

Распознавание образов

Формализм структурных преобразований изображений и методы их оптимизации

М.С. МЕХОВА^{I,III,IV}, П.В. БЕЗМАТЕРНЫХ^{I,II}

^I ООО «Смарт Энджинс Сервис», г. Москва, Россия

^{II} Федеральный исследовательский центр «Информатика и управление»
Российской академии наук, г. Москва, Россия

^{III} Национальный исследовательский технологический университет «МИСИС»,
г. Москва, Россия

^{IV} Институт проблем передачи информации им. А.А. Харкевича Российской
академии наук, г. Москва, Россия

Аннотация. Низкоуровневые методы обработки и преобразований изображений имеют фундаментальное значение для множества прикладных задач в совершенно различных областях. Частные вопросы их оптимизации рассмотрены в многочисленных технических и научных публикациях, однако общий подход к ним в литературе представлен явно неполно. В статье представлен новый формализм – структурное преобразование изображений, который нацелен на устранение данного пробела. Он позволяет формализовать ряд методов из области обработки изображений, оценить сложность их вычисления в количестве операций и выявить общие подходы к их ускорению. Приведено три таких метода оптимизации, с помощью которых структурные преобразования изображений конвертируются в быстрые структурные преобразования, обладающие меньшей сложностью по сравнению с наивным независимым вычислением отдельно взятых элементов исходных преобразований. Рассмотрены применения указанных методов для задач автоматической обработки изображений документов, чтения штриховых кодов и вычислительной томографии.

Ключевые слова: обработка изображений, комбинаторная оптимизация, автоматическая обработка изображений документа.

DOI: 10.14357/20790279250409 **EDN:** YUDLNN

Введение

Различные преобразования цифровых изображений, наряду с методами их анализа, являются ключевыми компонентами области цифровой обработки изображений (далее – ЦОИ). Преобразование изображений определяется как трансформация одного изображения в другое, обладающее некоторыми желаемыми свойствами. Под анализом изображений понимается класс задач, направленных на получение качественных или количественных суждений о характеристиках входного изображения или же его содержания. Минималь-

ным структурным элементом любого цифрового изображения является пиксель (англ. picture cell). Все высокоуровневые объекты, представленные на изображении, описываются тем или иным набором пикселей, сгруппированных по некоторым признакам.

Обе указанные компоненты базируются на использовании некоторого множества низкоуровневых методов обработки изображений, или, как их еще называют, примитивов. К ним относятся, например, методы математической морфологии, локальные усредняющие фильтры, интегральное

преобразование Хафа и знаменитое преобразование Фурье. Эти примитивы изложены практически в любом пособии по ЦОИ, включая классические учебники [1,2].

Неудивительно, что их оптимизации посвящено множество трудов. При этом необходимо отличать один способ оптимизации от другого. Повысить быстродействие методов можно, например, за счет использования таких аппаратных возможностей, как многопоточная обработка (англ. multithreading), применение специализированных процессорных SIMD-инструкций (англ. single instruction, multiple data), которые отлично себя зарекомендовали именно в вопросах обработки изображений [3,4] или же специализированных графических ускорителей.

В данной работе нас интересует группа методов, которые условно можно отнести к дисциплине комбинаторной оптимизации. Так, уже в 1984 году в замечательной книге под редакцией Т.С. Хуанга [5], были тщательно проанализированы и описаны такие методы, как быстрое транспонирование двумерных массивов, алгоритм Винограда для дискретного преобразования Фурье, алгоритм быстрой медианной фильтрации. При этом оптимизацией данных методов продолжают заниматься и сейчас. Так, например, в статье [6] предложены очередные улучшения для алгоритма медианной фильтрации. Изыскания в области оптимизации вычисления преобразования Хафа послужили фундаментом для формирования целой научной школы [7], которая активно внедряет данный примитив в различных прикладных задачах.

Несмотря на это, общий вопрос оптимизации классов алгоритмов выглядит недостаточно освещенным. Поэтому в данной работе вводится новый формализм структурных преобразований изображений (далее – СПИ). *Структурным преобразованием изображений* будем называть применение выбранной ассоциативной операции (например, сложение) к выделенному набору пикселей. Через этот формализм оказывается удобно выразить примитивы, перечисленные выше. Таким образом, СПИ становятся важной общей низкоуровневой частью обработки изображений для многих прикладных областей. Для каждого структурного преобразования можно определить его сложность: количество применений операции, требуемое для его вычисления. *Быстрым структурным преобразованием изображений* (далее – БСПИ) будем называть структурное преобразование, обладающее уменьшенной сложностью относительно алгоритма, независимо применяющего заданную операцию для каждого подмножества пикселей исходного СПИ.

Далее в Разделе 1 приведена формализация ключевых понятий, в Разделе 2 показаны различные примеры структурных преобразований. В Разделе 3 перечислены общие методы их оптимизации. В Разделе 4 рассмотрено их применение в системах обработки изображений документов, чтения штриховых кодов и компьютерной томографии.

1. Аппарат структурных преобразований изображений

Назовем двумерным растром R прямое произведение $\mathbb{Z} \times \mathbb{Z}$, а $R^{M \times N} = \mathbb{Z}_{M-1} \times \mathbb{Z}_{N-1} \subset R$ – ограниченным растром с размерами $M \times N$. *Цифровое изображение* – это дискретная функция $I^{M \times N}: R^{M \times N} \rightarrow V$, заданная на ограниченном растре $R^{M \times N}$. Область значений V определяется прямым произведением элементов кортежа $\langle \mathbb{Z}_{c_1}, \mathbb{Z}_{c_2}, \dots, \mathbb{Z}_{c_k} \rangle$, где \mathbb{Z}_{c_i} – каналы изображения, а c_i – количество цветов канала. *Пиксель* – упорядоченная пара $p = \langle r, v \rangle$, состоящая из позиции элемента раstra $r \in R^{M \times N}$ и соответствующего ему значения пикселя $v \in V$. Для обозначения значения яркости пикселя изображения $I^{M \times N}$ в позиции раstra $\langle r, c \rangle$ используется краткая запись $I(r, c)$. Таким образом, $p = \langle \langle x, y \rangle, v \rangle$ и $I(x, y) = v$.

Рассмотрим произвольный растр $R^{M \times N}$ и множество определенных на нем цифровых изображений $\mathbb{I} = \{I: R^{M \times N} \rightarrow V\}$. *Паттерном* π^m будем называть выделенное упорядоченное множество позиций элементов раstra $\langle r_1, r_2, \dots, r_m \rangle \subset R^{M \times N}$, а величину m – его размером. Булеан паттернов на растре $R^{M \times N}$ обозначим $\mathcal{P}(R)$. Для определенности зафиксируем порядок перечисления множества позиций элементов раstra как *лексикографическое упорядочение* по его координатам. На рис. 1,а приведен пример паттерна $\pi^4 = \langle \langle 0,0 \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 3,2 \rangle \rangle$, заданного на растре $R^{5 \times 5}$. Как видно из рисунка, паттерн не обязательно должен быть связным.

