

УДК 004.93

МЕТОД РАСПОЗНАВАНИЯ ЭФФЕКТА КРАСНЫХ ГЛАЗ НА ЦИФРОВЫХ ФОТОГРАФИЯХ

А. В. Агаян¹, Г. М. Агаян¹, С. Ю. Романов¹

Работа посвящена разработке алгоритмов распознавания эффекта красных глаз, возникающего при фотосъемке людей со вспышкой. В качестве входных данных используются оцифрованные фотографии в формате RGB. Алгоритм распознавания состоит из предварительного этапа, этапа конечнопараметрической фильтрации и этапа контекстного поиска. После отыскания областей красных глаз применяется коррекция эффекта. Алгоритм опробован на многочисленной группе фотографий, снятых в различных условиях, и показал свою высокую эффективность для фотографий, на которых размер зрачков достаточно большой.

Ключевые слова: распознавание образов, цифровые фотографии, эффект “красного глаза”, конечнопараметрическая модель.

1. Введение. В настоящее время фотографирование — это сложный технологический процесс, в котором используется множество различных приспособлений. Наиболее распространенным способом улучшения качества изображения является использование фотовспышки. Однако этот способ имеет и свои отрицательные стороны — возникает так называемый эффект красных глаз, состоящий в том, что у человека вместо зрачков получаются красные круги (рис. 1). Методам распознавания данного эффекта на цифровых фотографиях и посвящена настоящая работа.

Возможны различные пути решения поставленной задачи. Первым, и самым распространенным, является аппаратный путь. Он заключается в оснащении фотоаппарата устройством, уменьшающим проявление указанного эффекта. Это устройство выполняет несколько последовательных вспышек, что позволяет глазам приспособиться к яркому свету, и эффект красных глаз значительно уменьшается. Данный метод имеет как преимущества, так и недостатки.

Первым недостатком является энергоемкость метода. Значительно уменьшается время работы блоков питания фотоаппаратов. Второй существенный недостаток — невозможно скорректировать уже существующие фотографии.

Альтернативным методом решения обсуждаемой проблемы является программный подход, основанный на компьютерной обработке оцифрованных снимков и не требующий технических усовершенствований фотоаппаратов. Этому подходу и посвящена наша статья.

К сожалению, бумажная версия статьи содержит черно-белые иллюстрации. Поэтому наиболее полную информацию можно получить, прочитав ее электронную версию.

2. Постановка задачи. Решение задачи в общем виде представляется очень сложным. Основная трудность заключается в построении математической модели и разработке методов решения, которые давали бы приемлемые результаты для большинства обрабатываемых картинок.

Предлагается рассматривать следующую постановку задачи. Пусть имеется оцифрованное RGB-изображение. Красный, зеленый и синий каналы этого изображения моделируются неотрицательными функциями $R(x, y)$, $G(x, y)$ и $B(x, y)$, меняющимися от 0 до 1 (здесь x и y — координаты точки на прямоугольной картинке). Предположим, что на картинке имеется эффект красных глаз, т.е. глаза имеют неестественный красный оттенок. Требуется найти эти места и откорректировать эффект.



Рис. 1. Исходное изображение

¹ Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, 119992, Москва; e-mail: agagal@rambler.ru; romanov@srcc.msu.ru

© Научно-исследовательский вычислительный центр МГУ им. М. В. Ломоносова

Математически красный оттенок означает непропорционально большое значение в точке (x, y) функции $R(x, y)$ по сравнению с $G(x, y)$ и $B(x, y)$. Будем предполагать, что искомый эффект занимает область в виде круга радиуса r . Кроме того, будем предполагать, что за пределами круга расположены белки глаз, а в центре этого круга имеется светлое пятно (отражение вспышки).

В соответствии с такой априорной информацией строится последовательность этапов обработки изображения.

3. Подготовительный этап. На данном этапе используется информация о том, что искомые области имеют красный оттенок, и отбрасываются области, заведомо не подходящие по цвету.

Введем функцию $\text{Redness}(x, y) = \frac{R^2(x, y)}{G^2(x, y) + B^2(x, y) + K}$, где K — некоторая положительная константа [1]. Из вида функции следует, что ее значение тем больше, чем больше значения красного канала и чем меньше значения других каналов.

Фильтруем эту функцию по некоторому уровню M . Введем функцию, равную 1 там, где выполнено неравенство $\text{Redness}(x, y) > M$, и 0 в остальных точках. Такая фильтрация часто отбрасывает центры глаз (отражение вспышки необязательно красного оттенка), поэтому введем еще одну функцию $A(x, y) = R(x, y) + G(x, y) + B(x, y)$ и вычислим Лапласиан от A : $\Delta A(x, y) = A_{xx} + A_{yy}$ [2].

