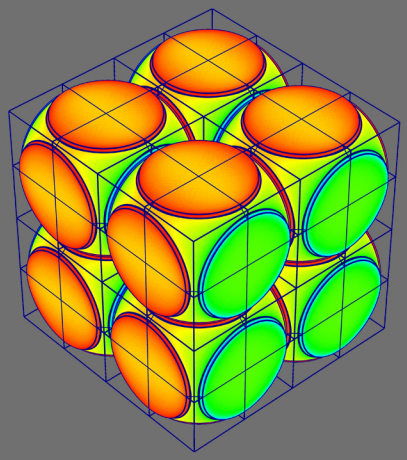
ISSN 2309-3684



Математическое моделирование и численные методы

Валишин А.А., Запривода А.В., Цухло С.С. Моделирование и сравнительный анализ эффективности перцептивных хеш-функций для поиска сегментированных изображений. Математическое моделирование и численные методы, 2024, № 2, с. 46-67.

Источник: https://mmcm.bmstu.ru/articles/340/

Параметры загрузки:
IP: 216.73.216.68
24.11.2026 13:37:29

УДК 519.7 DOI: 10.18698/2309-3684-2024-2-4667

Моделирование и сравнительный анализ эффективности перцептивных хеш-функций для поиска сегментированных изображений

© А.А. Валишин, А.В. Запривода, С.С. Цухло

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

В статье исследуется применение перцептивных хеш-функций для повышения эффективности поиска изображений аэрофотосъемки и спутникового дистанционного зондирования, сегментированных свёрточной нейронной сетью. Анализируются три алгоритма хеширования. Первый алгоритм основан на использовании низкочастотного фильтра и направлен на уменьшение детализации изображения с целью выделения наиболее устойчивых признаков изображения. Второй алгоритм использует двумерное дискретно-косинусное преобразование для создания хеша изображения. Третий алгоритм основан на преобразовании Радона, которое позволяет извлечь информацию о направлениях линий на изображении, а также обеспечить максимальную инвариантность к преобразованию на вращение входного изображения. В статье проводится тестирование данных алгоритмов, включающее анализ их инвариантности к преобразованиям на вращение, масштабирование и сдвиг исходного изображения. Результаты тестирования показывают, что алгоритм, основанный на преобразовании Радона, проявляет хорошую инвариантность к вращению, однако он чувствителен к качеству сегментации, что может привести к частым коллизиям при поиске схожих изображений. Более стабильными и имеющими меньший разброс значений оказались алгоритмы с использованием двумерного дискретно-косинусного преобразования и алгоритм с применением низкочастотного фильтра. Однако следует отметить, что алгоритмы, использующие низкочастотный фильтр и двумерное дискретно-косинусное преобразование, могут оказаться неприменимыми в случае повернутых изображений. По результатам анализа и сравнения производительности алгоритмов, рекомендуется отдать предпочтение либо второму, либо третьему алгоритму, т.к. каждый из них имеет свои достоинства и недостатки и решение о применении конкретного алгоритма в задаче поиска максимально схожего изображения, необходимо учитывать конкретные условия и ограничения задачи, а также требования к качеству сравнения изображений.

Ключевые слова: *машинное зрение, перцептивное хеширование, семантическая сегментация, дискретно-косинусное преобразование, преобразование Радона*

Введение. При обработке изображений различного происхождения (фотографических, радиолокационных, медицинских, сейсмических и др.) с целью их распознавания и идентификации используются различные приемы и алгоритмы. В частности для обработки радиолокационных изображений широко используются вероятностные методы с применением различных статистических критериев [1,2]. В данной статье мы проанализируем различные алгоритмы с использованием хеш-функций.