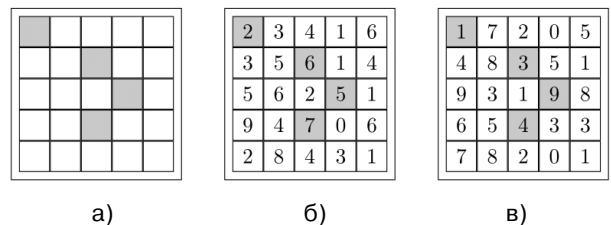


Рис. 1. Иллюстрация паттерна: а – $\pi^4 = \langle \langle 0,0 \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 3,2 \rangle \rangle$ на $R^{5 \times 5}$; б – его приложение $\langle 2,6,5,7 \rangle$ к изображению-примеру $I: R^{5 \times 5} \rightarrow \mathbb{Z}_{10}$; в – его приложение $\langle 1,3,9,4 \rangle$ к изображению-примеру $I: R^{5 \times 5} \rightarrow \mathbb{Z}_{10}$. Серым цветом выделены элементы паттерна

Отметим, что для паттерна значения пикселей, соответствующих позициям раstra, не определены вплоть до момента указания конкретного изображения. Упорядоченное множество значений пикселей $\pi_I^m = \langle v_1, v_2, \dots, v_m \rangle$ будем называть *приложением паттерна* π^m к изображению I . Два различных приложения одного паттерна π_I^4 к изображению $I: R^{5 \times 5} \rightarrow \mathbb{Z}_{10}$ проиллюстрированы рис. 1,б,в. Таким образом, паттерн определяет «геометрическую составляющую», но есть еще и яркостная, действующая элементы раstra.

Пусть на множестве значений V определена алгебраическая ассоциативная бинарная операция \circ с нейтральным элементом e . Тогда алгебраическая структура $S = (V, \circ)$ является моноидом [8]. В данной работе интерес представляют следующие два специализированных вида моноидов: коммутативные моноиды; группы (моноиды, где каждый элемент имеет обратный). Рассмотрим операцию $\mathcal{R}(\pi^m, \circ) = v_1 \circ v_2 \circ \dots \circ v_m$. Вычисление данной операции будем называть *редукцией паттерна*. Отметим, что эта операция определяет только шаблон вычислений, конкретное значение может быть получено только при приложении паттерна к изображению $I \in \mathbb{I}$. В таком случае будем обозначать редукцию символом $\mathcal{R}(\pi_I^m, \circ)$. На рис. 1,б показаны значения всех пикселей данного изображения. Для него выполняется $\pi_I^4 = \langle 2, 6, 5, 7 \rangle$, $\mathcal{R}(\pi_I^4, +) = 20$, $\mathcal{R}(\pi_I^4, \max) = 7$.

Структурным преобразованием \mathcal{S} изображений $I: R^{M \times N} \rightarrow V$ с алгебраической структурой $S = (V, \circ)$ на конечном множестве паттернов $\Pi_n = \{\pi_1^\alpha, \pi_2^\beta, \dots, \pi_n^\omega\}$, $\pi_i \in \mathcal{P}(R)$ будем называть задачу вычисления n совместных редукций по элементам множества Π_n :

$$\mathcal{S}_\Pi = \begin{cases} \mathcal{R}(\pi_1^\alpha, \circ) = v_1^1 \circ v_2^1 \circ \dots \circ v_\alpha^1 \\ \mathcal{R}(\pi_2^\beta, \circ) = v_1^2 \circ v_2^2 \circ \dots \circ v_\beta^2 \\ \dots \\ \mathcal{R}(\pi_n^\omega, \circ) = v_1^n \circ v_2^n \circ \dots \circ v_\omega^n \end{cases} \quad (1)$$

Обозначим через $\mathcal{A}(\mathcal{S}_\Pi)$ алгоритм вычисления задачи структурного преобразования, заданного системой (1), с использованием только операции \circ . Сложностью алгоритма $\mathcal{L}(\mathcal{A})$ назовем число операций \circ , требуемое для вычисления всех его совместных редукций. Например, сложность алгоритма \mathcal{A}_B , вычисляющего систему (1) методом «грубой силы», определяется соотношением $\mathcal{L}(\mathcal{A}_B) = \sum_i \#(\pi_i), \pi_i \in \Pi$, где $\# \pi^m \stackrel{\text{def}}{=} m - 1$ означает количество операций \circ в редукции одного паттерна. Очевидно, что в ряде случаев эта сложность может быть снижена за счет переиспользования общих подмножеств.

В ряде задач обработки изображений возникает необходимость рассматривать семейство паттернов, которое может быть порождено одним выделенным образцом (будем называть его *порождающим*) и его многочисленными сдвигами на растре. Пусть $S = \langle s_1, s_2 \rangle$ – вектор смещения. Определим тогда операцию сдвига паттерна π^m на вектор S :

$$\pi^m \gg S = \langle \langle r_{i_1} + s_1, r_{i_2} + s_2 \rangle \mid r_i \in \pi^m, 0 \leq i < m \rangle$$

Теперь, когда все необходимые определения введены, рассмотрим несколько конкретных примеров СПИ.

2. Примеры структурных преобразований изображений

Покажем, что примитивы обработки изображений могут быть сформулированы как задачи вычисления структурного их преобразования. Начнем с вычисления параллельной проекции изображения. На растре $R^{M \times N}$ i -строке или j -столбцу соответствует прямолинейный паттерн, который обозначим $r^M\{i\}$ и $c^N\{j\}$, соответственно. Математическое выражение для вычисления проекции по одной строке или одному столбцу изображений $I^{M \times N}$ совпадает с вычислением операции суммы над множеством яркостных значений пикселей. Пример проекции проиллюстрирован рис. 2.

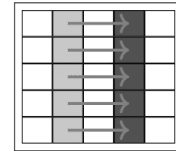


Рис. 2. Паттерн-столбец $c^5\{1\}$ для вертикальной проекции (выделен светло-серым цветом); вектор сдвига $s = \langle 2, 0 \rangle$ (выделен стрелками); сдвиг паттерна $c^5\{1\} \gg s$ (выделен темно-серым цветом).

Таким образом, возникает два структурных преобразования, задаваемых следующим набором редукций:

$$\mathcal{S}_H = \begin{cases} \sum_{i=1}^N r\{0\}[i] \\ \sum_{i=1}^N r\{1\}[i] \\ \dots \\ \sum_{i=1}^N r\{M-1\}[i] \end{cases} \quad \mathcal{S}_V = \begin{cases} \sum_{j=1}^M c\{0\}[j] \\ \sum_{j=1}^M c\{1\}[j] \\ \dots \\ \sum_{j=1}^M c\{N-1\}[j] \end{cases} \quad (2)$$

Данное преобразование выразимо с использованием порождающего паттерна и операции сдвига. Для этого достаточно одного порождающего паттерна π^M для вертикального и π^N для горизонтального направлений. Все остальные паттерны получаются путем сдвига данных паттернов вдоль

соответствующей оси на множество допустимых значений. Допустимым называется сдвиг, при котором пересечение паттерна с растром является непустым множеством. Тогда структурное преобразование определяется следующим набором редукций: $\mathbb{S}_H = \langle \mathcal{R}(c^N \gg \langle j, 0 \rangle, +) \mid 0 \leq j < N \rangle$, $\mathbb{S}_V = \langle \mathcal{R}(r^M \gg \langle 0, i \rangle, +) \mid 0 \leq i < M \rangle$.

Покажем, что вычисление ПХ для прямых также является задачей вычисления СПИ. Для построения Хаф-образа для прямых требуется вычислить суммарную яркость пикселей вдоль множества дискретных прямых, пересекающих растр $R^{M \times N}$. Таким образом, в данном случае речь идет об использовании алгебраической структуры $(\mathbb{Z}_n, +)$.

Рассмотрим прямую $L_{\alpha, \beta}$ на двумерной плоскости, задаваемую двумя параметрами. Как известно, соответствующая ей дискретная прямая может быть задана множеством способов. В работе Р. Клетте и А. Розенфельда [9] проведен качественный обзор различных способов подобных представлений, а также детально проанализированы их свойства. Далее воспользуемся одним из предложенных в указанной работе способом дискретизации прямых. «Ближайшей» дискретной прямой к $L_{\alpha, \beta}$ будем называть паттерн $\pi_{\alpha, \beta} = \langle \langle n, [\alpha \cdot n + \beta] \rangle \mid n \geq 0 \rangle$, где $\alpha \in [-1, 1]$, а $[\cdot]$ – это операция округления до ближайшего целого. Такой паттерн является 8-связной неприводимой цепочкой [9]. Обозначим множество указанных паттернов Π_{dsl}^H . Заметим, что «преимущественно» вертикальные прямые не могут быть описаны подобным образом, поскольку в таком случае паттерны не будут являться упомянутыми 8-связными неприводимыми цепочками. Поэтому по аналогии определим соответствующее множество паттернов для «преимущественно» вертикальных прямых: $\Pi_{dsl}^V = \{ \langle [\alpha \cdot n + \beta], n \rangle \mid n \geq 0 \}$.

