

АЛГОРИТМ НАХОЖДЕНИЯ РАДУЖКИ ДЛЯ СИСТЕМЫ ОТСЛЕЖИВАНИЯ НАПРАВЛЕНИЯ ВЗГЛЯДА НА ОСНОВЕ ДОСТУПНОЙ ВИДЕОАППАРАТУРЫ

Иван Малин

Факультет прикладной математики и физики
Московский авиационный институт, Москва, Россия
ivan.malin@gmail.com

Аннотация

Большое количество важной информации о человеческом поведении, интересах и распределении внимания может быть получено на основе анализа траектории движения его взгляда. Системы отслеживания направления взгляда используются с растущей популярностью в различных отраслях науки и техники. Однако большинство современных систем являются слишком дорогими и требуют специального оборудования. В данной работе предлагаются новые методы, позволяющие определить точку взгляда пользователя, применяя только доступную аппаратуру. Под доступной аппаратурой понимаются веб-, фото и видеокамеры бытового уровня.

Ключевые слова: *отслеживание взгляда, распознавание радужки, компьютерное зрение*

1. ВВЕДЕНИЕ

В настоящее время количество информации, представленной в сети Интернет, огромно, и продолжает расти с увеличивающимися темпами. Для удобства поиска и работы с ними, эти данные должны быть представлены в удобной и логичной форме. Для определения, насколько удобен для пользователя интерфейс программы или веб-сайта, проводятся исследования юзабилити. Одним из ключевых понятий подобных исследованиях является распределение внимания пользователя. В связи с этим, одним из широко используемых инструментов при проведении юзабилити-тестов являются системы отслеживания направления взгляда. Следует отметить, что это только одно из наиболее очевидных приложений подобных систем, также они используются в задачах человеко-машинного взаимодействия, маркетинговых и медицинских исследованиях, тренировочных симуляторах, системах виртуальной реальности и компьютерных играх. Большинство современных реализаций таких устройств нуждается в специальном оборудовании для корректной работы, например, инфракрасных камерах и светодиодах, стереоскопических камерах и прочих спецсредствах. Целью нашей работы является разработка метода получения точки взгляда, который может быть применен в системе, использующей доступное оборудование: веб-, фото и видеокамеры бытового уровня.

Задача определения точки взгляда по видеозаписи глаза может быть разделена на две подзадачи: выделение ряда признаков на кадре и построение функции отображения пространства этих признаков в координаты точки взгляда на экране. Разные методы предлагают различные признаки для выделения. Ими могут быть блики на роговице, создаваемые инфракрасными светодиодами (иногда установленными по определенному геометрическому шаблону), центр и границы зрачка или радужки, уголки глаз и т. д.

Наши исследования наиболее популярных методов показали, что наилучшим признаком для выделения в условиях использования доступной видеоаппаратуры является граница радужки. Зрачок не подходит в силу низкого контраста с радужной оболочкой. Фактически, получить достаточный контраст между зрачком и радужкой (особенно для людей с темными глазами) не представляется возможным без использования дополнительной подсветки, направленной на глаз. Это возможно сделать при работе в инфракрасном диапазоне, однако в видимом диапазоне направленный свет может повредить глаз.

На рисунке 1 представлены изображения глаза в разных условиях освещения.



Рис. 1. Глаз в инфракрасном диапазоне, в видимом спектре с дополнительной подсветкой, в условиях естественного освещения.

В ходе нашего исследования было испробовано несколько известных методов для выделения радужки, включая метод

Хаффа [3] и пороговую бинаризацию. Однако при их применении не была достигнута требуемая устойчивость к плохим условиям освещения. Пороговая бинаризация не подходит для выделения радужки в связи с невозможностью подобрать корректное значение порога, отделяющее радужку от век и теней около глаза. Метод Хаффа показал плохие результаты на сильно зашумленных и размытых изображениях, а также потребовал тонкой настройки параметров для выделения границ методом Кенни.