Перцептивные хеш-алгоритмы (функции) используются при распознавании изображений для сравнения и различения изображений с целью установления их идентичности или отсутствия таковой. С помощью хеш-алгоритмов можно быстро создавать (генерировать) уникальное и компактное представление изображения в виде перцептивного хеша. Перцептивный хеш — это способ представления изображения в виде короткого битового числа фиксированной длины. Он основан на анализе визуальных характеристик изображения, таких как текстуры, контрастность, яркость и расположение основных объектов на изображении. Характеристики изображения используются для создания (генерации) индивидуального отпечатка изображения — хеша, и эти хеши можно сравнивать друг с другом. Отображение первичного изображения в хеш практически однозначно. Очень редко, но возможны коллизии, когда хеши совпадают, а первичные изображения различны. Как правило, совпадающие хеши указывают на то, что исходные изображения идентичны. Сравнивая хеши, можно определить, являются ли два изображения идентичными или разными. Перцептивные хеши инвариантны относительно некоторых преобразований исходного изображения, таких как изменение размеров изображения, изменение его угла поворота, изменение цветовой палитры, незначительное внесение изменений, зашумление изображения. Исходные изображения можно изменять в размере, можно менять соотношение сторон, слегка менять цветовые характеристики (яркость, контраст и т.д.), но они все равно будут иметь одинаковый хеш, т.е., как говорят, они совпадают по хешу. Перцептивные хеши можно сравнивать между собой и делать вывод о степени различия двух исходных изображений.

Используя перцептивные хеш-функции, можно значительно сократить время поиска, необходимое для определения привязки локального изображения к общей предварительно сегментированной карте.

Постановка задачи. В статье рассматривается следующая задача: имеются сегментированные нейросетью снимки со спутника, которые являются картой местности, и отдельное изображение, полученное фотокамерой, и сегментированное такой же нейросетью. Рассматривается только двумерный случай. Сегментация изображений делается для того, чтобы однозначно определить принадлежность каждого пикселя изображения к определенному классу объектов: земля, пастбище, развитое пространство, дорога, дерево, водоем, сельскохозяйственная земля, здание. Каждому пикселю присваивается одно из восьми числовых значений, соответствующих определенному классу. Такой подход позволяет четко определять объекты на изображении вне зависимости от

времени суток и времени года. Необходимо найти максимально схожее с изображением из фотокамеры изображение со спутника. Предполагается, что изображения имеют одинаковые длину и ширину. Для решения этой задачи предлагается использовать алгоритмы перцептивного хеширования.

Мы рассмотрим три алгоритма перцептивного хеширования:

1) алгоритм с применением низкочастотного фильтра и хеша по среднему;

2) алгоритм с использованием дискретно-косинусного преобразования;

3) алгоритм с использованием преобразования Радона.

Цель данного исследования является оценка инвариантности этих алгоритмов к различным преобразованиям сегментированных изображений.

Важно понимать, насколько стабильны перцептивные алгоритмы хеширования на сегментированных изображениях, поскольку они могут быть более восприимчивы к изменениям содержимого изображения из-за процесса сегментации. Изучая поведение этих алгоритмов на сегментированных изображениях, мы можем лучше понять их пригодность для использования в приложениях поиска изображений и поиска сходства.

Описание перцептивных хеш-функций и алгоритмов хеширования. Перцептивные хеш-функции — это математические алгоритмы, которые используются для создания компактного цифрового отпечатка изображения или фрагмента мультимедийного контента, который можно использовать для различных целей, таких как поиск изображений, защита авторских прав и цифровая криминалистика. Цель перцептивных хеш-функций состоит в том, чтобы сгенерировать битовое число (или хеш-код), не изменяющийся при различных преобразованиях изображения.

Перцептивные хеш-функции работают, анализируя визуальные особенности изображения и преобразовывая их в компактное представление, которое можно легко сравнить с другими подобными представлениями. Алгоритм обычно включает в себя последовательность шагов, которые включают в себя следующие этапы [3]:

Трансформация. Чтобы уменьшить различия в разрешении и формате, входное изображение (матрица $M \times N$ байт) подвергается пространственным преобразованиям, такие как преобразование цвета, сглаживание, аффинные преобразования и т. д. или частотные преобразования, такие как дискретное косинусное преобразование (ДКП), дискретное вейвлет преобразование (ДВП) и др.