Рассмотрим множество всех «ближайших» дискретных прямых, пересекающих растр:

$$\Pi_{dsl} = \Pi_{dsl}^V \cup \Pi_{dsl}^H = \{ \pi_{\alpha, \beta} \mid \pi_{\alpha, \beta} \in \Pi_{\alpha, \beta} \cap \mathbb{R}^{M \times N} \neq \emptyset \}.$$

Согласно работе [10] для квадратного изображения со стороной K количество «ближайших» дискретных прямых можно оценить следующим образом: $|\Pi_{dsl}| < \mathcal{O}(K^4)$. Структурное преобразование для множества Π_{dsl} задается так:

$$\mathbb{S}_{dsl} = \{ \mathcal{R}(\pi, +) \mid \pi \in \Pi_{dsl} \}. \quad (3)$$

Наконец, рассмотрим множество паттернов $\Pi_{HT} \subset \Pi_{dsl}$. Тогда задача вычисления преобразования Хафа для прямых это просто задача вычисления структурного преобразования \mathbb{S}_{HT} .

Теперь рассмотрим аппарат математической морфологии, в котором используются алгебраические структуры с операциями, отличными от суммирования. Для структур $M_1 = (\mathbb{Z}_n, \min)$ и $M_2 = (\mathbb{Z}_n, \max)$ и паттерна π произвольной формы вычислительную задачу естественно формулировать с помощью рассмотренной ранее операции сдвига. При морфологической фильтрации требуется вычислить значение бинарной операции по позициям паттерна π с его сдвигом в каждую допустимую позицию раstra. Сдвиг является допустимым, если после его применения пересечение паттерна с растром оказывается не пусто, то есть

$$S = \{ s \mid (\pi \gg s) \cap R^{M \times N} \neq \emptyset \}.$$

Согласно определению из источника [2], дилатация изображения I с паттерном $P = \{ \langle s_1, s_2 \rangle \mid S_1 \leq |s_1|, S_2 \leq |s_2| \}$ в позиции раstra $\langle x, y \rangle$ обозначается как $(I \oplus P)(x, y)$ и записывается следующим образом:

$$(I \oplus P)(x, y) = \max_{|s_1| \leq S_1, |s_2| \leq S_2} \{ I(x + s_1, y + s_2) \}. \quad (4)$$

Эрозия изображения обозначается как $I \otimes P$ и определяется по аналогии с (4) с заменой операции \max на \min .

Переводя формулу дилатации (4) на «язык» СПИ для одной позиции раstra непосредственно получаем $\mathbb{S}_D = \mathcal{R}(\pi, \max)$ (и $\mathbb{S}_E = \mathcal{R}(\pi, \min)$ для эрозии, соответственно).

В итоге вычисления структурного преобразования для морфологических фильтров по типу «эрозия» и «дилатация» записываются следующим образом:

$$\mathbb{S}_D = \langle \min(\pi \gg s), s \in S \rangle, \mathbb{S}_E = \langle \max(\pi \gg s), s \in S \rangle. \quad (5)$$

Наконец, рассмотрим, как построение интегрального изображения и его последующее использование вписываются в рамки аппарата структурных преобразований изображений. Построение интегрального изображения само по себе является структурным преобразованием, что напрямую следует из формулы:

$$S(i, j) = \sum_{a=1}^i \sum_{b=1}^j I(a, b). \quad (6)$$

Однако куда более важным оказывается его последующее использование при синтезе быстрых методов анализа изображений. В частности, оно является неотъемлемым «ингредиентом» метода П. Виолы и М. Джонса [11] и применяется в нем для эффективного вычисления признаков Хаара. Напомним, что значением признака Хаара является разность между суммами яркости пикселей в двух регионах, которые на типовых признаках с рис. 3 обозначены черным и белым цветом, соот-

ветственно. Очевидно, что каждый из указанных признаков Хаара может быть рассмотрен в качестве порождающего паттерна.



Рис. 3. Примеры типовых признаков Хаара

Для вычисления признаков Хаара с помощью формулы [12]

$$\sum_{a=y_1}^{y_2} \sum_{b=x_1}^{x_2} I(a, b) = S(x_2, y_2) - S(x_1, y_2) - S(x_2, y_1) + S(x_1, y_1) \quad (7)$$

требуется не только операция сложения, но и вычитания, то есть алгебраическая структура $S = (\mathbb{Z}_n, +)$ в таком случае должна быть группой. Тогда если для F обозначить «черные» позиции паттерна как π_+ , а «белые» позиции – как π_- , то соответствующее значение признака применительно к позиции $\langle i, j \rangle$ на изображении I оказывается равно $F^I(i, j) = \sum \pi_+^I - \sum \pi_-^I$. Таким образом, вычисление структурных преобразований отдельно для «черных» и отдельно для «белых» позиций позволяет решить искомую задачу вычисления всех требуемых признаков во всех положениях.

3. Методы ускорения структурных преобразований изображений

В данной работе предлагается три метода оптимизации структурных преобразований изображений, обозначаемых далее S_a , S_x , S_c , которые мы последовательно рассмотрим в трех следующих подразделах.

3.1. Алгоритмическая оптимизация вычислений

В первую очередь рассмотрим алгоритмическую оптимизацию вычислений S_a . Напомним, что по определению СПИ вычисляется за $\mathcal{L}(\mathbb{S}_\Pi) = \sum_i \# \pi_i$, $\pi_i \in \Pi$ операций. Очевидно, что данная оценка не может быть уменьшена, если на множестве его редукций нет каких-либо общих подвыражений. Такому случаю соответствует, например, вычисление параллельных проекций изображения на координатные оси. Однако для многих других задач обработки изображений соответствующие им СПИ имеют довольно много общих подвыражений [5]. Для примера наличия в СПИ подвыражений рассмотрим морфологический фильтр типа «эрозия» размера 3×3 , при вычислении кото-

рого используется алгебраическая структура вида $S = (V, \min)$. На рис. 4 проиллюстрирован случай, когда при вычислении значения такого фильтра в соседних позициях число общих подвыражений оказывается существенным.

7	2	3	6	8	4
4	3	2	1	6	3
5	4	7	5	1	2

Рис. 4. Визуализация общего подмножества пикселей для приложений паттерна $\pi^{3 \times 3}$ в позициях $\langle 1, 2 \rangle$ и $\langle 1, 3 \rangle$ (отмечены кругами). Приложения паттерна в позициях отмечены серым цветом

Обратите внимание на приложение прямоугольного паттерна $\pi^{3 \times 3}$ к позициям раstra $\langle 1, 2 \rangle$ и $\langle 1, 3 \rangle$. Обозначим соответствующие множества значений пикселей $\Omega_1 = \pi_{\langle 1, 2 \rangle}^{3 \times 3}$ и $\Omega_2 = \pi_{\langle 1, 3 \rangle}^{3 \times 3}$, соответственно. Для них $\# \Omega_1 = \# \Omega_2 = 8$. Очевидно, что и в том, и в другом случае возникает необходимость вычислять минимум на множестве значений пикселей $\Omega = \Omega_1 \cap \Omega_2 = \{3, 6, 2, 1, 7, 5\}$. Тем самым для данных двух элементов достаточно следующих редукций:

$$\begin{cases} \mathcal{R}'(\Omega, \min) = \min\{3, 6, 2, 1, 7, 5\}, \\ \mathcal{R}(\pi_{\langle 1, 2 \rangle}^{3 \times 3}, \min) = \min\{2, 3, 4, \mathcal{R}'\}, \\ \mathcal{R}(\pi_{\langle 1, 3 \rangle}^{3 \times 3}, \min) = \min\{8, 6, 1, \mathcal{R}'\}. \end{cases} \quad (8)$$

Тогда $\# \Omega = 5$, $\# \mathcal{R}_{\langle 1, 2 \rangle} = 3$, $\# \mathcal{R}_{\langle 1, 3 \rangle} = 3$, что в сумме дает 11 применений операции минимума против исходных 16.

