

К вопросу о восстановлении изображения по стертому коду

Д. В. Алексеев¹

Предлагается алгоритм для восстановления изображения по стертому коду, т.е. коду из которого удалена информация об индексах. Алгоритм обладает полиномиальной сложностью. Введено понятие треугольников общего положения. Для множества точек, образующих треугольники общего положения, доказана корректность работы алгоритма восстановления.

Ключевые слова: код изображения, стертый код, кодирование изображений, аффинная эквивалентность.

1. Введение

Одной из задач распознавания образов является задача распознавания зрительных образов. Одним из вариантов распознавания является сопоставление с известным образцом. Очевидно, изображение может быть сдвинуто, повернуто, иметь другой масштаб. При этом человечески глаз распознает эти изображения (см. рис. 1(a)). Таким образом имеет смысл отождествлять изображения, полученные путем применения преобразования из некоторого класса геометрических преобразований. В частности, одним из таких классов является множество аффинных преобразований. Этот класс является достаточно богатым и включает в себя сдвиги, повороты, растяжения. Заметим, что на практике, когда, например, изображение проектируется на сетчатку глаза или фоточувствительный элемент камеры, с плоским изображением происходят, вообще говоря, проективные преобразования. Впрочем, если расстояние до объекта велико по сравнению с его характерными размерами, то можно считать аффинные преобразования достаточно хорошим приближением проективных.

Каждое изображение может быть представлено как множество точек (см. рис. 1(b)). Заметим, что координатное представление этих точек, очевидно, зависит от выбора системы координат. Кроме того, оно не является инвариантным относительно геометрических преобразований.

¹Алексеев Дмитрий Владимирович — кандидат физ.-мат. наук, с.н.с. лаборатории Проблем теоретической кибернетики мех.-мат. ф-та МГУ, e-mail: alekseev@intsys.msu.ru

Alekseev Dmitriy Vladimirovich — Candidate of Physical and Matematical Sciences, senior staff scientist, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Problems of Theorecical Cybernetics Lab.

Поэтому имеет смысл рассматривать другие представления, обладающие свойством инвариантности.

В.Н. Козлов в [7] ввел понятие *кода изображения*, являющегося *правильным* для какого-то семейства геометрических преобразований. Кодом является набор значений, инвариантный относительно этого семейства. Код является *правильным*, в случае, когда верно и обратное: из совпадения кодов вытекает эквивалентность фигур относительно данного семейства преобразований. Например, код, состоящий из попарных расстояний между точками, является правильным относительно движений плоскости. Там же в [7] был предложен код, состоящий из попарных отношений площадей треугольников, и доказана теорема о том, что такой код является правильным для множества невырожденных аффинных преобразований плоскости. В ходе доказательства теоремы был представлен алгоритм восстановления исходного изображения по его коду (с точностью до аффинной эквивалентности). Позднее П.Г. Агниашвили в [1] обобщил этот метод на случай n -мерного пространства. В [2] были выведены характеристические свойства кодов, т.е. условия, при которых некоторый набор чисел является кодом существующего изображения. Построение кода, правильного относительно невырожденных проективных преобразований плоскости, было сделано в [3].

Стертым кодом называется код, который получен из кода изображения стиранием информации об индексах. Впервые задача была сформулирована акад. В.Б. Кудрявцевым более 40 лет назад следующим образом: Если оставить только попарные расстояния между точками изображения («палки»), и выстроить их в некотором порядке, например, возрастающем, и сделать огибающую у этой «гребенки», то можно ли осуществлять распознавание по виду огибающей? Позднее задача была переформулирована для кода, сохраняющего аффинную эквивалентность. На данный момент автору не известны работы, содержащие продвижения по решению этих задач.

Стертый код может быть использован, например, как показано в [4], для построения алгоритма цифровой подписи.

В данной работе рассматривается задача восстановления изображения по стертому коду с точностью до аффинной эквивалентности. Предложено определение *треугольников общего положения*. Предложен алгоритм восстановления и доказана его корректность при условии, что точки исходного изображения образуют треугольники общего положения. Алгоритм восстановления обладает полиномиальной сложностью.

Автор хотел бы выразить благодарность проф. В.Н. Козлову за внимание к работе. Также автор хотел бы выразить этой статьей дань памяти акад. В.Б. Кудрявцева.

Дальнейшая часть работы организована следующим образом. В разделе 2 даются необходимые определения и формулировка основного результата. В разделе 3 описан алгоритм восстановления изображения по стертому коду и доказана его корректность. В разделе 4 дана алгоритмическая сложность и описаны результаты численных экспериментов. В разделе 5 содержатся выводы и дальнейшие планы.

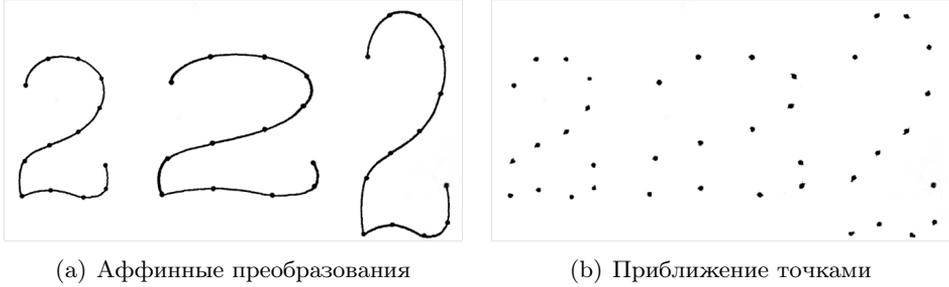


Рис. 1.

2. Основные понятия и формулировка результата

Определение 1. *Изображением* будем называть непустое конечное множество точек на плоскости $A = \{a_1, \dots, a_n\}$.

Определение 2. *Кодом* изображения $A = \{a_1, \dots, a_n\}$ (при $n \geq 3$) будем называть множество $\mathcal{R}(A) = \left\{ \rho_{ijk,pqr} = \frac{S(\Delta a_i a_j a_k)}{S(\Delta a_p a_q a_r)} \right\}^1$. При этом элементы, полученные перестановками индексов i, j, k и p, q, r будем считать одним и тем же элементом кода, таким образом мощность множества $|\mathcal{R}(A)| = (C_n^3)^2$.

Далее, в данной работе будут рассматриваться только такие изображения, в которых никакие три точки не расположены на одной прямой.

