

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Южный федеральный университет»

Институт математики, механики и компьютерных наук им.
И.И.Воровича

Кафедра алгебры и дискретной математики

Ковалев Никита Евгеньевич

**ВОССТАНОВЛЕНИЕ ЦИФРОВЫХ
ИЗОБРАЖЕНИЙ
С ВЫРОЖДЕННЫМ СМАЗОМ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА БАКАЛАВРА
по направлению 01.03.02 — Прикладная математика и информатика

**Научный руководитель —
доц., к.ф.-м.н. Козак Анатолий Всеволодович**

Ростов-на-Дону — 2018

Содержание

| | |
|--|----------|
| Введение | 2 |
| 1 Вспомогательные понятия | 4 |
| 2 Метод предобработки в случае большого числа смаза | 6 |
| 3 Метод минимизации погрешности | 9 |
| 3.1 Описание проблемы | 9 |
| 3.2 Возможные пути решения | 10 |
| 3.3 Особенности реализации на практике | 11 |
| 3.3.1 Поправка по умножению | 11 |
| 3.3.2 Поправка по сложению | 13 |
| 3.3.3 Сравнение поправок на практике | 15 |

Введение

С появлением фотокамер человечество получило инструмент, позволяющий сохранять информацию о любых видимых объектах окружающей среды. Но снимки, получаемые с помощью фотокамеры, не всегда хорошего качества по многим причинам. Безусловно, фотокамеры совершенствуются людьми, и сейчас они способны устранять многие искажения еще при обработке и сохранении снимка, однако, особенно при тяжелых условиях съемки, зачастую фотографии не отражают столько информации, сколько необходимо. С развитием техники появились инструменты для работы с искаженными фото, а ЭВМ обрели вычислительную мощность, достаточную для обработки фото высокого разрешения за достаточно быстрое время. Также из всех возможных искажений выделили некоторые классы явлений, с которыми есть возможность работать в рамках определенных методов. Одним из таких негативных эффектов является смаз.

В данной работе рассматривается явление горизонтального циклического смаза. Само явление возникает при съемке местности с помощью вращающейся камеры при создании панорамных снимков. При возникновении этого явления смазанное изображение содержит большое количество искаженной (потерянной) информации, соответственно, появляется необходимость восстановить эти данные. Поскольку задача актуальна для таких областей, как военная разведка, картография, геодезия, моделирование объектов местности и аналогичных, то отснять материал заново зачастую невозможно. В силу этой специфики задача восстановления искаженных снимков приобретает наиважнейший характер.

В реальной практике искажение происходит не только из-за смаза, но также из-за способа сохранения снимков. При обработке снимка фотокамера оцифровывает изображение с определенной точностью (в формате 8, 12

или 16 бит на пиксел), то есть переводит информацию о кадре в определенный интервал и округляет до ближайшего целого числа. Нетрудно понять, что уже на этом этапе идет потеря информации о всех объектах на изображении, и, чем больше бит отводится на пиксел, тем меньше информации теряется на первом этапе.

По причине того, что данная задача требует выполнения перемножений матриц большого размера, а также работы с матрицами комплексных чисел, необходимо выбрать наиболее эффективный язык для реализации методов. Таковым, определенно, является MATLAB (сокр. от MATrix LABoratory), и, так как лицензированная MATLAB-среда имеется в дисплейных классах Института математики, механики и компьютерных наук имени И.И. Воровича, то я остановился именно на ней. Это позволило добиться действительно быстрой работы описанных в работе методов.

Стоит отметить, что на протяжении всего учебного года мы вместе с другими студентами периодически обсуждали промежуточные результаты наших работ, а также различные тенденции в ведущих областях исследований по нашим темам. В процессе таких дискуссий рождалось множество идей для продолжения наших исследований, в том числе таким образом был придуман один из важнейших аспектов второго метода из описанных в данной работе.

1 Вспомогательные понятия

Введем некоторые обозначения и базовые понятия, которые будут использоваться в дальнейшем.

Определение 1.1. *Матрицей горизонтального циклического смаза* (далее - *матрицей смазы*) на k пикселей называется матрица $C(n, k) \in M_n$ вида

$$C(n, k) = \frac{1}{k} \begin{pmatrix} \underbrace{1 \dots 1}_k & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 1 & 1 & \dots & 0 \\ \dots & \ddots & \dots & \ddots & \dots \\ 1 & \dots & 0 & 1 & 1 & \dots & 1 \\ \dots & \ddots & \dots & \ddots & \dots \\ 1 & \dots & 1 & 0 & 0 & \dots & 1 \\ \underbrace{\quad}_{k-1} \end{pmatrix}$$

В дальнейшем будем отождествлять понятие матрицы смазы и обозначения $C(n, k)$ и просто C (если параметры n и k ясны из контекста).