Фильтруем полученный массив по некоторому уровню. Таким образом, остаются только те области, в которых наблюдаются резкие перепады функции A (иными словами, находим области центров глаз, где отразилась вспышка; см. рис. 2).

Дополняем области изображения преимущественно красного цвета областями, отфильтрованными с помощью Лапласиана. На этом завершается подготовительный этап распознавания (рис. 3).

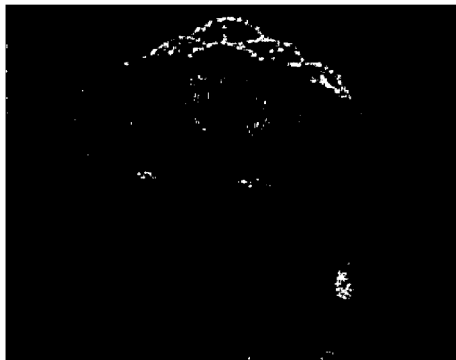


Рис. 2. Результат фильтрации функции A



Рис. 3. Результат фильтрации после подготовительного этапа

4. Этап конечнопараметрической фильтрации. На первом этапе была выделена подобласть H исходного изображения, имеющая красный оттенок и включающая в себя области возможного отражения вспышки. На втором этапе рассмотрим функцию “красноты” как линейную комбинацию функций исходного изображения [3]: $\mathcal{R}(x, y) = 0.6R(x, y) - 0.5G(x, y) - 0.1B(x, y)$.

Воспользуемся априорным предположением о том, что области красных глаз имеют вид круга. Будем также предполагать, что в пределах круга функция красноты имеет вид купола. Найдем области, удовлетворяющие этим предположениям в пределах области H .

Математически наши предположения записываются следующим образом: существует такой круг радиуса R с центром в точке (x_0, y_0) , содержащий красный глаз, в пределах которого выполняются соотношения

$$U(x, y) = C_1 G_1(x - x_0, y - y_0) + C_2 G_2(x - x_0, y - y_0) + C_3 G_3(x - x_0, y - y_0),$$

$$G_1(x, y) = \exp(-s_1^2(x^2 + y^2)), \quad G_2(x, y) = \exp(-s_2^2(x^4 + y^4)), \quad G_3(x, y) = \begin{cases} \text{const}, & x^2 + y^2 < R^2, \\ 0, & x^2 + y^2 \geq R^2, \end{cases}$$

где R больше предполагаемого радиуса красного глаза.

В рассматриваемом параметрическом виде функция $U(x, y)$ зависит от восьми параметров. Параметры C_1, C_2, C_3, x_0, y_0 неизвестны, а параметры s_1, s_2, R выбираем исходя из априорной информации о примерном радиусе r красных глаз. Параметры s_1 и s_2 выбираются так, чтобы средние радиусы гауссовых функций $G_1(x, y), G_2(x, y)$ отличались между собой и были близки к возможным значениям радиуса красных глаз. Использование двух гауссовых слагаемых позволяет лучше учитывать возможные вариации в изображениях красных глаз (рис. 4, 5).

Таким образом, ставится задача поиска наилучшего приближения функции $\mathcal{R}(x, y)$ функцией $U(x, y)$ с заданными параметрами s_1, s_2, R и произвольными параметрами C_1, C_2, C_3, x_0, y_0 .

Поставленная задача эквивалентна задаче минимизации функционала

$$D(x_0, y_0) = \|C_1 G_1(x - x_0, y - y_0) + C_2 G_2(x - x_0, y - y_0) + C_3 G_3(x - x_0, y - y_0) - \mathcal{R}(x, y)\|^2$$

по линейным параметрам C_1, C_2, C_3 и нелинейным параметрам x_0, y_0 , где $\|\cdot\|$ — L_2 -норма, взятая в пределах круга радиуса R с центром (x_0, y_0) .

Минимизация по параметрам x_0, y_0 осуществляется перебором по сетке с шагом h в области H . Минимизация по линейным параметрам выписывается в явном виде [4].

Таким образом, для произвольной точки (x_0, y_0) выбранной сетки области H находим линейные параметры, минимизирующие функционал $D(x_0, y_0)$.

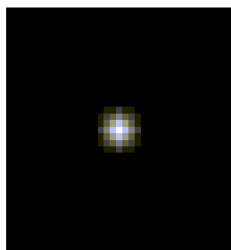


Рис. 4. Функция $G_1(x, y)$

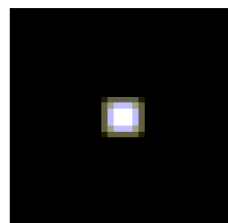


Рис. 5. Функция $G_2(x, y)$

Решающее правило для данного этапа состоит из двух частей.