Свойства кодов изображения

- 1) $\rho_{ijk,ijk} = 1$;
- 2) $\rho_{pqr,ijk} = (\rho_{ijk,pqr})^{-1}$;
- 3) $\rho_{ijk,pqr} \times \rho_{pqr,uvw} = \rho_{ijk,uvw}$;

¹Если точки a_p, a_q, a_r расположены на одной прямой, то ставим формальный символ $\rho_{ijk,pqr} = \infty$

- 4) $\pm \rho_{ijk,*} \pm \rho_{ikl,*} = \pm \rho_{ijl,*} \pm \rho_{jkl,*}$, где на месте «*» стоит произвольная (одна и та же) тройка, а знаки выбираются в соответствии с ориентацией треугольника, площадь которого стоит в знаменателе. Например, если вершины a_i, a_j, a_k идут против часовой стрелки, то $\rho_{ijk,*}$ идет со знаком «+», а иначе — со знаком «-».

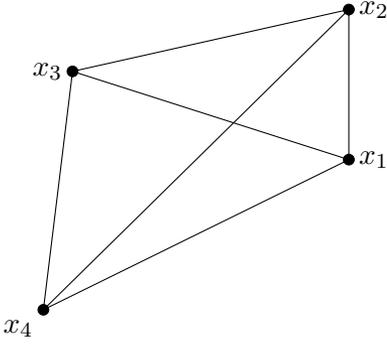
В работе [2] было показано, что эти свойства кода являются в некотором смысле характеристическими. А именно, если зафиксировать ориентацию каждого треугольника и взять набор произвольных чисел $\rho_{ijk,pqr}$, удовлетворяющий свойствам 1-4, то по этому набору строится (причем однозначно) множество точек, кодом которого являются эти числа.

Определение 3. Стертым кодом изображения $A = \{a_1, \dots, a_n\}$ будем называть мультимножество $\mathcal{Z}(A) = \{\rho_1, \dots, \rho_M\}$, $M = (C_n^3)^2$, полученное «стиранием» индексов во множестве $\mathcal{R}(A)$.

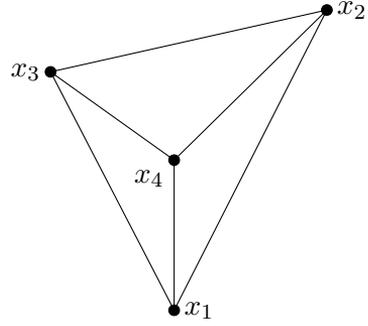
Определение 4. Будем говорить, что множество точек A образует множество треугольников общего положения $\mathfrak{T}(A)$, если выполнены свойства:

- 1) Никакие три точки не лежат на одной прямой, другими словами, все треугольники, образованные тройками точек из A — невырожденные;
- 2) Все нетривиальные (т.е. не $\rho_{ijk,ijk} = 1$) элементы кода $\mathcal{R}(A) = \{\rho_{ijk,lmn}\}$ попарно различны; Замечание: из этого вытекает, что площади всех треугольников, образованных точками из A , также различны;
- 3) Равенство $S(\Delta_1) + S(\Delta_2) = S(\Delta_3) + S(\Delta_4)$ выполняется только в случае, когда объединения треугольников $\Delta_1 \cup \Delta_2$ и $\Delta_3 \cup \Delta_4$ образуют один и тот же четырехугольник (см. рис. 2(a));
- 4) Равенство $S(\Delta_1) + S(\Delta_2) + S(\Delta_3) = S(\Delta_4)$ выполняется только в случае, когда объединения треугольников $\Delta_1 \cup \Delta_2 \cup \Delta_3$ образует Δ_4 (см. рис. 2(b));
- 5) Равенство $\rho_{i_1 j_1 k_1, l_1 m_1 n_1} \cdot \rho_{i_2 j_2 k_2, l_2 m_2 n_2} = \rho_{i_2 j_2 k_2, l_2 m_2 n_2}$ выполнено тогда и только тогда, когда $\{l_1, m_1, n_1\} = \{i_2, j_2, k_2\}$ или $\{i_1, j_1, k_1\} = \{l_2, m_2, n_2\}$, т.е. когда один и тот же треугольник присутствует в числителе и знаменателе.

Другими словами, множество треугольников общего положения таково, что между площадями треугольников отсутствуют нетривиальные соотношения, кроме тех, которые вытекают из равноставленности фигур.



(a) $\Delta x_1 x_2 x_3 \cup \Delta x_1 x_3 x_4 = \Delta x_1 x_2 x_4 \cup \Delta x_2 x_3 x_4$



(b) $\Delta x_1 x_2 x_4 \cup \Delta x_2 x_3 x_4 \cup \Delta x_1 x_3 x_4 = \Delta x_1 x_2 x_3$

Рис. 2. Треугольники общего положения

Определение 5. Если множество A' не аффинно эквивалентно множеству A , но обладает идентичным стертым кодом $\mathcal{Z}(A) \equiv \mathcal{Z}(A')$, то A' называем призраком множества A .

Замечание 1. Отношение «быть призраком» симметрично, т.е. A является призраком A' .

Гипотеза У множества треугольников общего положения призраков не существует.

Теорема 1. Пусть A образует множество треугольников общего положения. Существует алгоритм, который по стертому коду $\mathcal{Z}(A)$ либо строит изображение аффинно-эквивалентно исходному A , либо два изображения, одно из которых аффинно-эквивалентно A , а другое является его призраком A' .

3. Описание алгоритма восстановления и доказательство его корректности.

Для обоснования алгоритма докажем несколько вспомогательных утверждений.

Лемма 1. Рассмотрим произвольные треугольники $\Delta', \Delta'' \in \mathfrak{T}(A)$. Треугольники Δ' и Δ'' имеют общую сторону тогда и только тогда, когда существуют $\Delta_i, \Delta_j \in \mathfrak{T}(A)$ и $e_l \in \{\pm 1\}, l = 1, 2, 3, 4$, такие, что

$$e_1 S(\Delta') + e_2 S(\Delta'') + e_3 S(\Delta_i) + e_4 S(\Delta_j) = 0$$

$A = \{a_1, \dots, a_n\}$	Множество точек (изображение)
$A' = A \setminus \Delta_{\min}$	Множество точек без вершин минимального треугольника
$\mathcal{R}(A)$	Код изображения A
$\mathcal{Z}(A)$	Стертый код изображения A
$\mathfrak{S}(A)$	Множество пар $\rho^{(1)}, \rho^{(2)} \in \mathcal{Z}(A)$, для которых выполнено $\rho^{(1)} \cdot \rho^{(2)} = \rho^*$
$\mathfrak{T}(A)$	Множество всех треугольников с вершинами из A
$\mathfrak{T}'(A)$	Множество всех треугольников $\mathfrak{T}(A)$ без минимального
$\mathfrak{P}(A)$	Множество троек, соответствующих точкам A
$\mathcal{T}^*(A)$	Подмножество $\mathfrak{T}(A)$ — те треугольники, которые входят в тройки $\mathfrak{P}(A)$
β	Биекция между треугольниками $\mathfrak{T}'(A)$ и множеством пар $\mathfrak{S}(A)$
τ^*	Биекция между точками A' и соответствующими тройками