Выявление, подсчет и последующее переиспользование общих подвыражений является важной стратегией алгоритмической оптимизации при вычислении СПИ. В литературе известен не один пример использования подобной техники в частных случаях. Так, для рассматриваемого выше морфологического фильтра известен алгоритм, который позволяет вычислять требуемые значения минимума или максимума на прямоугольном паттерне всего за 6 операций независимо от размера самого паттерна. Речь идет об алгоритме, впервые предложенном в работе М. ван Херка [13], а годом позже независимо опубликованным Й. Гилом и М. Верманом [14]. Данный алгоритм прекрасно поддается распараллеливанию и хорошо ложится на современные специализированные процессорные инструкции [3]. С его помощью стало доступно применение морфологических фильтров большой площади, что оказалось существенным для выделения ряда графических примитивов [15].

В качестве примера способа алгоритмической оптимизации с использованием алгебраической структуры $S = (V, +)$ можно упомянуть применение интегрального изображения. С его помощью могут быть эффективно реализованы множественные фильтры обработки изображений. Например, классический усредняющий фильтр по прямоугольному окну есть не что иное, как вычисление суммы по этому окну с последующим делением на количество элементов. Поскольку эти операции выполняются для каждого пикселя, то прямоугольные окна, как и в методах морфологической фильтрации, существенно пересекаются. При наличии интегрального изображения каждая сумма может быть вычислена за четыре операции независимо от размера окна, что существенно быстрее наивного подхода. Этот подход широко используется при решении задачи бинаризации изображений [16-18].

3.2. Приближение множества паттернов, снижающее вычислительную сложность

Рассмотрим способ построения БСПИ, опирающийся на аппроксимацию исходного множества паттернов другим набором паттернов S_x . Такая аппроксимация зачастую оказывается полезна, если для предлагаемого взамен набора известны вычислительно более эффективные методы.

В качестве примера для алгебраической структуры $S = (V, +)$ рассмотрим ускорение вычисления преобразования Хафа. В 1992 году за авторством М. Брейди и В. Ёна [19] был опубликован алгоритм, осуществляющий вычисление ПХ для N^2 прямых за $O(N^2 \log N)$ суммаций вместо стандартных $O(N^3)$. К сожалению, в свое время этот алгоритм не получил широкого распространения и поэтому неоднократно переизобретался, как минимум, В. Гётцем и Х. Дрюкмюллером [20] и С.М. Карпенко и Д.П. Николаевым [21]. В настоящее время данный способ вычисления Хаф-образа известен как *быстрое преобразование Хафа* (далее – БПХ). Предпосылкой к существенному снижению числа суммаций в рамках данного алгоритма является наличие общих подсумм, возникающих при аппроксимации прямых дискретными паттернами специального вида, так называемых «диадических паттернов». На рис. 5 приведен пример таких паттернов P_t^8 с наклоном, изменяющимся в диапазоне $[0, 7]$.



Рис. 5. «Преимущественно вертикальные» порождающие паттерны P_t^8 с наклонами $t = \{0, 1, 2, 3, 4, 5, 6, 7\}$ (слева-направо)

На рис. 6 продемонстрировано соотношение между указанными множествами и диапазонами углов.

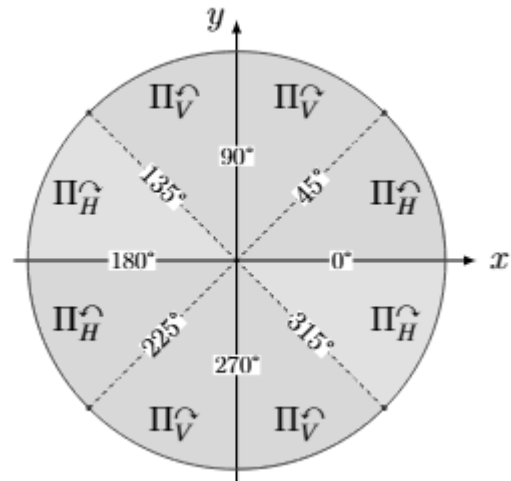


Рис. 6. Связь между диапазонами углов и множествами порождающих паттернов Π

Теперь приведем пример аппроксимации исходного множества паттернов другим набором паттернов для алгебраической структуры $S = (V, \min)$. В морфологической фильтрации изображений зачастую вместо более подходящего по смыслу прикладной задачи «круглого» паттерна используется его грубая аппроксимация в виде квадрата или прямоугольника. Связано это, как правило, именно с наличием упомянутого ранее вычислительно эффективного алгоритма ван Херка для прямоугольных паттернов (Раздел 3.1). Например, в популярной работе по нормализации изображений К. Хе [22] при оценке «темного канала» используется именно квадратный паттерн, также как и в недавней работе И.А. Куниной, посвященной различению изображений цифровой копии документа и его оригинала [23].

Итак, показано, как путем замены одного множества паттернов на другой задача вычисления СПИ сводится к такой, что для нее известен более быстрый алгоритм. Это позволяет добиться повышения быстродействия алгоритмов обработки изображений.

3.3. Аналитическое упрощение композиции СПИ

В задачах обработки изображений, где используется морфологическая фильтрация, часто требуется применить не один фильтр, а сразу несколько. При этом может оказаться, что при вычислении итогового значения возникнет частичное пересечение по операциям. Тогда возникает потребность в упрощении композиции структурных преобразо-

ваний S_c . Рассмотрим, например, применение морфологических фильтров, которые прекрасно разобраны в работе [24], с нетривиальным паттерном. Необходимость в подобной фильтрации возникает при поиске примитивов вида «стыков» или «углов» (англ. junction). Эти примитивы используются при решении таких важных прикладных задач, как детектирование таблиц на изображении документа и точное определение решетки знакомств [25]. В таком случае в каждом пикселе изображения требуется вычислять значения с использованием набора паттернов $\Pi = \{J_p, J_q, J_b, J_d\}$, которые показаны на рис. 7.