2. ПРЕДЛАГАЕМЫЙ МЕТОД

2.1. Поиск окружности

Сначала рассмотрим более подробно задачу нахождения окружности на изображении. Будем рассматривать полутоновое изображение как функцию яркости, определенную на прямоугольнике $I(x, y), x \in [0..W], y \in [0..H]$. Одним из способов нахождения окружности является определение максимума функционала $\Phi(I, x_0, y_0, r) = \int_{O_{x_0, y_0, r}} \nabla I \cdot \bar{n} dl$, где $O_{x_0, y_0, r}$ является окружностью радиуса r с центром в точке (x_0, y_0) , а \bar{n} – нормализованный радиус вектор, имеющий начало в (x_0, y_0) и конец в (x, y) . Другими словами, мы хотим найти такую окружность, что в каждой ее точке направление радиус вектора от ее центра к этой точке будет наиболее схоже с направлением градиента яркости изображения в этой же точке.

В дискретном пространстве будем рассматривать изображение как $(i, j) = I_{ij}, i = 1..H, j = 1..W$. Сначала найдем градиент изображения – рассчитаем вектор градиента яркости для каждого пиксела – путем свертки изображения с двумя масками Собела [3] горизонтального и вертикального направлений. Рисунок 3 показывает градиент изображения. Направление градиента на рисунке закодировано цветом, аналогично кодированию цветом комплексных значений в работе [4] – тон пиксела кодирует направление вектора, а яркость – длину. Будем обозначать градиент изображения как $\nabla I(i, j)$.

Затем насчитаем две маски одинакового размера – градиентную маску M_{ij} и радиальную R_{ij} . Градиентная маска – это двумерный массив, каждый пиксел которого содержит двумерный вектор единичной длины, направленный из центра маски к этому пикселу. Каждый пиксел радиальной маски содержит расстояние между центром маски и данным пикселом.

$$M_{ij} = \left(i - \frac{M_h}{2}, j - \frac{M_w}{2} \right), i = 1..M_w, j = 1..M_h$$

$$R_{ij} = \sqrt{\left(i - \frac{M_h}{2} \right)^2 + \left(j - \frac{M_w}{2} \right)^2}, i = 1..M_w, j = 1..M_h$$

Здесь i обозначат строку, j – столбец, M_h – высоту маски и M_w – ширину маски. После этого насчитывается куб данных $C_{ijr}, i = 0..H - 1, j = 0..W - 1, r = r_{min}..r_{max}$:

$$C_{ijr} = \sum_{k=1..M_h, l=1..M_w: r=R_{kl}} M_{kl} \cdot \nabla I \left(i - \frac{M_h}{2} + k, j - \frac{M_w}{2} + l \right)$$

Затем мы находим максимальное значение среди всех элементов куба, и координаты i, j и r максимального элемента дадут нам координаты центра и радиус искомой окружности. Другими словами, это просто организуемый способ исчерпывающего перебора наилучших параметров окружности.

На рисунке 2 изображены градиентная и радиальная маски.

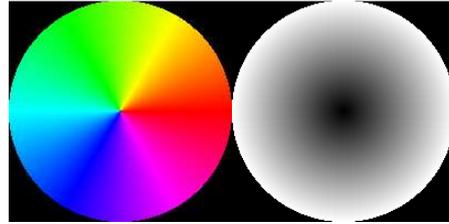


Рис. 2. Градиентная и радиальная маски.

На рисунке 3 изображены результаты работы алгоритма поиска окружности на реальном изображении глаза.

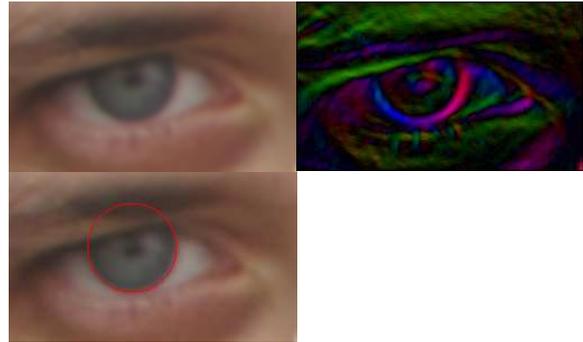


Рис. 3. Исходное изображение, градиент изображения, найденная окружность.

Мы также протестировали алгоритм на сильно зашумленных и размытых синтезированных изображениях, где он также показал хороший результат. Один из них представлен на рисунке 4.

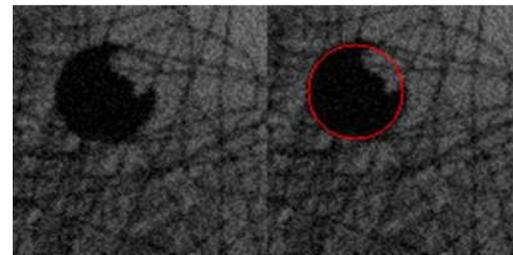


Рис. 4. Результат работы алгоритма определения окружности на сильно размытом и зашумленном синтезированном изображении.