Таблица 1. Список обозначений

Доказательство. Если в равенстве $\pm S(\Delta') \pm S(\Delta'') \pm S(\Delta_i) \pm S(\Delta_j) = 0$ знаков «плюс» больше, чем знаков «минус», то домножим его на -1 . Поскольку все знаки не могут быть одинаковыми, то в итоге получим либо равенство вида $S_1 + S_2 = S_3 + S_4$, либо вида $S_1 + S_2 + S_3 = S_4$, где $S_k \in \{S(\Delta'), S(\Delta''), S(\Delta_i), S(\Delta_j)\}$, $k = 1, 2, 3, 4$. В первом случае по свойству 3 треугольников общего положения получим, что любые два треугольника из этого множества имеют общую сторону. А во втором случае аналогичное утверждение вытекает из свойства 4.

В обратную сторону, пусть треугольники $\Delta_{i_1 i_2 i_3}$ и $\Delta_{i_1 i_2 i_4}$ имеют общую сторону $a_{i_1} a_{i_2}$. Точки a_{i_3} и a_{i_4} могут лежать по одну сторону от прямой $(a_{i_1} a_{i_2})$ или по разные стороны от нее. Выпуклая оболочка всех четырех точек может представлять собой треугольник или четырехугольник. Получаем 4 случая:

- Случай 1 (a_{i_3} и a_{i_4} по одну сторону, оболочка — треугольник): Не ограничивая общности, можно считать, что выпуклая оболочка это $\Delta_{i_1 i_2 i_3}$, в этом случае $S_{i_1 i_2 i_3} = S_{i_1 i_2 i_4} + S_{i_2 i_3 i_4} + S_{i_3 i_1 i_4}$ (см. рис. 3(a)).
- Случай 2 (a_{i_3} и a_{i_4} по одну сторону, оболочка — четырехугольник): Не ограничивая общности, можно считать, что порядок вершин при обходе выпуклой оболочки — $a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4}$, следовательно $S_{i_1 i_2 i_3} + S_{i_1 i_3 i_4} = S_{i_2 i_3 i_4} + S_{i_3 i_4 i_1}$ (см. рис. 3(b)).

- Случай 3 (a_{i_3} и a_{i_4} по разные стороны, треугольник): Не ограничивая общности, можно считать, что выпуклая оболочка $— \Delta_{i_1 i_3 i_4}$, в этом случае $S_{i_1 i_3 i_4} = S_{i_1 i_2 i_3} + S_{i_1 i_2 i_4} + S_{i_2 i_3 i_4}$ (см. рис. 3(c)).
- Случай 4 (a_{i_3} и a_{i_4} по разные стороны, четырехугольник): Не ограничивая общности, можно считать, что порядок вершин при обходе выпуклой оболочки $— a_{i_1}, a_{i_3}, a_{i_2}, a_{i_4}$, следовательно $S_{i_1 i_2 i_3} + S_{i_1 i_2 i_4} = S_{i_2 i_3 i_4} + S_{i_1 i_3 i_4}$ (см. рис. 3(d)).

□

Определение 6. Зафиксируем треугольник $\Delta^* \in \mathfrak{T}(A)$, не содержащий точек из A внутри и будем называть его центральным. Рассмотрим множество треугольников из $\mathfrak{T}(A)$, имеющих общую сторону с центральным. Можно разбить их на тройки следующим образом. Пусть $\Delta^* = \Delta_{a_1 a_2 a_3}$, для произвольной точки $a_i, i = 4, 5, \dots, n$ рассмотрим треугольники $\Delta_{1i} = \Delta_{a_2 a_3 a_i}$, $\Delta_{2i} = \Delta_{a_1 a_3 a_i}$, $\Delta_{3i} = \Delta_{a_1 a_2 a_i}$. Будем называть такую тройку $T_i = \{\Delta_{1i}, \Delta_{2i}, \Delta_{3i}\}$ соответствующей вершине a_i . Будем обозначать $\mathfrak{P}(A)$ множество всех таких троек и обозначать соответствие $\tau^* : A^* = A \setminus \{a_1, a_2, a_3\} \rightarrow \mathfrak{P}(A)$. Множество всех треугольников, входящих в соответствующие тройки будем обозначать $\mathcal{T}^*(A)$.

Замечание 2. Отображение τ^* является биективным по построению. Действительно, взяв разные вершины $a_i, a_j, i \neq j$, получим разные тройки. С другой стороны, множество $\mathfrak{P}(A)$ задано как образ этого отображения.

Рассмотрим тройку с вершинами $a_1, a_2, a_3, a_i, i = 4, \dots, n$. Найдём выпуклую оболочку этих точек. Очевидно, она может состоять из 3 или 4 точек.

- Допустим, что она состоит из 3 точек. Очевидно, a_i не может лежать внутри по выбору центрального треугольника. Значит внутри лежит вершина центрального треугольника, не ограничивая общности, допустим, вершина a_3 (см. рис. 4(a)). В этом случае будем называть эту тройку тройкой 1-го типа, а треугольник Δ_{12i} — главным (для этой тройки).
- Допустим, что она состоит из 4 точек. Очевидно, одна из сторон центрального треугольника будет диагональю этого четырехугольника. Не ограничивая общности, допустим, что это сторона $a_1 a_2$ (см. рис. 4(b)). В этом случае будем называть эту тройку тройкой 2-го типа, а треугольник Δ_{12i} — главным (для этой тройки).