1) Выбирается подобласть $H_1 \subset H$, где $C_1(x_0, y_0) + C_2(x_0, y_0) > L_1$ (L_1 — некоторый положительный пороговый параметр), т.е. хотя бы один коэффициент при гауссовых функциях должен быть достаточно большим.

2) Для оставшихся элементов подобласти H_1 вычисляем относительную невязку

$$N(x_0, y_0) = \frac{D(x_0, y_0)}{C_1(x_0, y_0) + C_2(x_0, y_0)}.$$

Отбрасываются те элементы, у которых $N(x_0, y_0)$ больше заданного порога.

В результате фильтрации определяются возможные области H_2 расположения красных глаз (рис. 6).

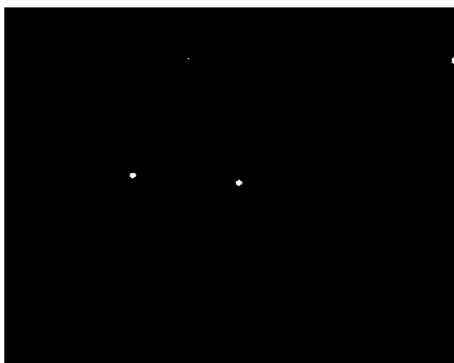


Рис. 6. Результат фильтрации после основного этапа фильтрации



Рис. 7. Функция $g(x, y)$

5. Этап контекстного поиска. Для выделения красных глаз в оставшейся после первых двух этапов распознавания области H_2 представляется целесообразным применять некоторые процедуры контекстного поиска. Контекстный этап заключается в следующем. В соответствии с априорной информацией за пределами круга с красным глазом расположены белки глаз. Поэтому следует определить, существуют ли белки вокруг красного зрачка. Для этого формируем в пределах исходного изображения подобласть H_3 оттенков белого цвета. Решающее правило $g(x, y)$ (рис. 7), которое применяется на этом этапе, выглядит так: $g(x, y) = 1$ для тех точек (x, y) , для которых

$$|R(x, y) - G(x, y)| < d, \quad |G(x, y) - B(x, y)| < d, \quad |R(x, y) - B(x, y)| < d, \quad R(x, y) > r_1.$$

Далее мы сворачиваем результат первых двух этапов распознавания с функцией, представляющей собой кольцо, внутренний радиус которого равен $0.5R$, а внешний равен $0.9R$ соответственно (рис. 8). Результат умножаем на $g(x, y)$. На этом этап контекстного поиска заканчивается (рис. 9).

Для окончательного выделения интересующих нас областей еще раз делаем свертку с кругом радиуса $0.9R$ и берем пересечение с областью H_2 . Выделенные области будем считать результатом распознавания красных глаз (рис. 10).

