовлетворяющих условию $\mathcal{B}(T; x, y) = 0$. Для ускорения работы метода представим задачу нахождения зрачка с использованием предложенного выше дескриптора в виде задачи многомерной оптимизации, где необходимо найти такую точку (x, y) – предполагаемый центр зрачка, в котором накопленные разности длин лучей достигнут своего минимума, т.е. найдем координаты наиболее равноудаленной точки от границ темной монотонной области.

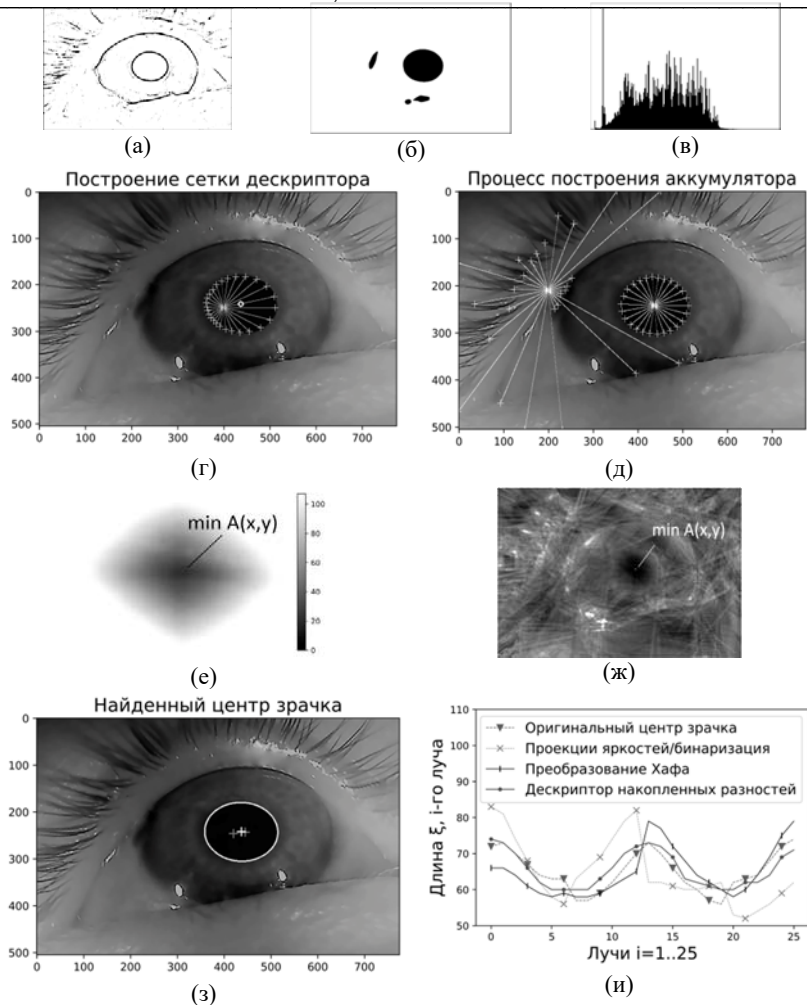


Рис. 2. Процесс построения аккумуляторной функции и нахождения зрачка в кадре: а) Лапласиан – Гауссиана изображения f ; б, в) пороговая обработка и гистограмма изображения; г, д) примеры построения лучей сетки (белыми крестиками отмечены граничные точки); е) аккумуляторная функция, рассчитанная для точек $B(T; x_0, y_0) = 0$; ж) полный аккумулятор изображения с проведением нормировки; з) найденный центр зрачка дескриптором накопленных разностей (яркий белый крестик) и эллипс вокруг зрачка, вписанный в граничные точки множества \bar{P} ; и) профиль границы зрачка, составленный лучами сетки в центрах зрачка, найденных различными методами

Другая интерпретация предложенного алгоритма – деформирование построенной сетки таким образом, чтобы центр этой сетки был наиболее равноудаленной точкой от границ, а длины ребер сетки приняли равное значение (рис. 3д):

$$A(\tau) = \lim_{\Delta\varphi \rightarrow 0} \sum_{\varphi=0}^{2\pi-\Delta\varphi} \left| \xi(\tau, P(\tau, \varphi)) - \xi(\tau, P(\tau, \varphi + \Delta\varphi)) \right| \Delta\varphi \rightarrow \min_{\tau}.$$

Так как целевая функция $A(x, y)$, выражающая дескриптор накопленных разностей, не является дифференцируемой по x, y , использование градиентных методов не представляется возможным. Воспользуемся методом многомерного поиска Хука-Дживса [22, 212 с.], который состоит из двух этапов: исследующего прямого поиска в окрестности базисной точки (используется информация только о значениях функции в точках) и, в случае его успеха, поиска по образцу, в направлении, выбранном для минимизации [23, 103 с.].

В качестве начальной точки возьмем центр зрачка $\bar{b}^0 = (x^0, y^0)$, вычисленного на предыдущем кадре, либо для повышения устойчивости метода возьмем любую точку области зрачка, взятую из разности двух кадров, взятых в моменты времени t и $t + \Delta t$, неподвижные составляющие взаимно уничтожатся, а ненулевые значения в точках, удовлетворяющих условию $\mathcal{B}(T; x, y) = 0$ останутся в местах, соответствующих подвижным компонентам изображений – пикселям зрачка то есть $|f(x, y, t) - f(x, y, t + \Delta t)| \leq T$. Если кадр первый, можно вычислить дескриптор накопленных разностей для всего кадра, тем самым найти центр зрачка, от которого и начнет работу алгоритм трекинга.