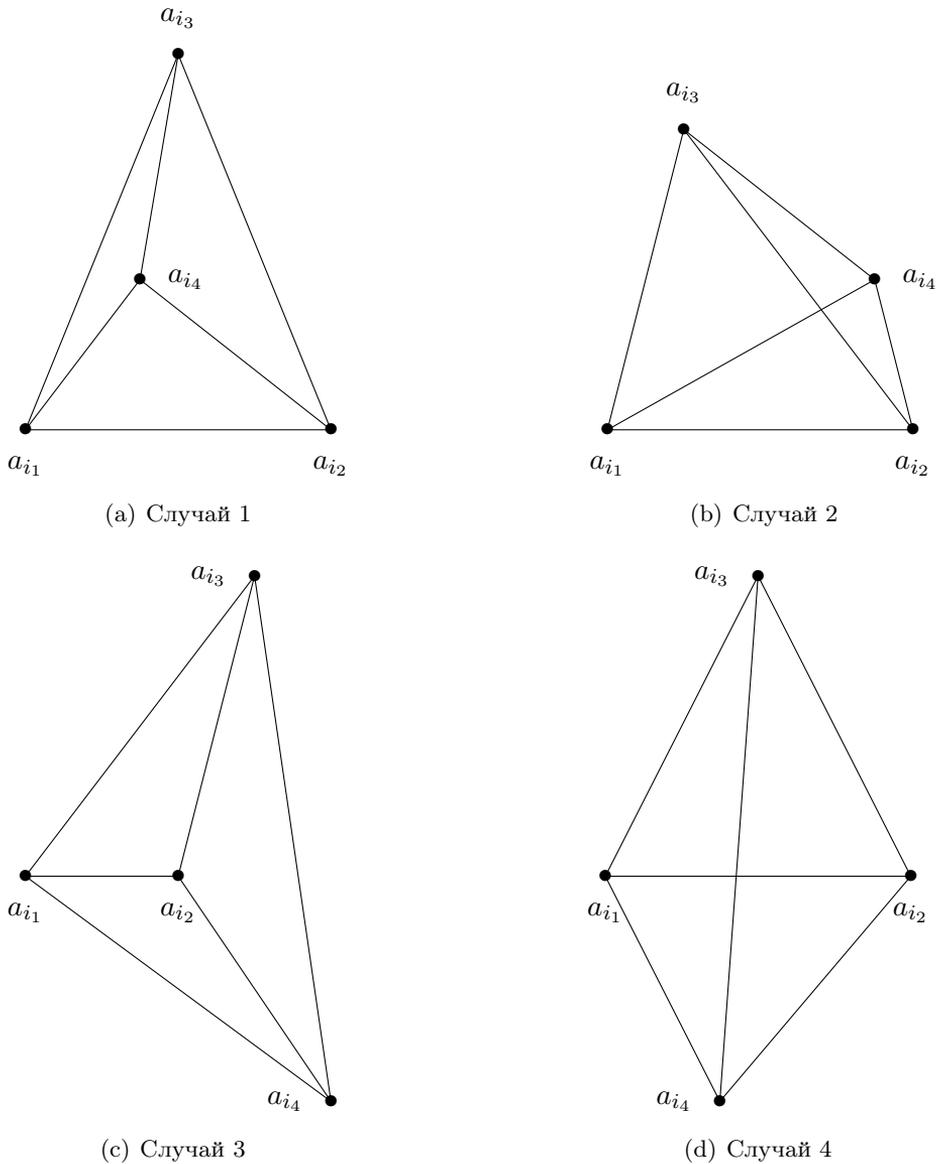


Рис. 3. К доказательству леммы 1 .

Заметим, что для троек первого типа выполнено равенство $S(\Delta_i^{(m)}) = S(\Delta^*) + S(\Delta') + S(\Delta'')$, а для второго типа — равенство $S(\Delta_i^{(m)}) = -S(\Delta^*) + S(\Delta') + S(\Delta'')$ где Δ^* — центральный треугольник, $\Delta_i^{(m)}$ — главный треугольник тройки, а Δ', Δ'' — два оставшихся

Это свойство троек, является, в некотором роде, характеристическим, как показано в следующей лемме.

Лемма 2. Если треугольники $\Delta^{(m)}, \Delta', \Delta'' \in \mathfrak{T}(A) \setminus \{\Delta^*\}$ удовлетворяют соотношению $S(\Delta^{(m)}) = \pm S(\Delta^*) + S(\Delta') + S(\Delta'')$, то найдется $a_i \in A^*$, для которой $\tau^*(a_i) = \{\Delta_i^{(m)}, \Delta', \Delta''\} \in \mathfrak{P}(A)$, т.е. эта тройка треугольников является соответствующей.

Доказательство. Очевидно вытекает из свойств 3 и 4 множества треугольников общего положения.

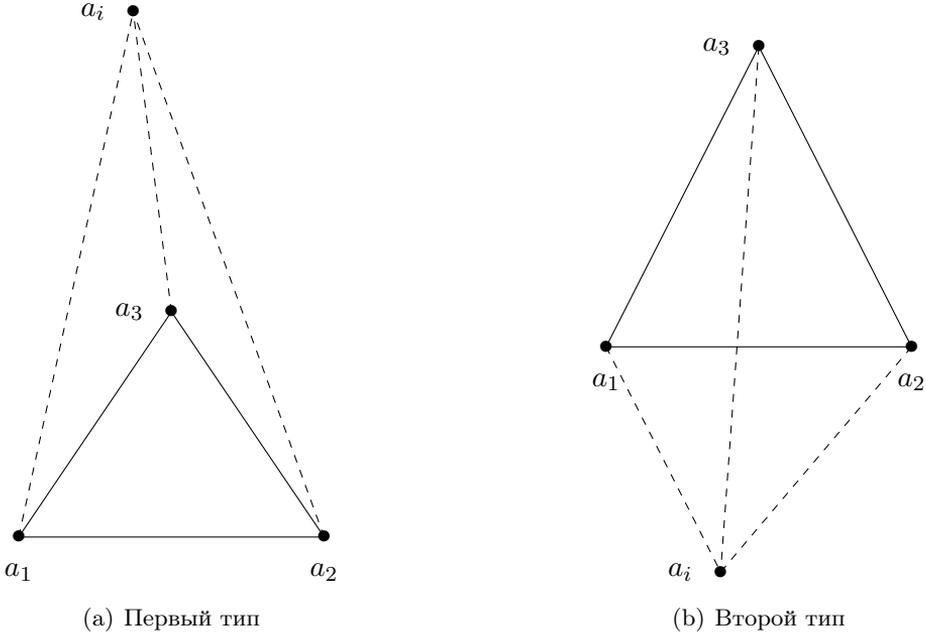


Рис. 4. Примеры соответствующих троек.

Введем отношение смежности на множестве $\mathcal{T}^*(A)$. Будем называть смежными треугольники, не входящие в одну тройку, у которых имеется общая сторона. Заметим, что если треугольники не входят в одну тройку, то общая сторона может быть только одной из сторон центрального треугольника Δ^* . Таким образом, это отношение разбивает множество $\mathcal{T}^*(A)$ на три класса эквивалентности.

Замечание 3. Свойство «иметь общую сторону» легко проверить, используя лемму 1.