Определение 1.2. *Горизонтальным циклическим смазом* (далее - *смазом*) будем называть умножение изображения на соответствующую матрицу смазы справа.

Утверждение 1.1 (об обратности матрицы смазы). Матрица смазы обратима тогда и только тогда, когда $\text{НОД}(n, k) = 1$.

Следствием 1.1 является, среди прочего, большее количество изображений с вырожденным смазом, нежели с невырожденным, что лишний раз подтверждает актуальность поставленной задачи.

Пример. Приведем пример изображений и их смазанные варианты.



Рис. 1: Смаз при $k = 248$



Рис. 2: Исходное изображение

Это изображение хорошо для исследования реальных ситуаций, так как оно является панорамным, то есть левая часть фото продолжает правую. Тем не менее, на самом деле, для качественного изучения результатов достаточно взять любое изображение.



Рис. 3: Смаз при $k = 248$



Рис. 4: Исходное изображение

2 Метод предобработки в случае большого числа смаза

В первом параграфе пойдет речь о методе, который эффективен только для восстановления 16-битных изображений, поскольку данный подход использует обычное восстановление для невырожденного случая, а оно эффективно только для 16-битных изображений. В силу этих ограничений метод интересен не столько своей практической пригодностью, сколько подходом, который, возможно, применим в комбинации с лучшим методом восстановления.

Подход заключается в следующем: в случае вырожденного смаза имеем $d = \text{НОД}(n, k) > 1$, где k - число смаза, n - ширина изображения в пикселях. Тогда поделим исходное изображение I на d подизображений, каждое из которых содержит i -й столбец и далее каждый столбец, на d правее предыдущего выбранного. В итоге получим последовательность $\{I_j\}_{j=1}^d$, каждое изображение которой содержит $\frac{n}{d}$ пикселей в ширину. Однако, каждое из этих изображений будет содержать смаз уже не на k , а на $\frac{k}{d}$, и, так как $d = \text{НОД}(n, k)$, то $\text{НОД}(\frac{n}{d}, \frac{k}{d}) = 1$, следовательно, смаз невырожденный. Восстанавливая I_j и объединяя их по тому же принципу, по какому происходило и разделение, мы получим изображение, содержащее смаз на d пикселей. Объясняется это тем, что каждая из восстановленных картинок предоставляет $\frac{1}{d}$ информации от необходимой, но информация разных частей не является уникальной - все они несут примерно одну и ту же информацию. Это подтверждается экспериментально на реальных примерах.

Пример. Первый пример демонстрирует случай работы с 16-битными входными данными. Здесь НОД двух параметров больше единицы, но незначительно (в данном случае, 16), и в таком случае восстановление происходит достаточно качественно и, возможно, даже не требует дальнейшего восстановления (в зависимости от задачи).



Рис. 5: Смаз при $k = 592$



Рис. 6: Улучшенное фото

Пример. Второй пример показывает, что метод действительно работает плохо для 8-битных изображений. Тем не менее, хотя на данный момент преимущественно именно такой формат используется в фотокамерах, уже существует техника, которая сохраняет фото в 12-битном формате, и скоро получится внедрить и 16-битный формат, что позволит использовать данный метод в полном объеме, поскольку он быстрее любого другого за счет деления проблемы на подзадачи меньшего размера. Также стоит отметить, что даже такой результат может оказаться достаточным, так как границы объектов вырисовываются достаточно четко.

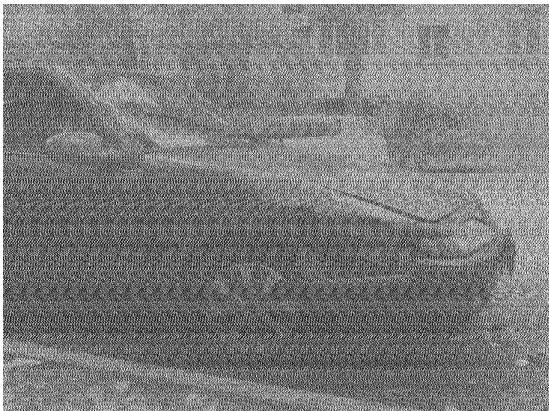


Рис. 7: Улучшение при $d = 2$,
 $k = 242$



Рис. 8: Улучшение при $d = 8$,
 $k = 248$

Также этот пример иллюстрирует характер поведения результата в зависимости от параметра - чем больше d , тем более смазан оказывается результат, но с другой стороны, меньшее d не позволяет нивелировать недостатки 8-битного невырожденного восстановления из-за большого числа смаза для вырезанных частей исходного изображения. Из этого следует ожидаемый вывод, что данный метод работает лучше при больших d .