Зададим величину шага h_i , $i=1,2$ по каждой координате, используемой при исследовании функции. Эмпирическим путем было установлено, что для поиска центра зрачка площадью $1/K$ от размера кадра, оптимальный шаг составит $h \in [1; K]$ пикселей.

Произведем исследующий поиск типа 1 [22, 212 с.] для определения удачного направления к $\mathit{loc} \min_{x,y} A(x, y)$ – новому центру зрачка.

Для этого необходимо вычислить значение целевой функции $A(x, y)$

в базисе \bar{b}^0 . Изменим в циклическом порядке каждую координату (каждый раз только одна), $x^1 = x^0 + h_1$ и вычислим $A(x^0 + h_1, y^0) = A(\bar{b}^1)$. В случае, если $A(\bar{b}^1) < A(\bar{b}^0)$, перемещение по координатной области оказалось удачным и базисная точка \bar{b}^0 заменяется на \bar{b}^1 . Если перемещение не улучшило целевую функцию, т.е. $A(\bar{b}^1) \geq A(\bar{b}^0)$, то x^0 изменяется на $-h_1$, так, что $x^1 = x^0 - h_1$ и значение $A(\bar{b}^1)$ проверяется как и ранее. Если ни $x^0 + h_1$, ни $x^0 - h_1$ не улучшают $A(\bar{b}^0)$, то базис оставим без изменений. Повторим процедуру для второй координаты y . В результате поиска получим новую базисную точку, соответствующую наименьшему значению целевой функции в исследуемом поиске. Если уменьшение функции не достигнуто, исследование повторяется в окрестности предыдущей базисной точки с уменьшенной длиной шага h по координатам. Если $\bar{b}^0 \neq \bar{b}^1$ – гарантируется, что в новом базисе \bar{b}^1 значение функции меньше и можно приступить к поиску по образцу.

Разумно переместиться из \bar{b}^1 в направлении вектора $\bar{b}^1 - \bar{b}^0$, так как поиск в этом направлении уже привел к уменьшению значения функции накопленных разностей. Сделаем шаг в «ускоряющем» направлении и вычислим точку образца (временный базис) $\bar{\eta}^1 = \bar{b}^1 + \alpha(\bar{b}^1 - \bar{b}^0)$, где $\alpha \geq 1$ – коэффициент усиления. В алгоритме использовано значение $\alpha = 1$, при использовании большего коэффициента алгоритм осуществляет переход ко временному базису, располагающемуся вне границ зрачка, что в дальнейшем делает невозможным поиск локального минимума исследуемой области. Осуществим исследующий поиск в окрестности \bar{b}^1 – поиск типа 2 [22, 214 с.], неудача или успех его поиска оцениваются путем сравнения полученной точки с временным базисом $\bar{\eta}^1$. В результате поиска типа 2 получим \bar{b}^2 . Сравним значения целевой функции $A(\bar{b}^2) < A(\bar{b}^1)$, если поиск по образцу успешен, то новым базисом будет \bar{b}^2 ,

после чего проведем новый поиск по образцу с ускоряющим шагом в направлении $\bar{b}^2 - \bar{b}^1$ и исследующим поиском типа 2. Если $A(\bar{b}^2) \geq A(\bar{b}^1)$ – поиск по образцу неудачен и из предыдущего базиса проведем исследующий поиск типа 1 для определения нового удачного направления.

Условием останова на j -ой итерации является уменьшение всех величин шагов $h_i < \epsilon$. Невозможность уменьшения целевой функции при очень маленьких h_i указывает на нахождение в текущем базисе локального минимума, т.е. $\bar{b}^j = \arg \min_{x,y} A(x,y)$. В случае если h – константа, условие останова выглядит как $\bar{b}^{j+1} = \bar{b}^j$, т.е. поиск типа 1 не минимизирует целевую функцию и базис не меняется.

На рисунке 3 а, б, в, г представлен процесс трекинга зрачка с использованием метода деформируемой сетки. Тестирование проводилось на пользователе в режиме реального времени. Формат видеосъемки составил 850×554 пикселей. В качестве источника инфракрасной подсветки глаза были выбраны светодиодные элементы в количестве 2 штук с пиковой длиной волны 940 нм., подключенные параллельно через резисторы с сопротивлением в 120 Ом к источнику питания камеры. За начальную точку спуска к $\text{loc min } A(x,y)$ был принят центр зрачка предыдущего кадра $\bar{b}^0 = (x^0, y^0)$. Используемые в данном примере параметры алгоритма были подобраны эмпирическим путем: шаг между лучами сетки составил $\Delta\varphi = 14,4^\circ$ ($N \sim 25$ лучей), коэффициент усиления $\alpha = 1$, начальный шаг по двум координатам $h_{x,y} = 3$. Алгоритм в среднем сошел за 12-21 базисные точки (зеленые окружности в кадре), и 54-94 точки во время исследующего поиска типа 1 и 2, не считая построения лучей по алгоритму Брезенхэма.