Извлечение признаков: Система хеширования изображений извлекает характеристики изображения из преобразованного

изображения для создания вектора из L признаков, где $L << M \times N$. Каждый признак может содержать p элементов типа float (чисел с плавающей запятой), что означает, что мы получаем матрицу $L \times p$ чисел с плавающей запятой. Различные перцептивные хеш-функции используют разные методы извлечения признаков, такие как ДКП, ДВП или локальные двоичные шаблоны.

Квантование. На этом этапе извлеченные признаки квантуются и преобразуются в двоичное представление, обычно путем установления пороговых значений признаков или путем применения какого-либо другого типа преобразования. Таким образом мы получаем квантованный промежуточный перцептивный хеш-вектор, который содержит $L \times p$ элементов типа байт.

Сжатие и кодирование: На последнем этапе системы хеширования бинарная промежуточная хеш-строка сжимается и шифруется в короткий перцептивный хеш фиксированного размера l байтов, где $l << L \times p$, который представляет окончательный перцептивный хеш. Полученный хеш-код представляет собой компактный цифровой отпечаток, который можно использовать для сравнения изображений на предмет сходства. Далее рассмотрим некоторые алгоритмы перцептивного хеширования.

Алгоритм с применением низкочастотного (НЧ) фильтра и хеша по среднему. В изображениях высокими частотами называют резкие изменения значения соседствующих пикселей и отображают мелкие детали изображения, а низкими частотами, соответственно, плавное изменение этих значений, которые не отображают деталей, но представляют общую структуру изображения. Изображение высокого разрешения с большим количеством мелких деталей будет содержать большее количество высоких частот по сравнению с уменьшенной копией того же изображения, которое будет содержать в основном низкие частоты. На основе этой идеи можно построить алгоритм вычисления хеш-функции изображения, учитывающий распределение только низких частот в изображении:

Шаг 1. Если изображение цветное, то перевести его в оттенки серого. Как правило, это преобразование проводится по формуле:

$$I_{ij} = 0,3R_{ij} + 0,59G_{ij} + 0,11B_{ij}, \quad (1)$$

где I_{ij} — яркость пикселя серого изображения, R_{ij} — яркость красной составляющей пикселя цветного изображения, G_{ij} — яркость зеленой составляющей пикселя цветного изображения, B_{ij} — яркость синей составляющей пикселя цветного изображения.

В нашем случае изображение сегментировано нейросетью на 8 классов, поэтому каждому пикселю присваивается яркость в

зависимости от принадлежности к определенному классу 0, 36, 72, 108, 144, 180, 216, 255.

Шаг 2. Уменьшить изображение до размера 8×8 пикселей. Изображение линейно интерполируется и дискретизируется для приведения к разрешению 8×8 пикселей. При уменьшении изображения происходит фильтрация высоких частот и выделяется структура изображения.

Шаг 3. Вычислить среднее значение яркости пикселей уменьшенного изображения:

$$m = \frac{1}{64}\sum_{i=0}^{7}\sum_{j=0}^{7} v_{ij}, \tag{2}$$

где v_{ij} — яркость пикселя, m — среднее арифметическое яркости пикселей уменьшенного изображения.

Шаг 4. Провести бинаризацию уменьшенного изображения. Пикселям, яркость которых будет выше средней, присваивается значение 1, иначе присваивается значение 0.

Шаг 5. Сформировать битовое число $\{b_0, b_1, ..., b_{63}\}$, которое будет представлять собой хеш-код:

$$b_k = v_{ij};\ \ k = i\cdot 8 + j;\ \ i, j = \overline{0,7}, \tag{3}$$

где v_{ij} — яркость пикселя, b_k — k-ый бит хеш-кода.

Результатом работы алгоритма будет 64-битный хеш-код.

Алгоритм с использованием дискретно-косинусного преобразования (АДКП). Дискретное косинусное преобразование (ДКП или DCT) используется как метод преобразования при обработке сигналов и для сжатия цифровых изображений. Используется в большинстве носителей: цифровое изображение, цифровое видео, цифровое телевидение, цифровое радио, кодирование речи.