3 Метод минимизации погрешности

3.1 Описание проблемы

Больший интерес представляют методы, которые позволяют избавиться от смаза полностью. Учитывая специфику уравнения смаза $XC = S$, где X - исходное изображение, C - матрица смаза и S - смазанное изображение, получим, что для идеального математически восстановления необходимо найти обратную к C матрицу и домножить на нее справа. Конечно, на практике это либо дает плохой результат (случай невырожденного смаза), либо невозможно (случай вырожденного смаза) в силу особенностей задачи, а именно, матрицы смаза.

Пример. Данный пример иллюстрирует, что необходимость в улучшении обычного восстановления для случая 8-битных входных данных действительно имеется даже для невырожденного случая. Здесь число смаза равно 491, то есть является простым числом, следовательно, смаз невырожденный.



Рис. 9: Восстановление
16-битного входа

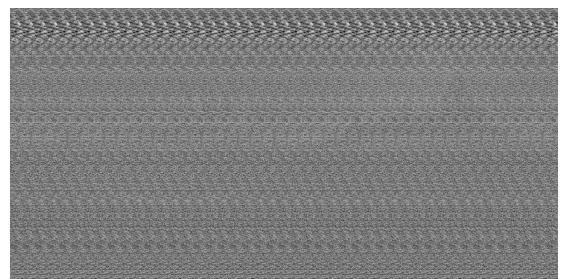


Рис. 10: Восстановление
8-битного входа

Первым препятствием является тот факт, что матрица смаза необратима при $\text{НОД}(n, k) = 1$, где k - число смаза. Это означает, что в этом случае мы можем получить только приближенно-обратную матрицу, то есть такую

матрицу \tilde{C} , что $C\tilde{C} \approx E$. Вторым ограничением является формат данных, с которым приходится работать при обработке изображений. В момент получения кадра изображение оцифровывается — непрерывная картина из окружающего мира приближается сеткой из пикселей, а значение каждого пикселя (интенсивности цвета в точке) в свою очередь округляется до ближайшего целого. Таким образом, уже на данном этапе происходит потеря огромного количества информации за счет погрешности округления.

Имея в виду вышесказанное, необходимо сконструировать матрицу так, что при умножении на матрицу смаза справа она дает приближение единичной матрицы — это даст сохранение результата, максимально приближенного к исходному теоретически, а также учесть уже имеющуюся погрешность округления и минимизировать погрешность вычислений.

3.2 Возможные пути решения

Известно, что матрица смаза представима в виде $C = FDF^{-1}$, где D - диагональная матрица, причем на диагонали - собственные числа матрицы C , а F - дискретное преобразование Фурье соответствующего размера. Понятно, что в случае, когда матрица полного ранга, ее обратная находится обращением всех чисел на диагонали. Будем действовать по этому принципу, но с некоторыми поправками:

- 1) Необратимость будем устранять заменой нулевых собственных чисел на некоторые числа с фиксированным модулем (модуль, так как собственные числа являются комплексными);
- 2) Погрешность вычислений минимизируем тем же способом - к ненулевым собственным числам, которые сильно малы по модулю, прибавим случайное комплексное число с тем же фиксированным модулем;
- 3) Кроме того, оставшиеся достаточно малые собственные числа домножим так, чтобы по модулю они были равны фиксированному числу.

После соответствующих операций мы получим вместо матрицы D^{-1} , которая, вообще говоря, не всегда существует, некоторую матрицу \tilde{D} такую, что $D\tilde{D} \approx E$ и $\tilde{C} = F\tilde{D}F^{-1}$. Рассмотрим, каким образом эти два преобразования повлияют на результат в теории и на практике. Эти результаты будут отличаться не только потому, что в реальной проблеме существуют погрешности, но так же и в силу особенностей порядка операций.