На рисунке 3д представлен процесс растяжения деформируемой сетки для кадра (г). Белая сетка построена из центра зрачка предыдущего кадра $\bar{b}^0 = (227, 456)$, в этой точке дескриптор составляет $A(\bar{b}^0) = 33.17$. В процессе оптимизации сетка начинает «расправляться» к новому минимуму так, что центр этой сетки становится наиболее равноудаленной точкой от границ, он и является центром зрачка. Найденный центр после оптимизации составляет $(\bar{x}, \bar{y}) = (243, 510)$, значение аккумулятора в центре $A(\bar{x}, \bar{y}) = 5.02$.

Старт (200, 280), $A(x,y)=45.65$
 Центр (233, 337), $\text{loc min } A(x,y)=1.89$



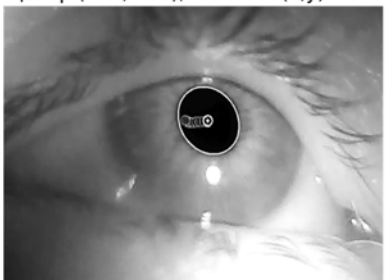
(a)

Старт (233, 337), $A(x,y)=54.51$
 Центр (226, 403), $\text{loc min } A(x,y)=3.4$



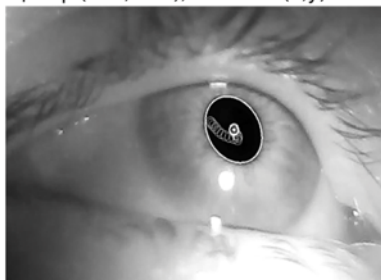
(б)

Старт (226, 403), $A(x,y)=39.86$
 Центр (227, 456), $\text{loc min } A(x,y)=4.73$



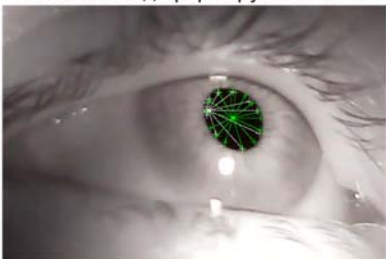
(в)

Старт (227, 456), $A(x,y)=33.17$
 Центр (243, 510), $\text{loc min } A(x,y)=5.02$



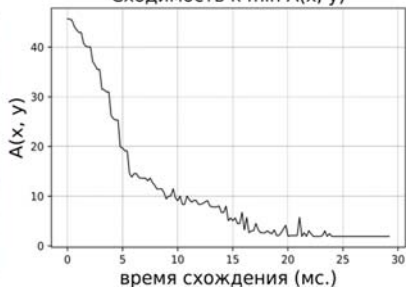
(г)

Растяжение деформируемой сетки



(д)

Сходимость к $\text{min } A(x, y)$



(e)

Рис. 3. Трекинг зрачка в режиме реального времени: а, б, в, г) процесс трекинга зрачка, используя метод деформируемой сетки (начальной точкой спуска считается центр зрачка предыдущего кадра); д) процесс растяжения сетки от начальной точки спуска до фактического центра зрачка. е) сходимость метода деформируемой сетки к минимуму аккумуляторной функции, в котором находится центр зрачка. График представлен для кадра (а)

Можно заметить, что в процессе трекинга меняется значение минимума аккумулятора, это связано с тем, что зрачок меняет свою форму с круга (рис. 3а) на эллипс (рис. 3г), в результате чего накопленные разности увеличиваются, увеличивая и минимум функции.

В случае, если метод деформируемой сетки «теряет» зрачок, для следующего кадра вычисляется аккумулятор $A(x, y)$, либо для всего кадра, либо в наиболее темных областях т.е. $B(T; x, y) = 0$, это позволит найти новый центр зрачка, с которого будет происходить дальнейший трекинг.

Так как алгоритм Хука–Дживса позволяет спускаться только к локальному минимуму целевой функции, условие «потери» зрачка в кадре определим следующим образом:

$$\left[\operatorname{loc\,min}_{x,y} A(x, y) < T_c \right] (1 - B(T; x, y)) = 0, \quad (2.2)$$

где $[\cdot]$ – возвращает 1 при истинности логического выражения, T_c – порог, определяющий наличие зрачка в кадре (т.е. наиболее круглого объекта), описан в выражении (1.9). Данное условие реализует конъюнктивный оператор на множестве логических значений. Результат обработки потери зрачка в кадре представлен в таблице 1, где:

А - зрачок (темная область, округлая, монотонная, высокий перепад яркости на границе);

Б - склера, кожный эпителий, радужка (как правило светлые области, без высоких перепадов яркости на границе);

В - ресницы (темные области, возможны резкие перепады, неправильная геометрическая форма);

Г - тень на кожном эпителии (темные области, не имеют ярких перепадов, неправильной формы);

Д - блики от инфракрасной подсветки (светлые области, имеют высокий перепад яркости на границе, как правило округлая форма).

Таблица 1. Результат обработки потери области интереса

Область	$\left[\underset{x,y}{\operatorname{loc\,min}} A(x,y) < T_c \right]$ Выражает округлость области, монотонность, наличие высокого перепада яркости на границе	$1 - \mathcal{B}(T; x, y)$ Темная область	Результат выражения, где 0 – потеря зрачка
А	1	1	1
Б	0	0	0
В	0	1	0
Г	0	1	0
Д	1	0	0

В общем случае метод деформируемой сетки (deformable mesh method), предложенный в данной работе, на языке псевдокода представлен на листинге 1.