2.2. От окружности к эллипсу

В общем случае проекцией окружности на плоскость является эллипс. Поэтому описанный выше метод не может дать точное положение радужки, если угол между оптическими осями глаза и камеры велик. Однако его можно использовать для поиска некоторого приближения искомого эллипса. Уточнение параметров эллипса может быть осуществлено путем простого исчерпывающего перебора в пространстве параметров с некоторым заданным шагом. При этом целевая функция похожа на используемую при поиске окружности – это свертка с градиентной маской. Но теперь у нас есть целый набор градиентных масок вместо одной – по одной для каждой комбинации параметров эллипса. Мы так же совмещаем центры масок последовательно с каждым пикселем изображения и суммируем скалярные произведения векторов градиентной маски и изображения для всех пикселей, накрытых маской. Маска и ее положение с максимальным откликом определяют параметры наилучшего эллипса.

На рисунке 5 изображены некоторые градиентные маски, используемые для поиска эллипса. Рисунок 6 показывает результат работы алгоритма поиска эллипса. Красная окружность на рисунке показывает предварительно найденное приближение, результирующий эллипс нанесен зеленым цветом.



Рис. 5. Эллиптические градиентные маски.

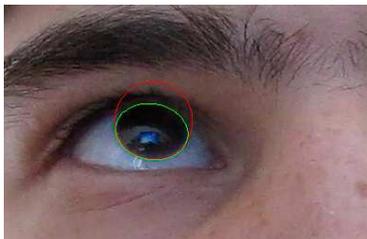


Рис. 6. Результат работы алгоритма определения эллипса.

Данная задача отнимает наибольшую часть времени вычисления в процессе поиска. Эллипс определяется пятью независимыми параметрами, например, координатами точки центра, длиной наибольшей оси, отношением длин осей и углом поворота наибольшей оси. Вычисление требует $O(M \cdot N \cdot m^2 \cdot K)$ операций умножения, где M и N определяют размеры области поиска центра эллипса, m обозначает длину маски в пикселях, а K – количество используемых масок.

В приведенном выше примере мы уменьшили размер изображения таким образом, чтобы радиус радужки примерно был равен 25 пикселям. Поиск осуществлялся в области 20×20 пикселей вокруг центра предварительно

найденной окружности. Было применено около 2000 градиентных масок, соответствующих различным параметрам эллипса. Процесс поиска занял 500 мс на процессоре AMD Turion 2GHz.

В дальнейшей работе над методом следует уделить внимание его ускорению. В частности, следует опробовать методы численной оптимизации, работающие быстрее полного перебора.

2.3. Функция отображения

Для определения координат точки взгляда на экране по координатам центра зрачка в видеокadre, требуется определить соответствующее преобразование. Для построения такого преобразования требуется сперва провести процедуру калибровки. Пользователю предлагается посмотреть последовательно на девять калибровочных точек: четыре по краям экрана, четыре в серединах сторон и одна в центре. В результате данной процедуры мы получим девять пар $\langle S_i, F_i \rangle, i = 1..9$, где S_i – точки взгляда на экране, а F_i – центр зрачка на видеокadre (точка S_9 совпадает с точкой O).

Теперь определим две полярные системы координат. Одна из них лежит в плоскости экрана, другая – в плоскости кадра. Началом координат первой системы является центр экрана. Началом второй – точка F_9 (см рис. 7). Будем обозначать центр зрачка в кадре как (φ, L) , а точку взгляда на экране как (φ', L') . L и L' являются расстояниями до точки от начала координат соответствующей системы. φ и φ' – углы поворота радиус-вектора. Для системы координат кадра угол отсчитывается по часовой стрелки относительно направления OF_1 , для системы координат экрана – против часовой относительно OS_1 .