J_p	J_q	J_b	J_d
$\begin{pmatrix} (0,0) & (0,1) & (0,2) \\ (1,0) & (1,1) & (1,2) \\ (2,0) & (2,1) & (2,2) \end{pmatrix}$	$\begin{pmatrix} (0,0) & (0,1) & (0,2) \\ (1,0) & (1,1) & (1,2) \\ (2,0) & (2,1) & (2,2) \end{pmatrix}$	$\begin{pmatrix} (0,0) & (0,1) & (0,2) \\ (1,0) & (1,1) & (1,2) \\ (2,0) & (2,1) & (2,2) \end{pmatrix}$	$\begin{pmatrix} (0,0) & (0,1) & (0,2) \\ (1,0) & (1,1) & (1,2) \\ (2,0) & (2,1) & (2,2) \end{pmatrix}$

Рис. 7. Паттерны «стыков» или «углов»

Для каждого из них, на подобии признаков Хаара, требуется определить разницу между максимальной интенсивностью по «черным» позициям π^+ и минимальной интенсивностью по «белым» позициям π^- :

$$J(i, j) \stackrel{\text{def}}{=} \max\{0, \max\{\Omega_{\pi^+}\} - \min\{\Omega_{\pi^-}\}\}. \quad (9)$$

Продemonстрируем способ вычисления данных величин с использованием аппарата СПИ. Обозначим значение пикселя изображения в позиции $\langle 0,0 \rangle$ как x_1 , $\langle 0,1 \rangle$, x_2 и так далее. Иногда для краткости будем опускать обозначение операции, если она уже указана в левой части, а также имена переменных, если они все одинаковы в рамках одной редукции. Тогда в каждом пикселе изображения необходимо вычислить два СПИ (10). После того как они посчитаны, искомые величины без труда вычисляются по формуле (9):

$$S_{max} = \begin{cases} \mathcal{R}(J_{b^+}, max) = \max\{1, 4, 5\} \\ \mathcal{R}(J_{p^+}, max) = \max\{4, 5, 7\} \\ \mathcal{R}(J_{q^+}, max) = \max\{3, 4, 7\} \\ \mathcal{R}(J_{d^+}, max) = \max\{1, 3, 4\} \end{cases}$$

$$S_{min} = \begin{cases} \mathcal{R}(J_{b^-}, min) = \min\{0, 2, 6, 8, 3, 7\} \\ \mathcal{R}(J_{p^-}, min) = \min\{0, 2, 6, 8, 1, 3\} \\ \mathcal{R}(J_{q^-}, min) = \min\{0, 2, 6, 8, 1, 5\} \\ \mathcal{R}(J_{d^-}, min) = \min\{0, 2, 6, 8, 5, 7\} \end{cases} \quad (10)$$

Рассмотрим подробнее вычисление S_{max} . Для одной позиции раstra $\langle i, j \rangle$ требуется применить 8 операций. Однако если ввести два допол-

нительных паттерна t_1 и t_2 , то можно сократить количество операций до 6. В случае S_{min} ситуация становится еще нагляднее. В исходной версии требуется вычислить $4 \times 5 = 20$ операций. При этом поднабор $\{x_0, x_2, x_6, x_8\}$ встречается в каждой редукции и предварительное вычисление минимума на нем позволяет преобразовать S_{min} в S'_{min} и снизить количество необходимых операций до 11. Таким образом, в рамках данного примера для каждого пикселя вместо 28 операций взятия минимума достаточно применить всего 17 (11).

$$S'_{max} = \begin{cases} t_1 = \max\{x_4, x_5\} \\ t_2 = \max\{x_3, x_4\} \\ \mathcal{R}(J_{b^+}, max) = \max\{x_1, t_1\} \\ \mathcal{R}(J_{p^+}, max) = \max\{t_1, x_7\} \\ \mathcal{R}(J_{q^+}, max) = \max\{t_2, x_7\} \\ \mathcal{R}(J_{d^+}, max) = \max\{x_1, t_2\} \end{cases}$$

$$S'_{min} = \begin{cases} t_1 = \min\{x_0, x_2, x_6, x_8\} \\ \mathcal{R}(J_{b^-}, min) = \min\{t_1, x_3, x_7\} \\ \mathcal{R}(J_{p^-}, min) = \min\{t_1, x_1, x_3\} \\ \mathcal{R}(J_{q^-}, min) = \min\{t_1, x_1, x_5\} \\ \mathcal{R}(J_{d^-}, min) = \min\{t_1, x_5, x_7\} \end{cases} \quad (11)$$

4. Примеры замен СПИ на БСПИ в прикладных задачах обработки изображений

Рассмотрим, где и как используется замена СПИ на их быстрые аналоги, как они задействованы в такой важной прикладной сфере, как распознавание изображений документов. Для дальнейшего изложения требуется общее понимание организации систем распознавания изображений документов (далее – СРИД). Эта тема достаточно полно раскрыта в публикациях [26,27].

По функциональному признаку СРИД обычно подразделяется на следующие модули: \mathcal{M}_i – ввода изображения; \mathcal{M}_p – нормализации и оценки качества изображения; \mathcal{M}_t – идентификации типа документа и установления его внутренней системы координат; \mathcal{M}_r – распознавания реквизитов; \mathcal{M}_c – пост-обработки реквизитов и контроля заполнения документа; \mathcal{M}_e – экспорта результатов обработки.

Модуль распознавания реквизитов \mathcal{M}_r обычно состоит из нескольких частей, каждая из которых отвечает за решение задачи распознавания конкретного вида реквизитов. Наиболее распространенные задачи: $\mathcal{T}_{printed}$ – распознавание машинописных текстовых фрагментов; \mathcal{T}_{hand} – распознавание рукопечатных/рукописных фрагментов; \mathcal{T}_{mark} – распознавание отметок выбора и

специальных пометок; \mathcal{T}_{bar} – чтение штриховых и матричных кодов.

Задачи распознавания текстовых фрагментов (как $\mathcal{T}_{printed}$, так и \mathcal{T}_{hand}), как правило, подразделяются на три основных подзадачи: \mathcal{T}_{line_norm} – нормализация изображения текстового фрагмента; \mathcal{T}_{line_segm} – сегментация текстового фрагмента на символы; \mathcal{T}_{line_ocr} – распознавание отдельных символов.

Еще в 2014 году авторитетный греческий ученый В. Гатос перечислил основные примитивы обработки изображений, которые задействованы при решении многих из приведенных выше задач в СРИД [28]. Это перечисление включает скелетизацию, анализ компонент связности, сглаживание длин серий, дистанционное преобразование, анализ проекций изображений (\mathcal{I}_{pp}), методы математической морфологии (\mathcal{I}_{mm}) и, конечно же, преобразование Хафа (\mathcal{I}_{ht}). С точки зрения авторов, наиболее важными являются три последних метода. Кроме того, важным видится еще один – метод интегрирования изображений \mathcal{I}_{ii} , который делает практически применимыми многие алгоритмы (см. Раздел 2).

Связи между модулями, выявленными задачами, методами их решений и СПИ проиллюстрированы рис. 8. Из рисунка видно, как именно методы решения задач в рамках распознавания изображений документов задействуют примитивы обработки изображений, которые были нами выражены на языке структурных преобразований изображений. Тем самым их оптимизация позволяет улучшить работу всей СРИД.

Теперь рассмотрим переход от СПИ к БСПИ на примере решения задачи точной локализации

опорных решеток полей заполнения в анкетах, предложенного в статье [25].

В данной работе алгоритм состоит из следующего набора шагов:

- определить угол наклона зоны интереса на документе и повернуть ее;
- определить положение горизонтальных опорных линий;
- определить положение вертикальных опорных элементов;
- скорректировать положение горизонтальных линий для каждой ячейки.

В основе данных этапов лежит вычисление того или иного СПИ. Так, на первом этапе вместо классического ПХ, который является СПИ, можно использовать алгоритм определения угла наклона на базе анализа БПХ-образа, который является БСПИ, полученным с помощью стратегии S_x (см. Раздел 3.2). Вопросы точности и увеличения производительности при такой замене детально рассмотрены в работе [29]. Для выделения горизонтальных линий и вертикальных элементов можно воспользоваться стыками, которые описаны в Разделе 3.3. Определение положения горизонтальных опорных линий может быть проведено с помощью морфологической фильтрации, которая может быть оптимизирована с помощью подвыражений, то есть с помощью стратегии S_a , описанной в Разделе 3.1.