4. Сравнительный анализ предложенного и существующих методов. Произведен сравнительный анализ предложенного метода с основными алгоритмами поиска зрачка. Результаты представлены в таблице 2. Изображения глаз были взяты из базы CASIA-IrisV4, собранной Центром исследований биометрии и безопасности Института автоматизации Китайской академии наук [24] и просмотрены медицинским экспертом офтальмологической клиники, который указал центр (x_c, y_c) на каждом из изображений. Выбор базы CASIA-IrisV4 обусловлен ее вариативностью: высококонтрастные изображения, изображения с различными типами бликов от инфракрасной подсветки, затемненные изображения, изображения глаз, сделанные на дистанции от 2.4 до 3 метров (данная база особенно актуальна для тестирования алгоритмов на базе смартфонов или устройств, находящихся на некотором отдалении от глаз пациента), синтезированные изображения. Размер входных изображений составлял 640×480 пикселей. (x_c, y_c) считается истинной точкой и служит для проверки метода на конкретном изображении.

Предложенным методом вычислялся центр зрачка (\bar{x}, \bar{y}) на изображении. Далее происходил расчет средней абсолютной ошибки с использованием Евклидовой метрики, т.е. расстояния вычисленного центра от истинного, выраженной в пикселях:

$$E_{err} = \left((x_c - \bar{x})^2 - (y_c - \bar{y})^2 \right)^{\frac{1}{2}}.$$

```

1: Function AccumulatedDifference (p)
2:   Input: Точка p, в которой рассчитываются накопленные разности
3:   Output: Величина функции s, граничные точки P
4:    $\bar{p} = [ ]$ ,  $s = 0$ 
5:   If  $f(x_0, y_0) \leq T$  then
6:     Строим N лучей, из  $(x_0, y_0)$ , угол  $\theta_i$  между осью OX и лучом i равен
7:      $\varphi_i = \varphi_{i-1} + \Delta\varphi$ .  $\varphi \in [0, 2\pi]$ 
8:      $y + y_0 = k(x - x_0)$ , где  $k = \tan \varphi$ .
9:     For каждого i-го луча do
10:      For каждой точки  $(x, y)$ , лежащей вдоль направления луча i do
11:        If  $\nabla^2 G_\sigma(x, y) * f(x, y) > 0$  then
12:          Останавливаем движение по направлению луча i
13:          Помещаем граничную точку  $q^i = (u, v)$  во множество P.
14:        For каждой  $q^i$  из множества P do
15:           $A(x_0, y_0) = A(x_0, y_0) + \left| \xi((x_0, y_0), q^i) - \xi((x_0, y_0), q^{i+1}) \right| \Delta\varphi$ 
16:        Return  $(\bar{x}, \bar{y})$ ,  $\bar{p}$ 
17:   Function SearchBasis ( $\bar{\sigma}^0$ ,  $h_i$ ,  $A(x, y)$ )
18:   Input: Базис  $\bar{\sigma}^0 = (x_i^0, x_j^0)$ ,  $h_i$  – шаги по x, y
19:   Output: Новая базисная точка  $\bar{\sigma}^1$  и массив P граничных точек
20:   For каждой точки  $x_i^0$  do
21:     If  $A(\dots, x_i^0 + h_i, \dots) < A(\bar{\sigma}^0)$  then
22:        $\bar{\sigma}^1 = (\dots, x_i^0 + h_i, \dots)$ 
23:     Else
24:       If  $A(\dots, x_i^0 - h_i, \dots) < A(\bar{\sigma}^0)$  then
25:          $\bar{\sigma}^1 = (\dots, x_i^0 - h_i, \dots)$ 
26:       Else
27:         If  $\max_i h_i < \epsilon$  then
28:           Return  $\bar{\sigma}^0$ ,  $\bar{p}$ 
29:         Else
30:            $h_i = 1$ 
31:       Return  $\bar{\sigma}^1$ 
32:   Function DeformableMeshMethod ( $f(x, y)$ ,  $\bar{\sigma}$ , N,  $\alpha$ ,  $h_i$ ,  $T_c$ )
33:   Input: Кадр f,  $\bar{\sigma} = (x, y)$ , N лучей,  $\alpha \geq 1$ ,  $h_i [1, K]$   $i = 1, 2, T_c$ 
34:   Output: Координата центра зрачка  $(\bar{x}, \bar{y})$ , набор граничных точек P,
35:    $\mathcal{S}_{prev} = \bar{\sigma}$ 
36:   While True do:
37:      $\mathcal{S}_{new} = \text{SearchBasis}(\bar{\sigma}, h_i, \text{AccumulatedDifference})$ 
38:     If  $\bar{\sigma} == \mathcal{S}_{new}$  then
39:       If  $[A(\bar{\sigma}) < T_c] (1 - B(T; x, y)) > 0$  then
40:         Return  $\bar{\sigma}$ ,  $\bar{p}$ 
41:       Else
42:         p = Расчет аккумулятора для изображения
43:         DeformableMeshMethod(p)
44:       While True do
45:          $q = \bar{\sigma}_{new} + \alpha(\bar{\sigma}_{new} - \bar{\sigma})$ 
46:          $\ell = \text{SearchBasis}(q, h_i, \text{AccumulatedDifference})$ 
47:         If AccumulatedDifference ( $\ell$ ) < AccumulatedDifference ( $\bar{\sigma}$ ) then
48:            $\mathcal{S}_{prev} = \bar{\sigma}$ 
49:            $\bar{\sigma} = \ell$ 
50:         Else break