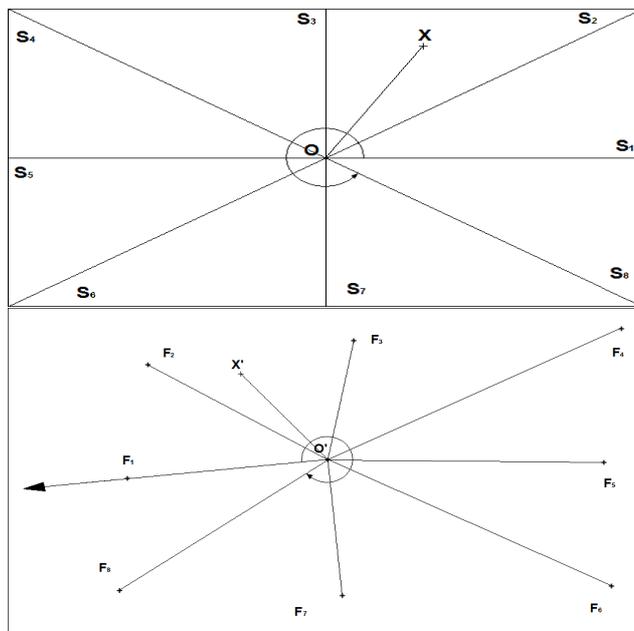


Рис. 7. Экранная система координат и система координат кадра.

Теперь определим простое правило для преобразования координат из системы кадра в систему снимка.

$$\varphi = \frac{\varphi' - \varphi'_i}{\varphi'_{i+1} - \varphi'_i} (\varphi_{i+1} - \varphi_i) + \varphi_i$$
$$L = \left(\frac{\varphi' - \varphi'_i}{\varphi'_{i+1} - \varphi'_i} \cdot \frac{L_{i+1}}{L'_{i+1}} + \frac{\varphi'_{i+1} - \varphi'}{\varphi'_{i+1} - \varphi'_i} \cdot \frac{L_i}{L'_i} \right) L'$$

3. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Приведем некоторые экспериментальные результаты работы наших алгоритмов. В качестве входных данных использовалась видеозапись с разрешением 640x480 пикселей, сделанная фотокамерой Canon Powershot SX20. Кадры были уменьшены до размера 320x240 пикселей, и вручную была выделена область обработки вокруг глаза для увеличения скорости детектирования радужки. Запись видео производилась в условиях электрического освещения лампой накаливания без дополнительной подсветки. На рисунке 8 показаны результаты работы алгоритма выделения радужки. Рисунок 8 показывает работу функции отображения координат. Черными квадратами обозначены точки на экране, в которые предлагалось посмотреть пользователю (включая калибровочные), а серыми линиями нанесена вычисленная траектория точки взгляда. Пользователь находился в 60 см от экрана разрешением 1280x800 пикселей. Максимальная ошибка составила 95 пикселей, что составляет 6.2% от диагонали экрана. Среднеквадратическая ошибка составила 48 пикселей (3.2% диагонали экрана).

4. ДАЛЬНЕЙШАЯ РАБОТА

Результаты экспериментов показали, что предлагаемые алгоритмы могут быть использованы для простой и дешевой системы отслеживания направления взгляда, работающей на основе видеоаппаратуры бытового уровня. Однако все еще имеется ряд проблем, требующих решения. Во-первых, метод работает медленно, что описано в разделе, посвященном определению эллипсов. Возможно, использование подходящего метода стохастической оптимизации вместо перебора по сетке для определения параметров эллипса позволило бы ускорить работу. Также алгоритм легко поддается распараллеливанию. Второй большой проблемой является необходимость соблюдать неподвижность головы пользователя для корректной работы функции отображения. Использование более сложной геометрической модели и отслеживание движений головы потенциально может позволить системе не ограничивать свободу движений пользователя.

ЛИТЕРАТУРА

- [1] Ying Huang, Zhiliang Wang, and An Ping “Non-Contact Gaze Tracking with Head Movement Adaptation based on Single Camera”, *World Academy of Science, Engineering and Technology* 59 2009
- [2] Dong Hyun Yoo Myung Jin Chung, “Non-contact Eye Gaze Estimation System using Robust Feature Extraction and Mapping of Corneal Reflections”
- [3] Gary Bradski, Adrian Kaehler, “Learning OpenCV”, *O’Reilly Media*, 2008
- [4] Bjorn Johansson, “Rotational Symmetries a Quick Tutorial”, 2003

Об авторе

Иван Малин – аспирант кафедры вычислительной математики и программирования Московского авиационного института.

e-mail: ivan.malin@gmail.com