3.3 Особенности реализации на практике

В теории исходная матрица X восстанавливается следующим порядком действий:

$$SF\tilde{D}F^{-1} = (XFDF^{-1})F\tilde{D}F^{-1} = XFD\tilde{D}F^{-1},$$

то есть в идеализированной модели результат будет зависеть от того, насколько полученные числа на диагонали будут близки к числам, обратным к собственным. Также отдельно стоит отметить нулевые собственные числа - на что бы их не умножили, они останутся нулями, и соответствующие подобранные значения в матрице \tilde{D} никакой роли не сыграют. На практике все выглядит немного иначе, поскольку матрица S не расщепляется на составные части. Таким образом, влияние чисел, соответствующих нулевым собственным значениям, оказывается вполне реальным, поскольку итоговое приближение исходной матрицы X будет выглядеть не иначе как $S(F\tilde{D}F^{-1}) = S\tilde{C}$. Соответственно, необходимо учитывать и этот фактор.

3.3.1 Поправка по умножению

Рассмотрим случай, когда сильно малых собственных значений не наблюдается. Тогда для каждого из них будет существовать некая поправка по умножению $\delta_i, i = \overline{0, n-1}$, а поправка по сложению не влияет на конечный результат. Это позволяет нам рассматривать данный случай так же,

как в теории, где внутренние преобразования Фурье взаимоуничтожаются. Обозначим $\tilde{E} = D\tilde{D}$, тогда:

$$X = S\tilde{C} = XFEF^{-1}, \quad \tilde{E} = \begin{pmatrix} \frac{d_0}{d_0\delta_0} & 0 & \dots & 0 \\ 0 & \frac{d_1}{d_1\delta_1} & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & \frac{d_{n-1}}{d_{n-1}\delta_{n-1}} \end{pmatrix} = \begin{pmatrix} \delta_0^{-1} & 0 & \dots & 0 \\ 0 & \delta_1^{-1} & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & \delta_{n-1}^{-1} \end{pmatrix},$$

где d - вектор собственных чисел матрицы C .

Обозначим получившуюся матрицу $R = F\tilde{E}F^{-1}$. Тогда $R_{ij} = \frac{1}{n} \sum_{k=0}^{n-1} \delta_k^{-1} \xi^{(j-i)k}$, где ξ есть первый корень из единицы степени n ($e^{\frac{2\pi i}{n}}$). Получим, что диагональные элементы R равны между собой и равняются среднему величин обратных к поправкам по умножению, а элементы вне диагонали есть взвешенные средние n различных корней из единицы. Важным замечением является тот факт, что эти корни различны для элементов на разных диагоналях, поскольку при удалении от главной диагонали угол между соседними корнями ($\xi^{(j-i)k}$ и $\xi^{(j-i)(k+1)}$) увеличивается, и, соответственно, различные симметричные свойства теряются. Например, на первой диагонали, которая находится выше главной, "базовый" корень есть не что иное, как ξ , и, чтобы минимизировать элементы на этой диагонали, необходимо, чтобы вторая половина вектора поправок по умножению в точности повторяла первую ($\delta_0 = \delta_{\lfloor \frac{n-1}{2} \rfloor + 1}, \delta_1 = \delta_{\lfloor \frac{n-1}{2} \rfloor + 2}$ и т.д.), тогда как уже на второй диагонали условие для минимизации меняется на $\delta_i = -\delta_{\lfloor \frac{n-1}{2} \rfloor + i + 1}$. Таким образом, объединить условия для разных диагоналей не представляется возможным, даже если использовать поправки только для малого количества собственных значений. Это означает, что поправки по умножению, несмотря на то, что имеют детерминистскую природу, не позволяют достичь хороших результатов (кроме, конечно, случая, когда $\delta_i = \delta_j \forall i, j \in \overline{0, n-1}$, о нем пойдет речь в следующем пункте).