```

Листинг 1. Метод деформируемой сетки. В качестве дескриптора нахождения центра зрачка используются накопленные разности длин ребер сетки, в качестве оптимизатора выступает метод Хука-Дживса

Таблица 2. Сравнительный анализ методов трекинга зрачка

Метод (использованная база изображений CASIA-IrisV4)	Вероятность правильного определения зрачка % (с погрешностью в 5-10 пикселей)		Среднее время вычислений, мс.	Возвращаемые параметры
Предложенный метод – Метод деформируемой сетки с использованием дескриптора накопленных разностей. $\varphi = 14,4^\circ$ $h_{x,y} = 3$ $\alpha = 1$	В режиме трекинга зрачка – оптимизация, без обхода всего изображения $f(x,y)$	Расчёт аккумулятора $A(x,y)$ всего изображения и поиск его $\max_{x,y} A(x,y)$	27,875 мс. в режиме трекинга. Для замера времени использовалась высокоточная библиотека chrono в C++11.	(x,y,r,\bar{P}) – центр зрачка и радиус (средние длины лучей в центре зрачка) \bar{P} – множество граничных точек
	99.15%	96.7%		
Метод рекурсивной эрозии [8]	90%		453 мс (с линейным обходом, для проверки оставшегося пикселя)	(x,y,r) r – радиус зрачка, количество итераций морфологической эрозии
Преобразование Хафа в двумерном пространстве параметров [14]	95.3%		249 мс.	(x,y)
Преобразование Хафа (Circular Hough Transform) [15-16], использовалась библиотека OpenCV в которой определен данный алгоритм	95% Возможно появление ложных окружностей (зависит от пределов поиска окружностей, т.е. минимального и максимального радиуса, установленных в алгоритме).		83,34 мс – для поиска окружности заданного радиуса (в диапазоне время значительно увеличивается).	(x,y,r)

Продолжение таблицы 2

Алгоритм Starburst [10]	74.4%	142 мс.	(x, y, \hat{P}) \hat{P} – множество граничных точек
Морфологическая обработка, контурный анализ и сверточная нейронная сеть YOLO v3 в качестве первичного детектора ROI (региона интереса) глаза в кадре [6]	98.7%	194,7 мс	(x, y, r)
Проекция яркости [8, 66 с.], бинаризация и центр масс [25].	83.1%	12-14 мс.	(x, y)
Адаптивная бинаризация на основе исследования процентильной функции [7]	96.5%	28 мс.	(x, y, r)
Гамма-коррекция, математическая морфология и преобразование Хафа [15]	97.44%	98 мс.	(x, y, r)

5. Заключение. Предложенный в данной статье способ поиска зрачка в кадре с использованием нового предложенного дескриптора накопленных разностей и нахождения аккумуляторной функции изображения обладает высокой точностью по сравнению с другими рассмотренными методами, но имеет большую вычислительную сложность.

Представив задачу трекинга зрачка с использованием дескриптора в виде задачи многомерной оптимизации, для ее решения был применен метод Хука–Дживса, который позволил не производить полный расчет аккумулятора для каждого кадра, а осуществить быстрый поиск

локального минимума аккумуляторной функции на основе координат центра зрачка, полученных с предыдущего кадра (начальная точка схождения), что дало прирост в скорости выполнения до 28 мс. на 1 кадр, при разрешении съемки в 640×480 пикселей. Данная скорость трекинга зрачка не является предельной и зависит от многих параметров, подбираемых эмпирическим путем: шаг h , коэффициент усиления α , начальная точка схождения, количество лучей сетки и точек, используемых для построения лучей.

В дальнейшем планируется произвести сравнительный анализ других численных методов оптимизации, таких как методы нулевого порядка (Нелдера–Мида, Розенброка), биоинспирированные методы оптимизации (дифференциальная эволюция, метод роя частиц и др.) [26], градиентные методы с конечно-разностными аппроксимациями с целью увеличения скорости и точности алгоритма, а также провести распараллеливание алгоритма с использованием технологии OpenCL. Также планируется детально рассмотреть вопросы устойчивости предложенного метода при обработке изображений низкого качества (зашумленные, низкая контрастность) и изображений, в которых частично или полностью отсутствует область интереса (естественное закрытие глаза, наличие линз очков между устройством айтрекером и глазом).

Метод деформируемой сетки позволяет производить поиск параметров зрачка на изображении в трехмерном пространстве параметров (x, y, r) , что дает возможность использовать алгоритм в задаче альтернативной аугментативной коммуникации для людей с нервно-паралитическими синдромами и другими болезнями ЦНС, такими как синдром «запертого человека» МКБ-10 G83.5, синдром БАС G12, спастический церебральный паралич и диплегия G80.0 и др. посредством видеоокулографии, а также в офтальмологических диагностических комплексах для определения патологии нейрорецепторного аппарата, тремора, фиксаций и саккадических движений, заболеваний зрительного нерва и др., с использованием процесса пупиллометрии, где необходимо учитывать не только положение зрачка, но и его размеры в динамике.