Переход от СПИ к БСПИ также крайне важен в задачах чтения штрихкодов. Поскольку их распознавание должно производиться в том числе на вычислительно слабых устройствах, к используемым алгоритмам применяются повышенные требования к скорости работы. Так как преобра-

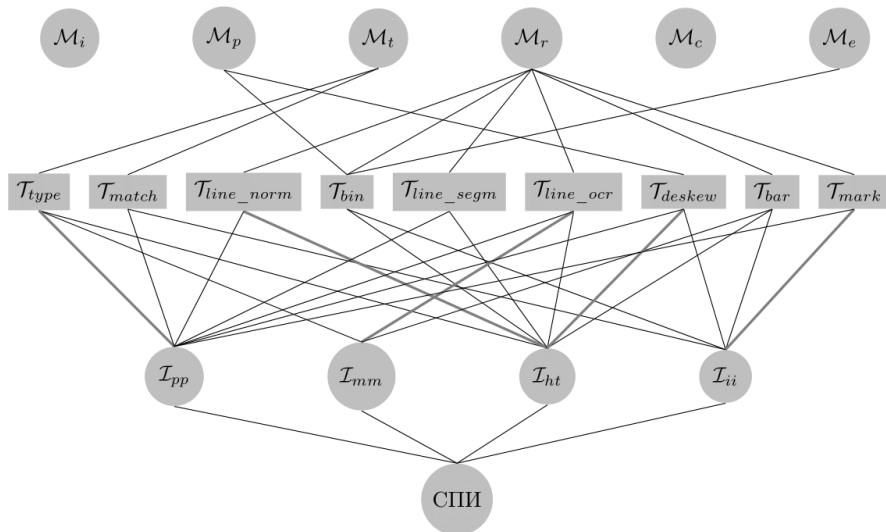


Рис. 8. Схема связи модулей (первая линия), задач обработки изображений (вторая линия) и методов решения данных задач (третья линия)

зование Хафа позволяет извлекать информацию о прямолинейных отрезках, которые часто содержатся в символах штрихкодов согласно особенностям их формирования, то его вычисление регулярно используется для последующего анализа, согласованного с особенностями спецификации символик [30–32]. Поэтому его замена на вычислительно более эффективное БПХ позволяет существенно ускорить данный этап, в том числе на мобильных устройствах. Так, например, алгоритм Брейди-Ёна используется в работе [33] для точной локализации штрихкода символика PDF417 в наведенной зоне интереса, которая совершается с помощью информации о столбчатой структуре этого «сложенного в стопку» кода, а в работе [34] для устойчивого определения сторон в шаблонах наведения Aztec Code-символик. Пример для применения стратегии S_a в штриховом кодировании привести также не сложно – она является важным элементом вычислительно эффективного генеративного распознавания матричных символик фиксированной размерности. Образец такой задачи представлен в работе [35]. Ее характерной особенностью является то, что паттерны, по которым производится суммирование, не отличаются локальной связностью.

БСПИ также активно внедряются вместо СПИ в области вычислительной томографии, в которой они используются в операторах прямого и обратного проецирования, являющихся наиболее вычислительно затратной компонентой почти всех томографических схем. Так, например, в работе А. Долматовой [36] впервые предложен способ ускорения метода «свертки и обратного проецирования» для двумерной томографической реконструкции. Ключевым моментом предложенного метода является применение БПХ для приближенного вычисления обратного дискретного преобразования Радона с интерполяционным пересчетом проекционных данных в координаты параметров s - t . Стоит отметить, что алгоритмически вычисление обратного преобразования сводится к транспонированию изображения и применению прямого БПХ. Предложенная оптимизация позволила достичь асимптотического ускорения томографической реконструкции изображений с $O(N^3)$ до $O(N^2 \log N)$ с несущественной потерей точности. Позже метод был также обобщен на трехмерный случай [37], принципиальный для конусно-круговой схемы сканирования в компьютерной томографии.

Заключение

В работе введен формализм структурных преобразований изображений. Несмотря на то, что

этот формализм относительно прост, он является чрезвычайно полезным на практике и открывает большие возможности для улучшений, поскольку множество ключевых методов анализа и преобразований ЦОИ сводятся к вычислению того или иного СПИ или же их композиции.

Сложность вычисления СПИ «по его определению» часто оказывается неоправданно неэффективна, поэтому поиск более оптимальных методов является важной задачей. В работе выделено три общих метода ускорения вычисления СПИ, то есть построения быстрых СПИ:

- алгоритмическая оптимизация;
- приближение множества паттернов, снижающее вычислительную сложность;
- аналитическое упрощение композиции СПИ.

Предложенный аппарат был успешно апробирован в рамках задачи обработки изображений документов, где он позволил повысить одну из важнейших характеристик СРИД, а именно – скорость ее работы при фиксированном вычислителе.

Отдельный интерес вызывает комбинирование БСПИ с методами обучения искусственных нейронных сетей. Например, БПХ в качестве слоя ИНС используется в архитектуре HoughNet [38], подходящей как для детекции документов, так и для обработки дорожных сцен.

Выбор задач, рассмотренных в данной работе, обусловлен практическими интересами авторов, однако не исчерпывается ими. Предложенный аппарат применим в иных доменах и прикладных системах обработки изображений. Его использование является перспективным с точки зрения масштабной оптимизации таких систем, идентификации и каталогизации новых СПИ и БСПИ, что в свою очередь закладывает фундамент для совершенствования как алгоритмических, так и нейросетевых методов обработки и анализа изображений.

Литература

1. Яне Б. Цифровая обработка изображений. М.: Техносфера. 2007. С. 584.
2. Gonzalez R.C., Woods R.E., Masters B.R. Digital image processing, third edition. 14. 029901.
3. Limonova E., Terekhin A., Nikolaev D., Arlazarov V. Fast implementation of morphological filtering using ARM NEON extension. // IJAER. 2016;11(24):11675–11680.
4. Zekri A.S. Enhancing the matrix transpose operation using intel AVX instruction set extension. // International Journal of Computer Science & Information Technology (IJCSIT). 2014;6(3):67–78. DOI: 10.5121/ijcsit.2014.6305.