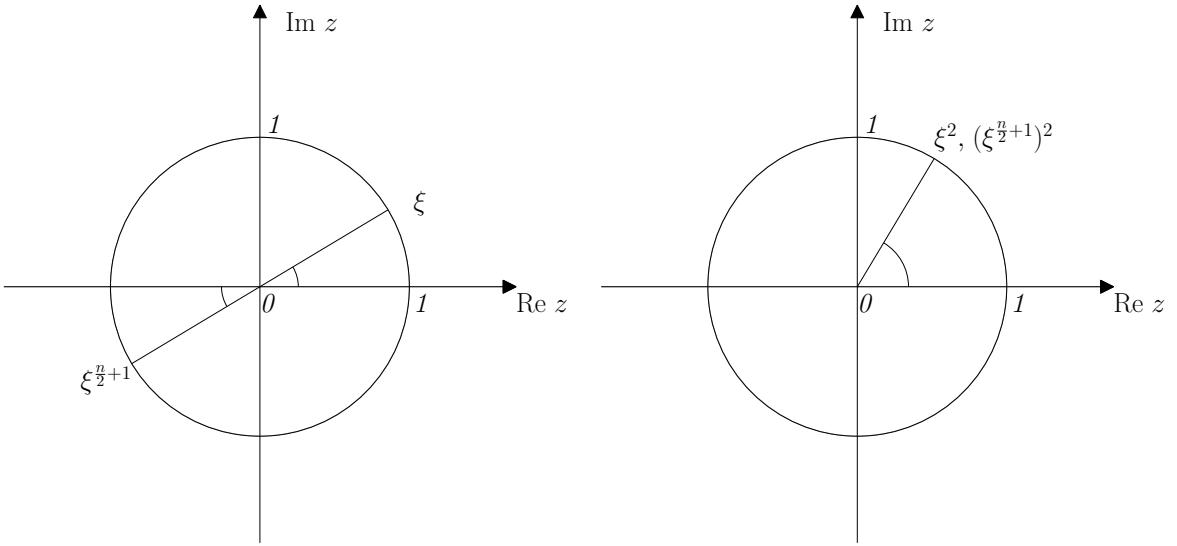


Рис. 11: Собственные числа для первой диагонали после главной

Рис. 12: Собственные числа для второй диагонали после главной

3.3.2 Поправка по сложению

С поправкой по сложению все иначе. В теоретическом представлении задачи она позволяет лишь только осуществить псевдообращение матрицы смазы, но сами числа, которые мы прибавим, никоим образом не повлияют на результат. Следовательно, необходимо рассматривать матрицу

$\tilde{D} = \text{diag}\left(\frac{1}{d_0 + \sigma_0}, \frac{1}{d_1 + \sigma_1}, \dots, \frac{1}{d_{n-1} + \sigma_{n-1}}\right)$, где σ - вектор значений поправки по сложению. Тогда аналогично п.3.3.1, положив $\gamma_i = d_i + \sigma_i$, $\tilde{R} = F\tilde{D}F^{-1}$, получим, что $\tilde{R}_{ij} = \frac{1}{n} \sum_{k=0}^{n-1} \gamma_k^{-1} \xi^{(j-i)k}$. Из уже приведенных рассуждений известно, что подбор для каждой отдельной диагонали не имеет смысла, соответственно, необходимо использовать новый подход. Так как для поправки по сложению мы отбираем числа с модулем, близким к или равным нулю, то соответствующие "обратные" собственные числа оказываются очень велики по модулю. Конечно, этого следует избегать, поскольку и так имеющаяся погрешность будет дополнительно усиlena погрешностью

вычислений. Поэтому, учитывая специфику поведения R_{ij} , перспективной выглядит идея установить для чисел d_i , которые попадают под условие на поправку по сложению, соответствующее σ_i сделать фиксированным числом для любого такого d_i . Тогда, так как для каждого собственного числа на первой диагонали после главной имеется пара, то и на последующих диагоналях найдутся пары для новых чисел, которые взаимоуничтожаются при равных коэффициентах перед ними. Тем не менее, поскольку поправка применяется не ко всем, а только к малым по модулю числам, некое подобие структуры, присущей исходной матрице, остается, и за счет этого восстановление остается возможным с теоретической точки зрения (теряются не все свойства матрицы).

Действительно, оказывается, именно такой подход, единственный математически оправданный, является удачным и позволяет найти баланс между теоретической точностью и практической минимизацией погрешности. Примеры так же показывают, что для достижения этих результатов подойдет не только некая константа для всего вектора, но так же случайные комплексные числа с фиксированным модулем, причем постоянство модуля является определяющим свойством системы.

3.3.3 Сравнение поправок на практике

Пример. Данное восстановленное фото показывает, что изменение части "обратных" собственных чисел при помощи поправки по умножению дает худший результат, чем восстановление без поправок, тем самым подтверждая, что математическая (теоретическая) точность теряется.



Рис. 13: Восстановление с поправкой по умножению 16-битного фото

Пример. Этот пример иллюстрирует восстановление на примере 8-битного изображения. Хорошо видно, что математическая точность пострадала ввиду изменения структуры матрицы путем изменения собственных значений, но, тем не менее, большая часть различимых деталей видна, что является следствием минимизации погрешности.



Рис. 14: Восстановление с применением поправки по сложению

Необходимо отметить, что в случае использования поправки по сложению оба типа изображений дают практически или совсем одинаковые результаты. Это говорит о том, что погрешность вычислений действительно удалось устраниТЬ или минимизировать, что лишний раз подтверждает приведенные выше рассуждения.