Лемма 3. Построим прямые a_1a_2, a_2a_3, a_1a_3 . Они разобьют плоскость на 6 областей (исключая центральный треугольник). Для каждой соответствующей тройки T_i по типу и классу эквивалентности главного треугольника однозначно определяется, в какой из областей расположена точка a_i (см. рис. 5).

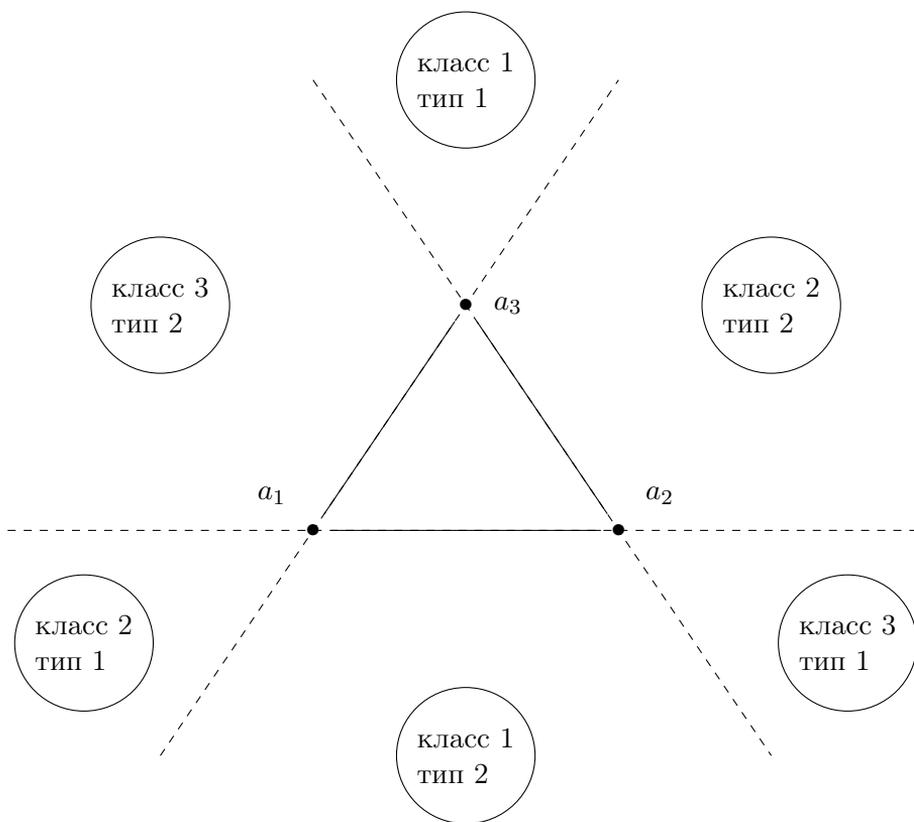


Рис. 5. Положение вершины a_i в зависимости от типа и класса.

Доказательство. Например, рассмотрим тройку первого типа, не ограничивая общности можно считать, что главный треугольник прилегает к стороне a_1a_2 . Из равенства

$$S(\Delta_{12i}) = S(\Delta_{123}) + S(\Delta_{13i}) + S(\Delta_{23i})$$

вытекает, что этот треугольник полностью содержит Δ_{123} , следовательно, его третья вершина a_i содержится в угле, образованном лучами $[a_1; a_3[$ и $[a_2; a_3[$ (см. рис. 5). Остальные случаи рассматриваются аналогично. \square

Замечание 4. Предполагается, что нумерация классов эквивалентности соответствует нумерации сторон: a_1a_2 — первый класс, a_2a_3 — второй, a_3a_1 — третий.

Замечание 5. Здесь и далее в статье предполагается, что $\mathfrak{T}(A)$ — множество треугольников общего положения.

Будем обозначать $\mathfrak{T}(A)$ множество всех треугольников, образованных точками множества A . Площади треугольников различны, обозначим $\Delta_{\min}, \Delta_{\max} \in \mathfrak{T}(A)$ треугольники наибольшей и наименьшей площади, соответственно. Индексы точек, соответствующих этим треугольникам будем обозначать $(i_{\min}, j_{\min}, k_{\min})$ и $(i_{\max}, j_{\max}, k_{\max})$. Обозначим $\rho^* = \frac{S(\Delta_{\max})}{S(\Delta_{\min})}$. Аффинным преобразованием отобразим Δ_{\min} в треугольник единичной площади и будем в дальнейшем полагать $S_{\min} = 1, S_{\max} = \rho^*$. Докажем (в этих предположениях) следующие вспомогательные леммы:

Лемма 4. Пусть $\mathfrak{T}(A)$ — множество треугольников общего положения, $\mathcal{Z}(A)$ — стертый код, соответствующий их вершинам. Существует биекция β между множеством треугольников

$$\mathfrak{T}'(A) = \mathfrak{T}(A) \setminus \{\Delta_{\min}, \Delta_{\max}\}$$

и множеством неупорядоченных пар

$$\mathfrak{S}(A) = \left\{ (\rho^{(1)}, \rho^{(2)}) : \rho^{(1)}, \rho^{(2)} \in \mathcal{Z}(A) \setminus \{1, \rho^*\}; \rho^{(1)} \cdot \rho^{(2)} = \rho^* \right\}$$

Доказательство. Построим отображение β следующим образом: Пусть $\Delta = \Delta_{ijk} \in \mathfrak{T}'(A)$ — произвольный треугольник, очевидно,

$$\frac{S(\Delta_{\max})}{S(\Delta)} \cdot \frac{S(\Delta)}{S(\Delta_{\min})} = \frac{S(\Delta_{\max})}{S(\Delta_{\min})} = \rho^*$$

и можно взять в качестве $\beta(\Delta)$ пару:

$$(\rho^{(1)}, \rho^{(2)}) = \left(\frac{S(\Delta_{\max})}{S(\Delta)} = \rho_{i_{\max}j_{\max}k_{\max},ijk}, \frac{S(\Delta)}{S(\Delta_{\min})} = \rho_{ijk,i_{\min}j_{\min}k_{\min}} \right).$$

Покажем, что отображение β является инъекцией. Предположим обратное: $\beta(\Delta) = \beta(\Delta') = (\rho^{(1)}, \rho^{(2)})$ для некоторых $\Delta \neq \Delta'$. Но это противоречит свойству 2 треугольников общего положения — для разных треугольников получаются одинаковые коды.