5. Хуанг Т.С. (ред.). Быстрые алгоритмы в цифровой обработке изображений. М.: Радио и связь. 1984. С. 224.
6. Weiss B. Fast isotropic median filtering. // SIGGRAPH Conference Papers '25: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers. 2025;(137):1–10. DOI: 10.1145/3721238.3730763.
7. Bezmaternykh P.V., Nikolaev D.P., Arlazarov V.L. High-performance digital image processing. // Pattern Recognit. Image Anal. 2023;33(4):743–755. DOI: 10.1134/S1054661823040090.
8. Винберг Э.Б. Курс алгебры. М.: МЦНМО. 2014. С. 290.
9. Klette R., Rosenfeld A. Digital straightness – a review. // Discrete Applied Mathematics. 2004;139(1-3):197–230. DOI: 10.1016/j.dam.2002.12.001.
10. Koplowitz J., Lindenbaum M., Bruckstein A. The number of digital straight lines on an NxN grid. // IEEE Transactions on Information Theory. 1990;36(1):192–197. DOI: 10.1109/18.50392.
11. Viola P., Jones M. Robust real-time object detection. // International Journal of Computer Vision – IJCV. 2001;57(2).
12. Crow F.C. Summed-area tables for texture mapping. // In Proceedings of the 11th annual conference on Computer graphics and interactive techniques – SIGGRAPH 84. ACM Press. 1984. P. 207–212. DOI: 10.1145/800031.808600.
13. Van Herk M. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. // Pattern Recognition Letters. 1992;13(7):517–521. DOI: 10.1016/0167-8655(92)90069-C.
14. Gil J., Werman M. Computing 2-d min, median, and max filters. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993;15(5):504–507.
15. Слугин Д.Г., Арлазаров В.В. Поиск текстовых полей документа с помощью методов обработки изображений // Труды ИСА РАН. 2017. Т. 67. № 4. С. 65–73.
16. Bradley D., Roth G. Adaptive thresholding using the integral image. // Journal of Graphics Tools. 2007;12(2):13–21. DOI: 10.1080/2151237x.2007.10129236.
17. Shafait F., Keysers D., Breuel T. Efficient implementation of local adaptive thresholding techniques using integral images // In Document Recognition and Retrieval XV. 2008;6815:317–322. International Society for Optics and Photonics, SPIE. DOI: 10.1117/12.767755.
18. Samorodova O.A., Samorodov A.V. Fast implementation of the Niblack binarization algorithm for microscope image segmentation. // Pattern Recognition and Image Analysis. 2016;26(3):548–551. DOI: 10.1134/s1054661816030020.
19. Brady M.L., Yong W. Fast parallel discrete approximation algorithms for the radon transform. // In Proceedings of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '92. 1992. P. 91–99. ACM. DOI: 10.1145/140901.140911.
20. Gotz W.A., Druckmuller H.J. A fast digital Radon transform – an efficient means for evaluating the hough transform // Pattern Recognition. 1995;28:1985–1992.
21. Карпенко С.М., Николаев Д.П., Николаев П.П., Постников В.В. Быстрое преобразование Хафа с управляемой робастностью // In IEEE AIS'04, CAD-2004. 2004. Т. 2. С. 303–309.
22. He K., Sun J., Tang X. Single image haze removal using dark channel prior. // In 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009. P. 1956–1963. DOI: 10.1109/CVPR.2009.5206515.
23. Kunina I.A., Sher A.V., Nikolaev D.P. Screen recapture detection based on color-texture analysis of document boundary regions // Computer Optics. 2022.
24. Пытьев Ю.П., Чуличков А.И. Методы морфологического анализа изображений. М.: ФИЗМАТЛИТ. 2010. С. 336.
25. Куроптев А.В., Николаев Д.П., Постников В.В. Точная локализация опорных решеток полей заполнения в анкетах методами динамического программирования и морфологической фильтрации. // Труды ИСА РАН. 2013. Т. 63. № 3. С. 111–116.
26. Bulatov K.B., Bezmaternykh P.V., Nikolaev D.P., Arlazarov V.V. Towards a unified framework for identity documents analysis and recognition. // Computer Optics. 2022;46(3):436–454. DOI: 10.18287/2412-6179-CO-1024.
27. Арлазаров В.В., Постников В.В., Шоломов Д.Л. Cognitive forms – система массового ввода структурированных документов. // Труды ИСА РАН. 2002. Т. 1. С. 35–46.
28. Gatos B.G. Imaging techniques in document analysis processes. // In Handbook of Document Image Processing and Recognition. 2014. P. 73–131. Springer London. DOI: 10.1007/978-0-85729-859-1_4.
29. Безматерных П.В. Нормализация изображения текста с помощью быстрого преобразова-

- ния Хафа // ИТиВС. 2024. Т. 4. С. 3-16. DOI: 10.14357/20718632240401.
30. *Muniz R.* A robust software barcode reader using the Hough transform // Proceedings 1999 International Conference on Information Intelligence and Systems. 1999. P. 313–319. DOI: 10.1109/ICIIS.1999.810282.
 31. *Szentandrási I., Herout A., Dubska M.* Fast detection and recognition of QR codes in high-resolution images. // In Proceedings of the 28th Spring Conference on Computer Graphics (SCCG '12). 2012. P. 129–136. DOI: 10.1145/2448531.2448548.
 32. *Zamberletti A., Gallo I., Albertini S., Noce L.* Neural 1D Barcode Detection Using the Hough Transform // IPSJ Transactions on Computer Vision and Applications. 2014;7:1-9. DOI: 10.2197/ipsjtcva.7.1.
 33. *Mitrofanov D.G., Zlobin P.K., Shemiakina Y.A., Bezmaternykh P.V.* Precise localization of PDF417 code based on fast hough transform // Vestnik YuUrGU «Vychislitel'naya matematika i informatika». 2024;13(4):19-34. DOI: 10.14529/cmse240402.
 34. *Martynov S.I., Bezmaternykh P.V.* Aztec core symbol detection method based on connected components extraction and contour signature analysis. // ICMV 2019. Society of Photo-Optical Instrumentation Engineers (SPIE). 2020. P. 11433. DOI: 10.1117/12.2559183.
 35. *Безматерных П.В., Вылегжанин Д.В., Гладиллин С.А., Николаев Д.П.* Генеративное распознавание двумерных штрихкодов. // ИИиПР. 2010. Т. 4. С. 63–69.
 36. *Долматова А.В., Николаев Д.П.* Ускорение свертки и обратного проецирования при реконструкции томографических изображений // Сенсорные системы. 2020. Т. 34. № 1. С. 64–71. DOI: 10.31857/S0235009220010072.
 37. *Bulatov K., Chukalina M., Kutukova K., Kohan V., Ingacheva A., Buzmakov A., Arlazarov V.V., Zschech E.* Monitored tomographic reconstruction - An advanced tool to study the 3D morphology of nanomaterials // Nanomaterials. 2021;11(10):1-12. DOI: 10.3390/nano11102524.
 38. *Sheshkus A., Ingacheva A., Arlazarov V., Nikolaev D.* HoughNet: neural network architecture for vanishing points detection // In ICDAR 2019. 2020;(8978201):844–849. The Institute of Electrical and Electronics Engineers (IEEE).

Мехова Мария Сергеевна. Национальный исследовательский технологический университет «МИСИС», г. Москва, Россия. Студентка. Институт проблем передачи информации им. А.А. Харкевича Российской академии наук, г. Москва, Россия. Стажер-исследователь. ООО «Смарт Энджинс Сервис», г. Москва, Россия. Лаборант-программист. Область научных интересов: обработка изображений, компьютерная томография. E-mail: m.mehova@smartengines.com (Ответственная за переписку).

Безматерных Павел Владимирович. Федеральный исследовательский центр «Информатика и управление» Российской академии наук, г. Москва, Россия. Программист первой категории. ООО «Смарт Энджинс Сервис», г. Москва, Россия. Научный сотрудник-программист. Область научных интересов: обработка изображений, распознавание штрихкодов, анализ документов. E-mail: bezmpavel@gmail.com

A formal approach to image structural transformations and optimization

M.S. Mehova^{I,III,IV}, P.V. Bezmaternykh^{I,II}

^I Smart Engines Service LLC, Moscow, Russia

^{II} Federal Research Center «Computer Science and Control», Russian Academy of Sciences, Moscow, Russia

^{III} National University of Science and Technology «MISIS», Moscow, Russia

^{IV} Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia

Abstract. Low-level image processing methods and their transformations are crucial across many diverse fields. Although many studies have examined their optimization in particular cases, a thorough, systematic approach remains unavailable. This paper introduces a formal structural image transformation approach to address this issue. It offers a unified way to express various image processing methods and transformations, analyze their computational complexity, and identify ways for their computation acceleration. This paper describes three optimization techniques that convert structural image transformations to faster versions, decreasing their complexity relative to the independent calculation of each element of the initial transformation. Finally, it explores the application of these methods to tasks like automated document image processing, barcode reading, and computed tomography.