Использование косинуса, а не синуса более экономично и имеет решающее значение для сжатия изображения, т.к. требует меньше функций косинуса [4].

ДКП применяется к изображению путем его разделения на небольшие квадратные блоки пикселей. Затем каждый блок преобразуется независимо с помощью ДКП. Результатом является матрица коэффициентов, которые представляют частотные компоненты блока. Коэффициенты ДКП обычно располагаются в порядке убывания их абсолютной величины. Эту матрицу можно квантовать, т.е. округлять её значения до ближайшего целого значения. Квантование позволяет уменьшить количество полезных коэффициентов, необходимых для восстановления изображения, т.к. многие высокочастотные коэффициенты могут быть отброшены или

уменьшены до нуля без существенного влияния на визуальное качество изображения.

Округленные коэффициенты могут быть преобразованы в пространственную область с использованием обратного ДКП. В результате восстанавливается исходное изображение, которое может иметь некоторую потерю визуального качества из-за процесса квантования. Восстановление данных происходит не только по самим числам, но также по их положению в матрице [4].

Это означает, что после применения ДКП коэффициенты близкие к верхнему левому углу матрицы коэффициентов имеют более высокие значения, чем коэффициенты в остальной части матрицы. Эти коэффициенты большой величины представляют наиболее важные частотные компоненты изображения и содержат большую часть визуальной информации изображения. По сути все информативные компоненты остаются в левой верхней части матрицы, а остальные с большой вероятностью зануляются при квантовании. Т.е. происходит уплотнение информации (энергии) в левой верхней части матрицы коэффициентов.

Дискретное косинусное преобразование (ДКП) — это один из частных случаев дискретного преобразования Фурье[5]. В общем случае преобразование имеет вид

$$\overline{X}(k)=\sum_{n=0}^{N-1}x(n)c(n,k). \tag{4}$$

Ядром преобразования являются различного вида функции косинуса. В литературе известны, восемь вариантов ДКП, из которых четыре являются наиболее употребительными [4]. Все они отличатся друг от друга видом ядра:

- ДКП (DCT) – 1

$$c(n,k)=\cos\left(\frac{nk\pi}{N-1}\right); \tag{5}$$

- ДКП (DCT) – 2

$$c(n,k)=\cos\left(\frac{n(2k+1)\pi}{2N}\right); \tag{6}$$

- ДКП (DCT) – 3

$$c(n,k)=\cos\left(\frac{(2n+1)k\pi}{2N}\right); \tag{7}$$

- ДКП (DCT) – 4

$$c(n,k)=\cos\left(\frac{(2n+1)(2k+1)\pi}{4N}\right). \tag{8}$$

В практических приложениях чаще всего используется ДКП (DCT) – 3. Оно обладает важным свойством «уплотнения энергии». Именно его мы и будем использовать в дальнейшем.

Если $x(n)$ — исходный дискретный массив данных длиной N (т.е. n-мерный вектор, или матрица-столбец), то ДКП этого вектора имеет вид

$$\overline{X}(k)=\sqrt{\frac{2}{N}}\varphi(k)\sum_{n=0}^{N-1}x(n)\cos\frac{(2n+1)k\pi}{2N}, \quad k=\overline{0,N-1}, \tag{9}$$

где индикаторная функция $\varphi(k)$ принимает два значения

$$\varphi(k)=\begin{cases}\dfrac{1}{\sqrt{2}}\ , & k=0\\ 1\ , & k\neq 0\end{cases} \tag{10}$$

Здесь добавлен нормирующий множитель $\sqrt{\frac{2}{N}}$ и индикаторная функция, чтобы преобразование было ортогональным. Величины

$$\omega(k)=\frac{\pi}{2N}k, \quad k=\overline{0,N-1} \tag{11}$$

называются пространственные частоты. Результатом преобразования является спектр частот. Вариации изображения отражаются в спектре частот. Изображение $\overline{X}(k)$ вектора $x(n)$, т.е. его частотный спектр — это тоже n-мерный вектор, т.е. матрица-столбец. Ядро косинус-преобразования (9) — это квадратная матрица N - го порядка

$$C=\left(c(n,k)\right)=\left(\sqrt{\frac{2}{N}}\varphi(k)\cos\frac{(2n+1)k\pi}{2N}\right), \quad n,k=\overline{0,N-1}. \tag{12}$$