Покажем, что отображение β является сюръекцией. Предположим, что найдется пара $(\rho^{(1)}, \rho^{(2)})$, не принадлежащая образу $\beta(\mathfrak{T}'(A))$, такая, что $\rho^{(1)} \cdot \rho^{(2)} = \rho^*$. По свойству 5 это означает, что для некоторых треугольников $\Delta_1, \Delta_2, \Delta_3$ выполнено равенство $\frac{S(\Delta_1)}{S(\Delta_2)} \cdot \frac{S(\Delta_2)}{S(\Delta_3)} = \rho^*$, т.е. $\frac{S(\Delta_1)}{S(\Delta_3)} = \frac{S(\Delta_{\max})}{S(\Delta_{\min})}$. Из этого следует, что $\Delta_1 = \Delta_{\max}$ и $\Delta_3 = \Delta_{\min}$, поскольку для любой другой пары треугольников отношение площадей меньше ρ^* , а значит $\beta(\Delta_2) = (\rho^{(1)}, \rho^{(2)})$. \square

Лемма 5. Пусть $\mathfrak{T}(A)$ — множество треугольников общего положения. Выберем произвольные $\Delta_{ijk}, \Delta_{pqr} \in \mathfrak{T}'(A)$, $\Delta_{ijk} \neq \Delta_{pqr}$. Обозначим $\beta(\Delta_{ijk}) = (\rho^{(1)}, \rho^{(2)})$, $\beta(\Delta_{pqr}) = (\rho^{(3)}, \rho^{(4)})$, пусть для определенности $S_{ijk} = \rho^{(1)}$. Тогда из чисел $\rho' = \frac{\rho^{(3)}}{\rho^{(1)}}$ и $\rho'' = \frac{\rho^{(4)}}{\rho^{(1)}}$ одно и только одно принадлежит стертому коду $\mathcal{Z}(A)$.

Доказательство. Из условия $\beta(\Delta_{pqr}) = (\rho^{(3)}, \rho^{(4)})$ вытекает, что либо $S(\Delta_{pqr}) = \rho^{(3)}$, либо $S(\Delta_{pqr}) = \rho^{(4)}$. Если $S(\Delta_{pqr}) = \rho^{(3)}$, то в коде должен присутствовать $\rho' = \frac{\rho^{(3)}}{\rho^{(1)}}$, а если $S(\Delta_{pqr}) = \rho^{(4)}$, то $\rho'' = \frac{\rho^{(4)}}{\rho^{(1)}}$. Одновременно два таких элемента присутствовать не могут. Действительно, предположим для определенности, что в исходном изображении $S_{pqr} = \rho^{(3)}$. Тогда

$$\rho'' = \frac{\rho^{(4)}}{\rho^{(1)}} = \frac{\rho^*}{\rho^{(1)}\rho^{(3)}} = \frac{\rho^{(2)}}{\rho^{(3)}},$$

следовательно, $\rho'' \cdot \rho^{(3)} = \rho^{(2)}$, что противоречит свойству 5. \square

3.1. Алгоритм восстановления изображения по стертому коду

- 1) Выберем в качестве центрального треугольник минимальной площади. Поскольку восстановление происходит с точностью до эквивалентности, то можно назначить минимальному треугольнику любую площадь, например $S(\Delta_{\min}) = S_{\min} = 1$ для определенности. Тогда площадь наибольшего треугольника определена однозначно $S(\Delta_{\max}) = S_{\max} = \rho^*$. Заметим, что ρ^* известно, т.к. это максимальный элемент стертого кода.
- 2) Зафиксируем одну из пар $(\rho^{(1)}, \rho^{(2)}) \in \mathfrak{S}(A)$. По лемме 4 ей соответствует треугольник $\Delta_{ijk} = \beta^{-1}(\rho^{(1)}, \rho^{(2)})$. Возможно два варианта

$S_{ijk} = \rho^{(1)}$ или $S_{ijk} = \rho^{(2)}$. В дальнейшем построение идет параллельно для обоих вариантов. В одном из них должно получиться исходное изображение. Для определенности будем считать, что $\frac{S_{ijk}}{S_{min}} = S_{ijk} = \rho^{(1)}$, второй вариант рассматривается аналогично.

- 3) Перебираем все пары $(\rho^{(3)}, \rho^{(4)}) \in \mathfrak{S}(A)$. По лемме 4 в исходном изображении найдется треугольник $\Delta_{pqr} = \beta^{-1}(\rho^{(3)}, \rho^{(4)})$. По лемме 5 либо $\rho' = \frac{\rho^{(3)}}{\rho^{(1)}} \in \mathcal{Z}(A)$, тогда $S(\Delta_{pqr}) = \rho^{(3)}$, либо $\rho'' = \frac{\rho^{(4)}}{\rho^{(1)}} \in \mathcal{Z}(A)$ и тогда $S(\Delta_{pqr}) = \rho^{(4)}$. Повторив эту процедуру для всех пар из $\mathfrak{S}(A)$, получаем список площадей всех треугольников из $\mathfrak{T}'(A)$. А $S(\Delta_{min}), S(\Delta_{max})$ уже известны из первого шага алгоритма.
- 4) Выберем в качестве центрального Δ_{min} (очевидно, не содержащий внутри себя точек из A). По лемме 2 найдем все тройки $\mathfrak{P}(A)$, соответствующие точкам $A' = A \setminus \{a_1, a_2, a_3\}$ (т.е. кроме вершин центрального треугольника). В каждой тройке выделим главный треугольник $\Delta_i^{(m)}$, $i = 4, \dots, n$.
- 5) Разбиваем треугольники $\mathcal{T}^*(A)$ на классы эквивалентности. Треугольники эквивалентны тогда и только тогда, когда а) Они не содержатся в одной тройке; б) Имеют общую сторону (что можно проверить по лемме 1).
- 6) Для каждой тройки из $T_i = \mathfrak{P}(A)$ определим по лемме 5 в какой области находится соответствующая точка. Таким образом, мы получим, в какой полуплоскости относительно прямых b_1b_2 , b_2b_3 и b_3b_1 будет расположена восстановленная точка. Если выбрать восстановленные точки как $b_1(0, 0)$, $b_2(1, 0)$, $(0, 1)$, то это позволяет определить знаки координат (см. рис. 6).
- 7) Зафиксируем на плоскости треугольник $\Delta b_1b_2b_3$ и будем считать, что это образ центрального треугольника Δ_{min} при некотором аффинном преобразовании. Для каждой тройки из $\mathfrak{P}(A)$ определим к каким сторонам центрального треугольника прилегают главный и побочные треугольники. Если известна площадь $S_i = S(\Delta_{12i})$, то расстояние точки b_i до прямой b_1b_2 равно $d_i = 2S_i/|b_1b_2|$. Геометрическим местом таких точек являются две прямые, параллельные b_1b_2 и проходящие на расстоянии d_i от нее. Но, поскольку известно (из предыдущего пункта), в какой полуплоскости находится b_i , то остается только одна прямая. Прделавав аналогичные построения для стороны b_2b_3 , получим вторую прямую, не параллельную первой. Очевидно, что точка их пересечения и есть b_i .