Keywords: *image processing, combinatorial optimization, automatic document image processing.*

DOI: 10.14357/20790279250409 **EDN:** YUDLHH

References

1. Yane B. Digital Image Processing. M.: Technosfera. 2007. P. 584.
2. Gonzalez R.C., Woods R.E., Masters B.R. Digital image processing, third edition. 14. 029901.
3. Limonova E., Terekhin A., Nikolaev D., Arlazarov V. Fast implementation of morphological filtering using ARM NEON extension. // IJAER. 2016;11(24):11675–11680.
4. Zekri A.S. Enhancing the matrix transpose operation using intel AVX instruction set extension. // International Journal of Computer Science & Information Technology (IJCSIT). 2014;6(3):67–78. DOI: 10.5121/ijcsit.2014.6305.
5. Huang T.S. (ed.). Fast Algorithms in Digital Image Processing. Moscow: Radio i svyaz. 1984. P. 224.
6. Weiss B. Fast isotropic median filtering. // SIGGRAPH Conference Papers '25: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers. 2025;(137):1–10. DOI: 10.1145/3721238.3730763.
7. Bezmaternykh P.V., Nikolaev D.P., Arlazarov V.L. High-performance digital image processing. // Pattern Recognit. Image Anal. 2023;33(4):743–755. DOI: 10.1134/S1054661823040090.
8. Vinberg E.B. Algebra course. Moscow: MCNMO. 2019. P. 290.
9. Klette R., Rosenfeld A. Digital straightness – a review. // Discrete Applied Mathematics. 2004;139(1-3):197–230. DOI: 10.1016/j.dam.2002.12.001.
10. Koplowitz J., Lindenbaum M., Bruckstein A. The number of digital straight lines on an NxN grid. // IEEE Transactions on Information Theory. 1990;36(1):192–197. DOI: 10.1109/18.50392.
11. Viola P., Jones M. Robust real-time object detection. // International Journal of Computer Vision – IJCV. 2001;57(2).
12. Crow F.C. Summed-area tables for texture mapping. // In Proceedings of the 11th annual conference on Computer graphics and interactive techniques - SIGGRAPH 84. ACM Press. 1984;207-212. DOI: 10.1145/800031.808600.
13. van Herk M. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. // Pattern Recognition Letters. 1992;13(7):517–521. DOI: 10.1016/0167-8655(92)90069-C.
14. Gil J., Werman M. Computing 2-d min, median, and max filters. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993;15(5):504–507.
15. Slugin D.G., Arlazarov V.V. Text fields extraction based on image processing. // Trudy ISA RAN. 2017;67(4):65–73.
16. Bradley D., Roth G. Adaptive thresholding using the integral image. // Journal of Graphics Tools. 2007;12(2):13–21. DOI: 10.1080/2151237x.2007.10129236.
17. Shafait F., Keysers D., Breuel T. Efficient implementation of local adaptive thresholding techniques using integral images. // In Document Recognition and Retrieval XV, 2008;6815:317–322. International Society for Optics and Photonics, SPIE. DOI: 10.1117/12.767755.

18. *Samorodova O.A., Samorodov A.V.* Fast implementation of the Niblack binarization algorithm for microscope image segmentation // *Pattern Recognition and Image Analysis*. 2016;26(3):548–551. DOI: 10.1134/s1054661816030020.
19. *Brady M.L., Yong W.* Fast parallel discrete approximation algorithms for the radon transform // In *Proceedings of the Fourth Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '92*. 1992. P. 91–99. ACM. DOI: 10.1145/140901.140911.
20. *Gotz W.A., Druckmuller H.J.* A fast digital Radon transform – an efficient means for evaluating the hough transform // *Pattern Recognition*. 1995;28:1985–1992.
21. *Karpenko S.M., Nikolaev D.П., Nikolaev P.P., Postnikov V.V.* Bystroe preobrazovanie Hafa s upravlyaemoy robustnostyu // In *IEEE AIS'04, CAD-2004*. 2004;2:303–309.
22. *He K., Sun J., Tang X.* Single image haze removal using dark channel prior // In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009. P. 1956–1963. DOI: 10.1109/CVPR.2009.5206515.
23. *Kunina I.A., Sher A.V., Nikolaev D.P.* Screen recapture detection based on color-texture analysis of document boundary regions // *Computer Optics*. 2022.
24. *Pytyev U.P., Chulichkov A.I.* *Methody morfologicheskogo analiza izobrazheniy*. M.: FIZMATLIT. 2010. P. 336.
25. *Kuroptev A.V., Nikolaev D.P., Postnikov V.V.* Tochnaya localizacia opornykh reshetok poley zapolneniya v anketakh metodami dinamicheskogo programirovaniya i morfologicheskoy filtracii. // *Trudy ISA RAN*. 2013;63(3):111–116.
26. *Bulatov K.B., Bezmaternykh P.V., Nikolaev D.P., Arlazarov V.V.* Towards a unified framework for identity documents analysis and recognition. // *Computer Optics*. 2022;46(3):436–454. DOI: 10.18287/2412-6179-CO-1024.
27. *Arlazarov V.V., Postnikov V.V., Sholomov D.L.* Cognitive forms – sistema massovogo vvoda strukturirovannykh dokumentov. // *Trudy ISA RAN*. 2002;1:35–46.
28. *Gatos B.G.* Imaging techniques in document analysis processes. // In *Handbook of Document Image Processing and Recognition*. 2014. P. 73–131. Springer London. DOI: 10.1007/978-0-85729-859-1_4.
29. *Bezmaternykh P.V.* Text Image Normalization Using Fast Hough Transform // *ITiVS*. 2024;(4):3–16. DOI: 10.14357/20718632240401.
30. *Muniz R.* A robust software barcode reader using the Hough transform // *Proceedings 1999 International Conference on Information Intelligence and Systems*. 1999. P. 313–319. DOI: 10.1109/ICIIS.1999.810282.
31. *Szentandrási I., Herout A., Dubská M.* Fast detection and recognition of QR codes in high-resolution images // In *Proceedings of the 28th Spring Conference on Computer Graphics (SCCG '12)*. 2012. P. 129–136. DOI: 10.1145/2448531.2448548.
32. *Zamberletti A., Gallo I., Albertini S., Noce L.* Neural 1D Barcode Detection Using the Hough Transform // *IPSI Transactions on Computer Vision and Applications*. 2014;7:1–9. DOI: 10.2197/ipsjtca.7.1.
33. *Mitrofanov D.G., Zlobin P.K., Shemiakina Y.A., Bezmaternykh P.V.* Precise localization of PDF417 code based on fast hough transform // *Vestnik YuUrGU «Vychislitel'naya matematika i informatika»*. 2024;13(4):19–34. DOI: 10.14529/cmse240402.
34. *Martynov S.I., Bezmaternykh P.V.* Aztec core symbol detection method based on connected components extraction and contour signature analysis // *ICMV 2019, Society of Photo-Optical Instrumentation Engineers (SPIE)*. 2020. P. 11433. DOI: 10.1117/12.2559183.
35. *Bezmaternykh P.V., Vylegzhanin D.V., Gladilin S.A., Nikolaev D.P.* Generativnoe raspoznavanie dvumernykh shtrikhkodov // *IiPR*. 2010;(4):63–69.
36. *Dolmatova A.V., Nikolaev D.P.* Fast filtering and back projection for CT image reconstruction // *Sensornye sistemy*. 2020;34(1):64–71. DOI: 10.31857/S0235009220010072.
37. *Bulatov K., Chukalina M., Kutukova K., Kohan V., Ingacheva A., Buzmakov A., Arlazarov V.V., Zschech E.* Monitored tomographic reconstruction - An advanced tool to study the 3D morphology of nanomaterials // *Nanomaterials*. 2021;11(10):1–12. DOI: 10.3390/nano11102524.
38. *Sheshkus A., Ingacheva A., Arlazarov V., Nikolaev D.* HoughNet: neural network architecture for vanishing points detection // In *ICDAR 2019*. 2020;(8978201):844–849. The Institute of Electrical and Electronics Engineers (IEEE).

Maria S. Mehova. National University of Science and Technology «MISIS», Moscow, Russia. Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia. Smart Engines Service LLC, Moscow, Russia. Topics of interest: image processing, computational tomography. E-mail: m.mehova@smartengines.com

Pavel V. Bezmaternykh. Federal Research Center «Computer Science and Control», Russian Academy of Sciences, Moscow, Russia. Smart Engines Service LLC, Moscow, Russia. Topics of interest: image processing, barcode recognition, document image analysis. E-mail: bezmpavel@gmail.com