Распишем ее подробно

$$C=\begin{pmatrix} c(0,0) & c(0,1) & c(0,2) & \cdots & c(0,N-1)\\ c(1,0) & c(1,1) & c(1,2) & \cdots & c(1,N-1)\\ c(2,0) & c(2,1) & c(2,2) & \cdots & c(2,N-1)\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ c(N-1,0) & c(N-1,1) & c(N-1,2) & \cdots & c(N-1,N-1)\end{pmatrix} \tag{13}$$

или

$$C=\sqrt{\frac{2}{N}}\begin{pmatrix}\frac{1}{\sqrt{2}} & \cos\frac{\pi}{2N} & \cos\frac{2\pi}{2N} & \cdots & \cos\frac{(N-1)\pi}{2N}\\ \frac{1}{\sqrt{2}} & \cos\frac{3\pi}{2N} & \cos\frac{6\pi}{2N} & \cdots & \cos\frac{3(N-1)\pi}{2N}\\ \frac{1}{\sqrt{2}} & \cos\frac{5\pi}{2N} & \cos\frac{10\pi}{2N} & \cdots & \cos\frac{5(N-1)\pi}{2N}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \frac{1}{\sqrt{2}} & \cos\frac{(2N-1)\pi}{2N} & \cos\frac{(2N-1)2\pi}{2N} & \cdots & \cos\frac{(2N-1)(N-1)\pi}{2N}\end{pmatrix}. \quad (14)$$

Легко проверить, что матрица ДКП (9) не вырожденная и ортогональная. Это означает, что

$$\begin{aligned}&\det C\neq 0,\\ &C^{-1}=C^{T}.\end{aligned} \quad (15)$$

В матричном представлении формула (6) перепишется так

$$\overline{X}=CX. \quad (16)$$

Отсюда легко получить обратное преобразование

$$X=C^{-1}\overline{X}=C^{T}\overline{X}. \quad (17)$$

Мы будем рассматривать фотографические изображения. Фотоснимок для дальнейшей обработки дискретизируется и в цифровом формате представляется в виде матрицы (чаще квадратной размером $N\times N$). Обозначим как $X=\left(x\left(n_1,n_2\right)\right)$ — двумерный исходный массив (матрицу), это исходное изображение, дискретизированное и представленное в цифровом формате как квадратная числовая матрица размером $N\times N$. Через $\overline{X}=\left(\overline{x}(k_1,k_2)\right)$ обозначим изображение исходной матрицы $X=\left(x\left(n_1,n_2\right)\right)$ после применения двумерного дискретного косинус-преобразования. Двумерное дискретное косинусное преобразование (ДКП или DCT) имеет вид

$$\begin{aligned}&\overline{X}(k_1,k_2)=\frac{2}{N}\varphi(k_1)\varphi(k_2)\times\\ &\times\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}x(n_1,n_2)\cos\frac{(2n_1+1)k_1\pi}{2N}\cos\frac{(2n_2+1)k_2\pi}{2N},\\ &k_1,k_2=\overline{0,N-1}.\end{aligned} \quad (18)$$

В матричном виде это будет

$$\begin{aligned} \overline{X} &= C^T X C = C^{-1} X C, \\ X &= C\overline{X}\ C^{-1} = C\overline{X}\ C^T. \end{aligned} \tag{19}$$

Общий недостаток дискретных ортогональных преобразований состоит в их высокой вычислительной сложности. Поэтому используются быстрые алгоритмы выполнения ДКП.

Двумерное дискретное косинус преобразование позволяет обнаруживать ключевые особенности изображения с помощью процесса, называемого «уплотнением энергии». Изображение в цифровом представлении — это числовая матрица (чаще всего квадратная). С помощью преобразования (15), (16) оно преобразуется в спектральное представление — это тоже матрица того же размера.