Использование стабильного метода оценки параметров модели на основе случайных выборок (RANSAC) позволяет увеличить точность и вероятность сходимости метода деформируемой сетки путем отброса «шумовых» точек, которые могут образовываться в процессе неправильно подобранного значения порога при исследовании гранично-ступенчатой модели (точки, прошедшие сквозь границу зрачка, влияющие

на корректный расчет дескриптора). Данный вопрос будет подробно освещен в следующей работе.

Стоит добавить, что применение предложенного в данной работе дескриптора накопленных разностей возможно и в задаче распознавания графических образов (например, при автоматической навигации беспилотного автотранспорта по знакам), используя различные вариации порога T_c . Сама концепция накопленных разностей может быть применима во время навигации и определения местоположения робототехнических устройств в замкнутых помещениях с использованием лазерных или ультразвуковых лидаров – технологии дистанционного зондирования.

Литература

1. *Masrori P., Van Damme P.* Amyotrophic lateral sclerosis: a clinical review // *European journal of neurology*. 2020. vol. 27. pp. 1918–1929.
2. *Yang Z.* Intelligent Evaluation of Strabismus in Videos Based on an Automated Cover Test // *Applied Sciences*. 2019. pp. 59.
3. *Swirski. L. Bulling. A. Dodgson. N.* Robust real-time pupil tracking in highly off-axis images // *Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA)*. 2012. pp. 173–176.
4. *Utaminigrum F., Prasetya R., Arum S.Y.* Image Processing For Rapidly Eye Detection Based On Robust Haar Sliding Window // *International Journal of Electrical and Computer Engineering*. 2017. vol. 07.
5. *Durna Y., Ari F.* Design of a Binocular Pupil and Gaze Point Detection System Utilizing High Definition Images // *Applied Sciences*. 2017. vol. 7. pp. 498.
6. *Грушко Ю.В.* Аппаратно-программный комплекс аугментативной системы коммуникации на основе технологии Eye-tracking. // *Вестник КРАУНЦ*. 2019. № 27:2. С. 55–73.
7. *Bonteanu P., Cracan A., Bonteanu G., Bozomitu R.* A Robust Pupil Detection Algorithm Based on a New Adaptive Thresholding Procedure. // *IEEE International Conference on e-Health and Bioengineering EHB*. 2019. pp. 276.
8. *Матвеев И.А.* Методы и алгоритмы автоматической обработки изображений радужной оболочки глаза // *Диссертация на соискание ученой степени доктора технических наук*. 2014.
9. *Bozomitu R. et al.* Pupil detection algorithm based on RANSAC procedure // *International Symposium on Signals, Circuits and Systems (ISSCS)*. 2017. pp. 1–4.
10. *Dongheng L., Winfield D., Parkhurst D.J.* Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches // *Paper Presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2005. vol. 3. pp. 79.
11. *Fuhl W., Santini T., Kübler T., Kasneci E.* EISE: ellipse selection for robust pupil detection in real-world environments // *The Ninth Biennial ACM Symposium*. 2016. pp. 123–130.
12. *Mohammed G., Hong B., Alkazzaz A.* Accurate Pupil Features Extraction Based on New Projection Function // *Computing and Informatics*. 2010. vol. 29. pp. 663–680.
13. *Лукошков И.А., Артемова А.А., Белов Ю.С.* Адаптивный алгоритм нахождения границ зрачка на изображении. // *Научное обозрение. Технические науки*. 2020. № 3. С. 19–23.

14. *Mameev I.A.* Поиск центра радужки на изображении методом Хафа с двумерным пространством параметров. // Известия Российской академии наук. Теория и системы управления. 2012. № 6. С. 44–51.
15. *Alkuzayy M., Alshemmary E.* Towards Accurate Pupil Detection Based on Morphology and Hough Transform // Baghdad Science Journal. 2020. vol. 17(2). pp. 583–590.
16. *Ершов Е.И.* Быстрое преобразование Хафа как инструмент анализа двумерных и трехмерных изображений в задачах поиска прямых и линейной кластеризации // Диссертация на соискание ученой степени кандидата физико-математических наук. 2018.
17. *Bresenham J.* Algorithm for Computer Control of a Digital Plotter // IBM Syst. J. 1965. vol. 4. pp. 25–30.
18. *Zongli S.* Graphics Programming Principles and Algorithms. 2017.
19. *Rafael C. Gonzalez, Richard E. Woods.* Digital image processing 4th. 2019. 1104 с.
20. *Marr D., Hildreth E.* The theory of edge detection // M.I.T. Psychology Department and artificial Intelligence Laboratory. 1980. pp. 187–217.
21. *Ansari M., Kurchaniya D., Dixit M.* A Comprehensive Analysis of Image Edge Detection Techniques // International Journal of Multimedia and Ubiquitous Engineering. 2017. vol. 12. pp. 1–12.
22. *Hooke R., Jeeves T.A.* Direct Search Solution of Numerical and Statistical Problems // Westinghouse Research Laboratories. 1961. 256 p.
23. *Медьанский М.М., Дьячук А.К.* Численные методы оптимизации с использованием MAPLE 11: Учебное пособие // М.: МАИ-ПРИНТ. 2009.
24. Chinese Academy of Sciences Institute of Automation. Iris image database, version 4. URL: <http://www.cbsr.ia.ac.cn/china/Iris20Databases20CN.asp>. (дата обращения: 2010).
25. *Mompeán J. et al.* Design of an accurate and high-speed binocular pupil tracking system based on GPGPUs // J Supercomput. 2018. vol. 74. pp. 1836–1862.
26. *Darwish A.* Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications // Future Computing and Informatics Journal. 2018. vol. 3. pp. 231–246.