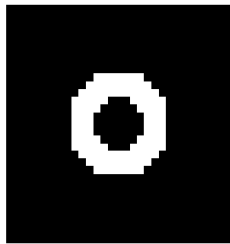


Рис. 8. Ядро используемое для свертки

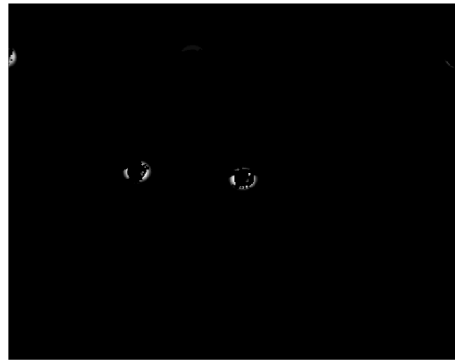


Рис. 9. Результат контекстного поиска

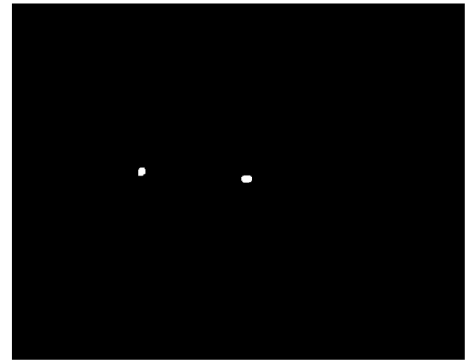


Рис. 10. Результат распознавания

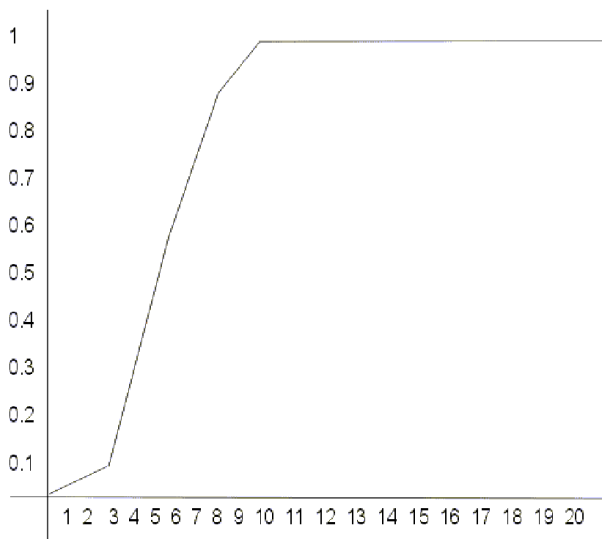


Рис. 11

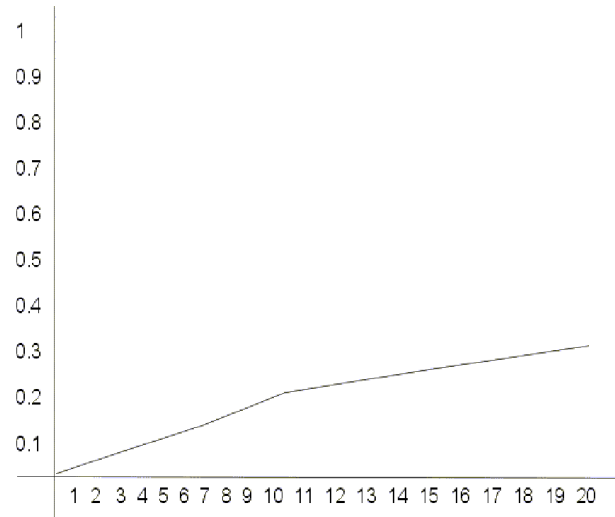


Рис. 12

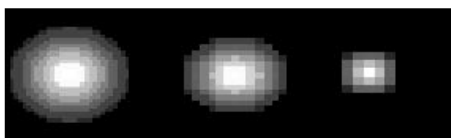


Рис. 13. Модели, применяемые на этапе конечнопараметрической фильтрации

6. Корректировка. Корректировку выделенных областей красных глаз можно проводить различными способами. В настоящей работе корректировка проводилась согласно [1]. Она показала свою достаточно высокую эффективность.

7. Апробация алгоритма. Данный алгоритм реализован на MatLab v 6.1 и протестирован на 200 фотографиях. Фотографии были сняты различными фотоаппаратами, в разное время, в разных местах и при разном освещении. Оцифровывались изображения как при помощи различных сканеров, так и различными цифровыми камерами. Средняя продолжительность обработки одной фотографии размером 512×512 пикселей — 30 секунд на компьютере с процессором Pentium4 2400 мегагерц и оперативной памятью 512 мегабайт.

В данной версии алгоритма параметры подбирались так, чтобы распознать как можно большее число зрачков глаз. Количество мусора (лже-зрачков) не имело большого значения.

На рис. 11 изображено отношение количества найденных с помощью предложенного алгоритма красных зрачков к общему количеству красных зрачков для данного радиуса зрачка (в пикселах).

На рис. 12 для каждого радиуса зрачка изображено отношение числа фотографий, в которых после распознавания данным алгоритмом отсутствует мусор, к общему числу фотографий.



Рис. 14



Рис. 15. Пример фотографии, на которой распознается большое количество мусора

8. Основные выводы. Алгоритм работает лучше при больших размерах зрачка глаза. На рис. 13 представлены модели красного глаза, применявшиеся для распознавания и отличающиеся друг от друга одним параметром — размером зрачка.

На качественных фотографиях либо с крупными лицами, либо больших размеров алгоритм работает со 100 % распознаванием и меньшим количеством мусора.

Одной из отрицательных черт алгоритма является сильная зависимость от априорно заданного радиуса зрачков. Есть проблемы при распознавании красных глаз, например, на фотографиях людей, носящих очки, что объясняется появлением бликов от вспышки. На пестрых поверхностях также распознается много мусора (рис. 14, 15).

К положительным свойствам алгоритма можно отнести то, что данный алгоритм не зависит от положения лица на фотографии, его наклона и формы глаз. Время работы алгоритма не увеличивается, если на фотографии имеется много лиц с красными глазами.

СПИСОК ЛИТЕРАТУРЫ

1. *Schildkraut J.S., Gray R.T.* A fully automatic red-eye detection and correction algorithm // The 9th International Conference on Image Processing. September 23–26. I. Rochester, 2002. 801–803.
2. *Верлань А.Ф., Сизиков В.С.* Интегральные уравнения: методы, алгоритмы, программы. Киев: Наукова Думка, 1986.
3. *Gaubatz M., Ulichney R.* Automatic red-eye detection and correction // The 9th International Conference on Image Processing. September 23–26. II. Rochester, 2002. 103–107.
4. *Васильев Ф.П.* Численные методы решения экстремальных задач. М.: Наука, 1980.

Поступила в редакцию
21.09.2005