ДКП можно использовать для вычисления перцептивного хеша для изображения путем анализа спектрального представления изображения и определения его ключевых характеристик.

Алгоритм построения перцептивного хеша с использованием ДКП.

Шаг 1. Если изображение цветное, то перевести его в оттенки серого. По аналогии с шагом 1 алгоритма с применением низкочастотного фильтра.

Шаг 2. Уменьшить изображение до размера 32×32 пикселя. По аналогии с шагом 2 алгоритма с применением низкочастотного фильтра.

Шаг 3. Выполнить дискретное косинусное преобразование для уменьшенного изображения;

Шаг 4. Выделить из преобразованного с помощью ДКП изображения матрицу 8×8 начиная со второй строки и столбца;

Шаг 5. Провести бинаризацию матрицы. По аналогии с шагом 4 алгоритма с применением низкочастотного фильтра.

Шаг 6. Сформировать битовое число. По аналогии с шагом 5 алгоритма с применением низкочастотного фильтра.

Результатом работы алгоритма будет 64-битный хеш-код.

Алгоритм с использованием преобразования Радона (АПР). Преобразование Радона [7] вычисляется путем взятия линейных интегралов двумерного изображения $f(x,y)$ по набору направлений. Множество линейных интегралов по определенному направлению θ называется проекцией. Линейный интеграл от функции $f(x,y)$ вдоль прямой, определяемой направлением θ и расстоянием x' от начала координат в координатах (x',y'), где

$$\begin{aligned} x &= x'\cos(\theta) - y'\sin(\theta), \\ y &= x'\sin(\theta) + y'\cos(\theta) \end{aligned} \tag{20}$$

равен:

$$R_\theta(x') = \int_{-\infty}^{+\infty} f\left(x'\cos(\theta) - y'\sin(\theta), x'\sin(\theta) - y'\cos(\theta)\right)dy'. \quad (21)$$

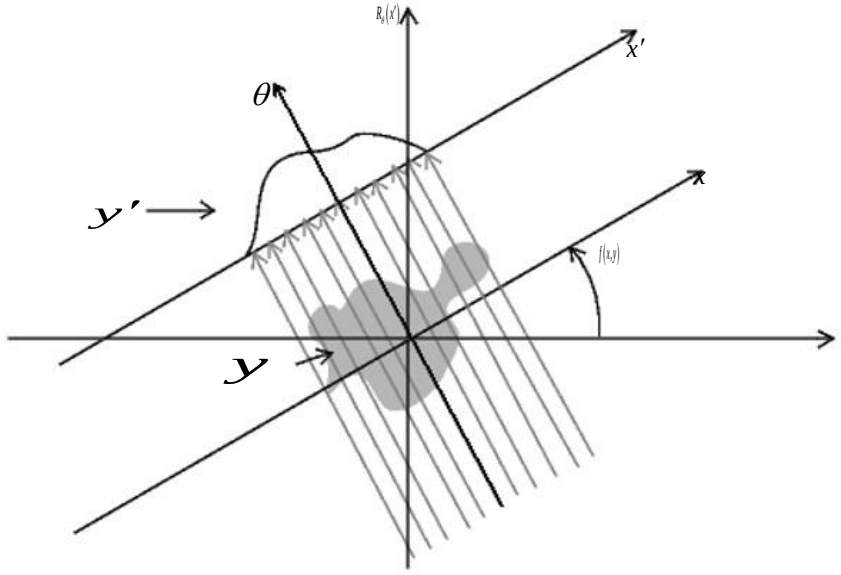


Рис. 1. Графическое представление преобразования Радона: x' — плоскость проекции, $f(x,y)$ — преобразовываемая функция, θ — угол проекции, $R_\theta(x')$ — результат проекции на плоскости x'

Графически преобразование радона изображено на рис. 1. Результаты интегрирования преобразовываются в синограмму — двумерное распределение проекций, где угол проекции расположен по оси абсцисс, а результат линейной проекции — по оси ординат. Если используется несколько параллельных проекций, то синограмма может быть представлена в виде тепловой карты, где по оси абсцисс расположен угол проекции, по оси ординат распределены линии проекции, а результат проекции в каждом угле отображается с помощью цвета как показано на рис 2.