Грушко Юрий Васильевич – аспирант, кафедра математики и физики, ФГБОУ ВО "Камчатский государственный университет им. Витуса Беринга". Область научных интересов: численные методы, математическое моделирование, data mining, нейронные сети, применение методов математического моделирования и численных методов в цифровой обработке медицинских изображений, программная инженерия комплексов с СУБД и многомерными хранилищами данных. Число научных публикаций – 3. neuralpill@gmail.com; ул. Пограничная, 4, 683032, Петропавловск-Камчатский, Россия; р.т.: +79006841757.

Паровик Роман Иванович – д-р физ.-мат. наук, доцент, декан, физико-математический факультет, ФГБОУ ВО «Камчатский государственный университет им. Витуса Беринга». Область научных интересов: численные методы, математическое моделирование, математическое моделирование эрдитарных процессов. Число научных публикаций – 225. romano84@mail.ru; ул. Пограничная, 4, 683032, Петропавловск-Камчатский, Россия; р.т.: +79247926209.

Поддержка исследований. Работа выполнена при грантовой поддержке Министерства образования Камчатского края, Региональный инновационный конкурс в Камчатском крае, в номинации «научно-исследовательские и инновационные проекты в области техники и технологии», приказ №123, от 29.10.2019 г. Работа выполнена в рамках темы НИР КамГУ им. Витуса Беринга №АААА-А20-120021190005-5.

Y. GRUSHKO, R. PAROVIK
**FAST PUPIL TRACKING BASED ON THE STUDY
OF A BOUNDARY-STEPPED IMAGE MODEL AND MULTIDI-
MENSIONAL OPTIMIZATION HOOK-JIVES METHOD**

Grushko Y., Parovik R. Fast Pupil Tracking based on the Study of a Boundary-stepped Image Model and Multidimensional Optimization Hook-Jives Method.

Abstract. A new fast method for pupil detection and eyetracking real time is being developed based on the study of a boundary-step model of a grayscale image by the Laplacian-Gaussian operator and finding a new proposed descriptor of accumulated differences (point identifier), which displays a measure of the equidistance of each point from the boundaries of some relative monotonous area (for example, the pupil of the eye). The operation of this descriptor is based on the assumption that the pupil in the frame is the most rounded monotonic region with a high brightness difference at the border, the pixels of the region should have an intensity less than a predetermined threshold (but the pupil may not be the darkest region in the image). Taking into account all of the above characteristics of the pupil, the descriptor allows achieving high detection accuracy of its center and size, in contrast to methods based on threshold image segmentation, based on the assumption of the pupil as the darkest area, morphological methods (recursive morphological erosion), correlation or methods that investigate only the boundary image model (Hough transform and its variations with two-dimensional and three-dimensional parameter spaces, the Starburst algorithm, Swirski, RANSAC, ElSe).

The possibility of representing the pupil tracking problem as a multidimensional unconstrained optimization problem and its solution by the Hook-Jeeves non-gradient method, where the function expressing the descriptor is used as the objective function, is investigated. In this case, there is no need to calculate the descriptor for each point of the image (compiling a special accumulator function), which significantly speeds up the work of the method.

The proposed descriptor and method were analyzed, and a software package was developed in Python 3 (visualization) and C ++ (tracking kernel) in the laboratory of the Physics and Mathematics Faculty of Kamchatka State University of Vitus Bering, which allows illustrating the work of the method and tracking the pupil in real time.

Keywords: Mathematical Models, Detectors, Descriptors, Oculography, Eye Tracking, Amyotrophic Sclerosis, Pupil, Laplacian-Gaussian, Hook-Jeeves Method, Numerical Optimization, Hough Transform

Grushko Yuriy – Ph.D. Student, Department of Mathematics and Physics, Kamchatka State University named after Vitus Bering. Research interests: numerical methods, mathematical modeling, data mining, neural networks, application of mathematical modeling and numerical methods in digital processing of medical images, soft-ware engineering of complexes with database management systems and multidimensional data warehouses. The number of publications – 3. neuralpill@gmail.com; 4, Pogranichnaya str., 683032, Petropavlovsk-Kamchatsky, Russia; office phone: +79006841757.

Parovik Roman – Ph.D., Dr.Sci., Associate Professor, Dean, Faculty of Physics and Mathematics, Kamchatka State University named after Vitus Bering. Research interests: numerical methods, mathematical modeling, Mathematical modeling of hereditary processes. The number of publications – 225. romano84@mail.ru; 4, Pogranichnaya str., 683032, Petropavlovsk-Kamchatsky, Russia; office phone: +79247926209.