- 8) Находим стертый код восстановленного изображения и сравниваем с $\mathcal{Z}(A)$. Если коды не совпадают (такое возможно в случае, если пошли по неправильной ветке в пункте 2), то отбрасываем результат.

Корректность алгоритма вытекает из доказанных ранее лемм.

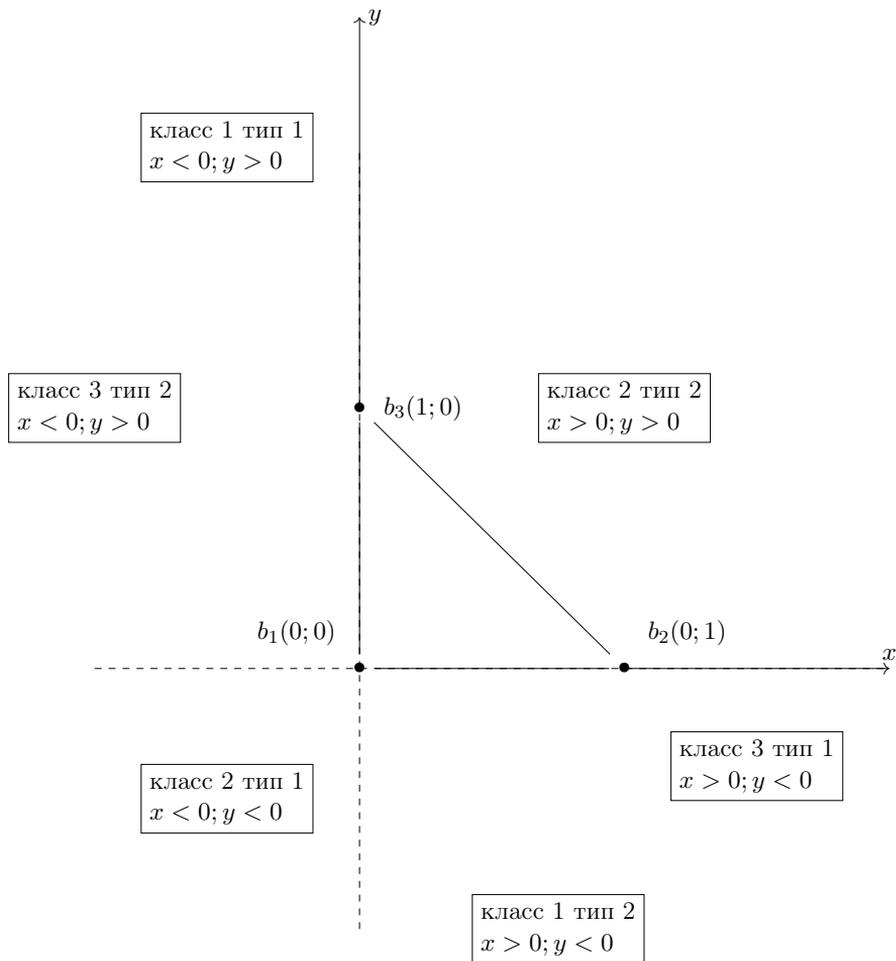


Рис. 6. Определение знака координат по классу и типу тройки (шаг 6).

Шаг	Сложность	Комментарий
1	$O(n^6)$	Перебор элементов стертого кода $ \mathcal{Z}(A) = C_{C_n}^2$
2	$O(1)$	Фиксируем пару из $\mathfrak{S}(A)$
3	$O(n^3)$	Перебор пар из $ \mathfrak{S}(A) = C_n^3$
4	$O(n^9)$	Перебор троек треугольников $C_{C_n}^3$
5	$O(n^7)$	Разбиение троек из $ \mathfrak{P}(A) = n - 3$ на классы эквивалентности ²
6	$O(n)$	Перебор точек и определение положения относительно Δ^*
7	$O(n)$	Перебор точек и восстановление координат
8	$O(n^6)$	Нахождение стертого кода восстановленного изображения и сравнение с исходным кодом

Таблица 2. Оценка алгоритмической сложности

4. Алгоритмическая сложность и эксперименты

Алгоритмическая сложность оценивается в случае, когда все точки множества A обладают целочисленными координатами. В этом случае площади всех треугольников целые или полуцелые, следовательно, все элементы стертого кода являются рациональными. Модель вычислителя предполагает сложность 1 для любого арифметического действия с рациональными числами произвольной точности. Это достаточно адекватно моделирует случай, когда и числитель, и знаменатель дроби помещаются в регистр вычислительного устройства (CPU).

Наиболее трудоемкой является процедура перебора троек треугольников на этапе 4 (см. таблицу 2). Количество операций сравнения равно $2 \cdot C_{C_n}^3 = O(n^9)$, где n — количество точек изображения. Сложность, хотя и является полиномиальной по n , не позволяет проводить алгоритм для большого количества точек (например, $n = 100$).

Алгоритм был реализован в виде скрипта для Matlab (версия R2020b). Краткое описание скрипта:

- Случайным образом порождает координаты n точек;
- Строит стертый код для этих точек;
- Применяет алгоритм из раздела 3.1;
- Проводит замер времени;

n	t (сек.)
8	0.6565
9	2.1726
10	5.7466
11	15.595
12	35.471
13	70.919
14	153.00
15	280.39
16	542.05
17	954.20
18	1489.2
19	2553.6

Таблица 3. Время работы алгоритма

- Проверяет корректность восстановления с точностью до аффинной эквивалентности.

Скрипт выложен в общий доступ и доступен по адресу https://github.com/dvalex/erased_code_recovery.

Тестирование проводилось на компьютере с процессором Intel® Core™ i7-8700 (тактовая частота 3.20GHz), объем оперативной памяти 32GB, ОС Windows 10 (версия 22H2), Проводилось восстановление для $n = 8, 9, \dots, 19$ с замером времени выполнения (1 запуск для каждого n).

Результаты приводятся в таблице 3 и изображены на рис. 7.

Замечание 6. *В алгоритме шаг 4 (перебор троек треугольников) обладает наибольшей вычислительной сложностью. Этот шаг может быть выполнен параллельно, что, впрочем, не снижает асимптотическую сложность.*

5. Заключение: выводы и дальнейшие планы.

Итогом данной работы является создание алгоритма для восстановления изображения по стертому коду за полиномиальное время и доказательство корректности его работы при условии треугольников общего положения. Сделана оценка алгоритмической сложности показано ее соответствие экспериментальным данным.