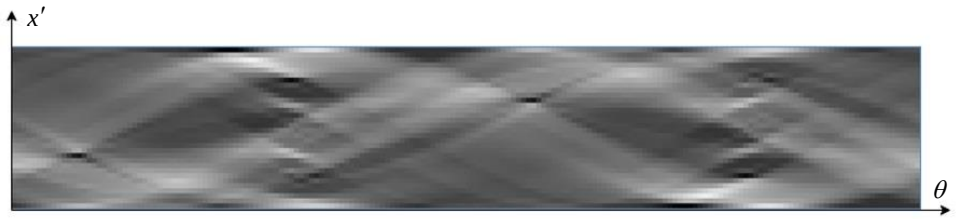


Рис.2. Пример тепловой карты синограммы дискретного преобразования Радона: результаты проекции формируют столбцы, т. е. строки — линии проекции, столбцы — углы проекции

Выражение (21) приводит к двум заметным свойствам [7].

Масштабирование:

$$f(ax, ay) \leftrightarrow \frac{1}{a} R_\theta(ax'),$$

где $a > 0$. Преобразование Радона масштабированного изображения $f(ax,ay)$ пропорционально преобразованию Радона изображения $f(x,y)$ с тем же коэффициентом масштабирования a, т. е.

$$\frac{R_\theta(ax')}{R_\theta(x')} = a.$$

Вращение: если изображение $f(x,y)$ повернуто на ω градусов, преобразование Радона повернутого изображения равно

$$f(x\cos(\omega) - y\sin(\omega), x\sin(\omega) - y\cos(\omega)) \leftrightarrow R_{\theta+\omega}(x'), \qquad (22)$$

т. е. его можно получить циклическим сдвигом коэффициентов преобразования изображения $f(x,y)$ на ω.

Свойства преобразования позволяют получить хорошую инвариантность к атакам вращения, масштабирования и обработки изображений. Чтобы распространить преобразование Радона на дискретные изображения, линейный интеграл по $x\cos(\theta) - y\sin(\theta) = d$ аппроксимируется суммированием пикселей, лежащих в полосе шириной в n пикселей:

$$d - \frac{n}{2} \le x\cos(\theta) - y\sin(\theta) \le d + \frac{n}{2}. \qquad (23)$$

Пусть (x', y') обозначают координаты центрального пикселя изображения, тогда (22) можно переписать следующим образом:

$$-\frac{n}{2} \le (x - x')\cos(\theta) - (y - y')\sin(\theta) \le -\frac{n}{2}. \qquad (24)$$

В этой статье за основу был взят метод дискретного преобразования Радона (ДПР) предложенный в [8]. В этом методе строится синограмма по всей ширине изображения, используя несколько параллельных проекций, ограничивая угол θ полуинтервалом [0, 180).

Метод примечателен тем что использует дисперсию вместо суммы значений пикселей вдоль линейных проекций. Это было сделано по двум причинам:

1) из-за того, что при разных углах проекции суммируется разное

количество пикселей изображения, т.к. изображения квадратные, то на углах 45 и 135 градусов из изображения захватывается наибольшее количество пикселей по каждой полосе, поэтому при проекции на этих углах в синограмме возникают участки с повышенной энергией;

2) было замечено, что дисперсия эффективно захватывает разрывы яркости вдоль линий проекции, на изображении эти разрывы соответствуют краям, ортогональным направлению проекции.

В данный метод мы внесли следующие изменения.

Так как в рассматриваемой задаче на вход подаются квадратные изображения, то для обеспечения большей устойчивости к поворотам изображений, мы можем ограничить область сканирования изображений окружностью:

$$\left(x-x'\right)^2+\left(y-y'\right)^2=\left(N/2\right)^2, \qquad (25)$$

где N — ширина изображения. Это позволит добиться максимальной схожести синограмм при любой разности углов поворота изображений.