Acknowledgements. This research is supported by the Ministry of Education of the Kamchatka Territory, the Regional Innovation Competition in the Kamchatka Territory, in the nomination "Research and Innovation Projects in the Field of Engineering and Technology", Order No. 123, dated October 29, 2019. them. Vitus Beringa No. AAAA-A20-120021190005-5.

References

1. Masrori P., Van Damme P. Amyotrophic lateral sclerosis: a clinical review. *European journal of neurology*. 2020. vol. 27. pp. 1918–1929.
2. Yang Z. Intelligent Evaluation of Strabismus in Videos Based on an Automated Cover Test. *Applied Sciences*. 2019. pp. 59.
3. Swirski L., Bulling A., Dodgson N. Robust real-time pupil tracking in highly off-axis images. Proceedings of the Symposium on Eye Tracking Research & Applications (ETRA). 2012. pp. 173–176.
4. Utamingrum F., Prasetya R., Arum S.Y. Image Processing For Rapidly Eye Detection Based On Robust Haar Sliding Window. *International Journal of Electrical and Computer Engineering*. 2017. vol. 07.
5. Durna Y., Ari F. Design of a Binocular Pupil and Gaze Point Detection System Utilizing High Definition Images. *Applied Sciences*. 2017. vol. 7. pp. 498.
6. Grushko Y.V. [Hardware-software complex of augmentative communication system based on eyetracking technology]. *Vestnik KRAUNC – Bulletin KRAUNZ*. 2019. vol. 27:2. pp. 55–73. (In Russ.).
7. Bonteanu P., Cracan A., Bonteanu G., Bozomitu R. A Robust Pupil Detection Algorithm Based on a New Adaptive Thresholding Procedure. IEEE International Conference on e-Health and Bioengineering EHB. 2019. pp. 276.
8. Matveev I.A. [Methods and algorithms for automatic processing of images of the iris of the eye. Dissertation for the degree of Doctor of Technical Sciences]. 2014. (In Russ.).
9. Bozomitu R. et al. Pupil detection algorithm based on RANSAC procedure. *International Symposium on Signals, Circuits and Systems (ISSCS)*. 2017. pp. 1–4.
10. Dongheng L., Winfield D., Parkhurst D.J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. Paper Presented at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005. vol. 3. pp. 79.
11. Fuhr W., Santini T., Kübler T., Kasnecki E. ElSe: ellipse selection for robust pupil detection in real-world environments. The Ninth Biennial ACM Symposium. 2016. pp. 123–130.
12. Mohammed G., Hong B., Alkazzaz A. Accurate Pupil Features Extraction Based on New Projection Function. *Computing and Informatics*. 2010. vol. 29. pp. 663–680.
13. Lukoshkov I.A., Artemova A.A., Belov Ju.S. [Adaptive algorithm for finding the boundaries of the pupil in the image]. *Nauchnoe obozrenie. Tehnicheskije nauki – Scientific Review. Technical science*. 2020. vol. 3. pp. 19–23. (In Russ.).
14. Matveev I.A. [Finding the center of the iris in the image by the Hough method with a two-dimensional space of parameters]. *Izvestija Rossijskoj akademii nauk. Teorija i sistemy upravlenija – Bulletin of the Russian Academy of Sciences. Control theory and systems*. 2012. vol. 6. pp. 44–51. (In Russ.).
15. Alkuzaay M., Alshemmary E. Towards Accurate Pupil Detection Based on Morphology and Hough Transform. *Baghdad Science Journal*. 2020. vol. 17(2). pp. 583–590.
16. Ershov E. I. [Fast Hough transform as a tool for analyzing 2D and 3D images in problems of search straight lines and linear clustering. Dissertation for the degree of candidate of physical and mathematical sciences]. 2018. (In Russ.).

17. Bresenham J. Algorithm for Computer Control of a Digital Plotter. *IBM Syst. J.* 1965. vol. 4. pp. 25–30.
18. Zongli S. Graphics Programming Principles and Algorithms. 2017.
19. Rafael C. Gonzalez, Richard E. Woods. Digital image processing 4th. 2019. 1104 с.
20. Marr D., Hildreth E. Theory of edge detection. M.I.T. Psychology Department and artificial Intelligence Laboratory. 1980. pp. 187–217.
21. Ansari M., Kurchaniya D., Dixit M. A Comprehensive Analysis of Image Edge Detection Techniques. *International Journal of Multimedia and Ubiquitous Engineering.* 2017. vol. 12. pp. 1–12.
22. Hooke R., Jeeves T.A. Direct Search Solution of Numerical and Statistical Problems. Westinghouse Research Laboratories. 1961. 256 p.
23. Medynskij M.M., D'jachuk A.K. *Chislennye metody optimizacii s ispol'zovaniem MAPLE 11: Uchebnoe posobie* [Numerical optimization methods using Maple 11: A Tutorial]. Moscow: publishing house MAI-PRINT, 2009. 288 p. (In Russ.).
24. Chinese Academy of Sciences Institute of Automation. Iris image database, version 4. Available at: <http://www.cbsr.ia.ac.cn/china/Iris20Databases20CH.asp>. (accessed: 2010).
25. Mompeán J. et al. Design of an accurate and high-speed binocular pupil tracking system based on GPGPUs. *J Supercomput.* 2018. vol. 74. pp. 1836–1862.
26. Darwish A. Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal.* 2018. vol. 3. pp. 231–246.