Возможные направления дальнейшего развития:

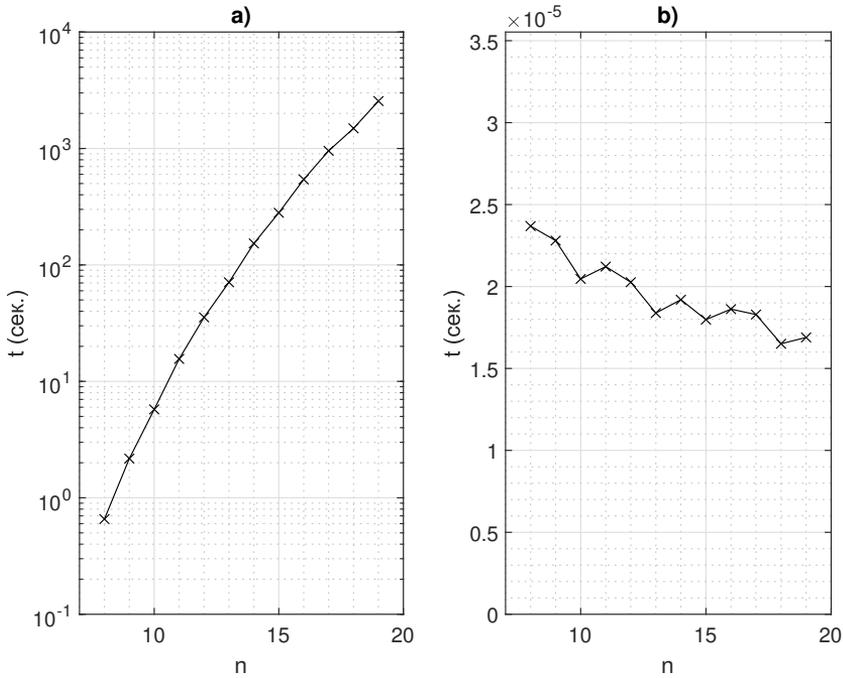


Рис. 7. а) Время выполнения в зависимости от числа точек; б) Время выполнения деленное на $C_{C_n}^3$.

- 1) Ослабить условия на множество треугольников общего положения — сейчас они выглядят не очень естественно и сложными для проверки. Поскольку для проверки требуется перебор четверок треугольников, то алгоритмическая сложность составляет $O(n^{12})$, т.е. превосходит сложность решения самой задачи!
- 2) В случае целочисленных координат, лежащих в определенном диапазоне, оценить вероятность генерации множества треугольников общего положения.
- 3) Сделать построение однозначным. Например, доказать, что у изображения в общем положении нет призраков или вывести достаточные условия, при которых изображение не может иметь призрак.
- 4) Снизить алгоритмическую сложность, которая в текущей версии составляет $O(n^9)$.

- 5) Реализовать алгоритм с использованием рациональных чисел произвольной точности (например, на Python). Сейчас используется встроенный в Matlab тип «double», сравнение таких чисел производится приближенно, с точностью до 10^{-8} .
- 6) Применить аналогичный подход к решению изначальной задачи (про попарные расстояния между точками)

Список литературы

- [1] Агниашвили П.Г., “Однозначность восстановления изображения по его коду в n -мерном случае”, *Интеллектуальные системы.*, **15**:1–4 (2011), 293–332.
- [2] Алексеев Д.В., “Необходимые и достаточные условия существования изображения с заданным кодом”, *Интеллектуальные системы. Теория и приложения (ранее: Интеллектуальные системы по 2014, № 2, ISSN 2075-9460)*, **24**:2 (2020), 55–66.
- [3] Алексеев Д.В., “Кодирование изображений, инвариантное относительно проективных преобразований.”, *Интеллектуальные системы. Теория и приложения.*, **13**:1–4 (2009), 35–40.
- [4] Козлов В.Н., *Способ аутентификации электронного изображения.*, Патент на изобретение № 2779379., Дата государственной регистрации в Государственном реестре изобретений Российской Федерации 06 сентября 2022 года.
- [5] Козлов В.Н., “Доказательность и эвристика при распознавании визуальных образов”, *Интеллектуальные системы.*, **14**:1–4 (2010), 35–52.
- [6] Козлов В.Н., *Элементы математической теории зрительного восприятия.*, Изд-во ЦПИ при мех.-мат. ф-те МГУ, Москва, 2001, 128 с.
- [7] Козлов В.Н., “О кодировании дискретных фигур”, *Дискретная математика*, **8**:6 (1996), 57–61.
- [8] Kozlov V.N., “Image Coding and Recognition and Some Problems of Stereovision”, *Pattern Recognition and Image Analysis*, **7**:4 (1997), 448–466.

**To the question of restoring an image from an erased code
Alekseev D.V.**

The article proposes an algorithm for image recovery by erased code, i.e. the code without information about indices. The algorithm has a polynomial complexity. The definition of general position triangles is given. The algorithm correctness is proved for wide class of points set which form general positions triangles.

Keywords: image code, image encoding, erased code, affine equivalence.

References

- [1] Agniashvili P.G., “Unique image reconstruction from its code in the n -dimensional case”, *Интеллектуальные системы.*, **15**:1–4 (2011), 293–332 (in Russian).
- [2] Alekseev D.V., “Necessary and sufficient conditions for the existence of an image with a given code”, *Intelligent Systems. Theory and Applications (formerly: Intelligent Systems by 2014, №2, ISSN 2075-9460)*, **24**:2 (2020), 55–66 (in Russian).
- [3] Alekseev D.V., “Image encoding invariant with respect to projective transformations.”, *Intelligent Systems.*, **13**:1–4 (2009), 35–40 (in Russian).
- [4] Kozlov V.N., *A way of authenticating a digital image.*, Patent for invention № 2779379., Date of the official registration in the State Register of Inventions of the Russian Federation on September 06, 2022 (in Russian).
- [5] Kozlov V.N., “Evidence and heuristics in visual pattern recognition”, *Intelligent systems*, **14**:1–4 (2010), 35–52 (in Russian).
- [6] Kozlov V.N., *Elements of the mathematical theory of visual perception*, Central Publishing House at the Faculty of Mechanics and Mathematics, Moscow State University, Moscow, 2001 (in Russian), 128 c.
- [7] Kozlov V.N., “About discrete images encoding”, *Discrete Mathematics*, **8**:6 (1996), 57–61 (in Russian).
- [8] Kozlov V.N., “Image Coding and Recognition and Some Problems of Stereovision”, *Pattern Recognition and Image Analysis*, **7**:4 (1997), 448–466.