Вместо дисперсии будет использоваться среднее арифметическое значений пикселей по следующим причинам.

Во-первых, так как входные изображения сегментированы, они содержат большое количество одинаковых значений и определённое значение несет полезную информацию о том к какому классу принадлежит определенный пиксель. Поэтому нам не столько важно предположительное количество разрывов на изображении, сколько предположительное количество пикселей определенного значения на нем.

Во-вторых, увеличение количества пикселей на линии проекции также не создает областей с повышенной энергией.

В-третьих, вычисление среднего с вычислительной точки зрения более простой процесс по сравнению с дисперсией. Так мы делаем алгоритм быстрее.

Увеличим область определения синограммы до $\theta\in[0,360)$ для достижения периодичности синограммы. В дальнейшем это позволит нам воспользоваться свойством вращения преобразования радона и достигнуть максимальной надежности алгоритма хеширования к повороту изображения.

Теперь отметим, что синограммы двух одинаковых изображений, одно из которых повернуто на угол ω, выходят одинаковыми с тем лишь отличием, что они сдвинуты относительно друг друга на аналогичную величину по оси θ. На рис. 2 можно заметить, что синограмма имеет участки с более высокой и низкой суммарной яркостью пикселей. Тогда, используя свойство вращения

преобразования радона и периодичность синограммы, мы можем определить область с максимальной яркостью и сдвинуть синограмму так, чтобы она начиналась с этой области. Таким образом вращение изображений будет меньше влиять на конечный результат. Этот этап назовем ориентированием синограммы.

Таким образом, используя предложенный метод ДПР, можно построить следующий алгоритм перцептивного хеширования.

Шаг 1. Если изображение цветное, то перевести его в оттенки серого. По аналогии с шагом 1 алгоритма с применением низкочастотного фильтра.

Шаг 2. Применить ДПР к изображению и получить синограмму. ДПР проводится с использованием 32 полос проекций шириной 1/32 ширины изображения. Проекции строятся с разбиением в 180 углов, т. е. $\theta = \{0, 2, 4, ..., 358\}$.

Шаг 3. Провести ориентирование синограммы. Для этого в верхней части синограммы (первые 16 строк) найти такую область размера 5×16, в которой суммарная яркость пикселей будет максимальна. Затем сдвинуть синограмму так чтобы она начиналась с этой области. Нижняя часть не учитывается поскольку она симметрична верхней, но сдвинута по фазе на 180 градусов.

Шаг 4. К каждой строке синограммы применить ДКП для уменьшения количества полезных коэффициентов синограммы чтобы получить сжатую синограмму

Шаг 5. Из сжатой синограммы вырезать область 8×32 начиная с 1-ой строки 2-го столбца и провести её бинаризацию. По аналогии с шагом 4 алгоритма с применением низкочастотного фильтра.

Шаг 6. Сформировать битовое число $\{b_0, b_1, ..., b_{255}\}$, которое будет представлять собой хеш-код:

$$b_k = v_{ij};\ k = i \cdot 8 + j;\ i = \overline{0,31};\ j = \overline{0,7}, \tag{26}$$

где v_{ij} — яркость пикселя.

Результатом работы алгоритма будет 256-битный хеш-код.

Тестирование алгоритмов. Для тестирования алгоритмов использовались изображения из набора данных «OpenEarthMap» [9]. Набор предназначен для обучения нейронных сетей для задачи сегментации спутниковых фотоснимков. В нем содержится около 5000 аэрофотоснимков и спутниковых снимков 97 регионов из 44 стран на 6 континентах. Каждый снимок (рис. 3, *а*) сопровождается сегментированным изображением или разметкой (рис. 3, *б*). Маркировка сегментов представлена на рис. 3, *г*.

Разделим количество входных данных на 2 группы:

1) «чистые данные» (пример на рис. 3, *б*) — это сегментированные изображения, которые являются разметкой из

набора «OpenEarthMap»;

2) «реальные данные» (пример на рис. 3, *в*) — это сегментированные изображения, которые были получены нейросетью, построенной по архитектуре U-Net [10].

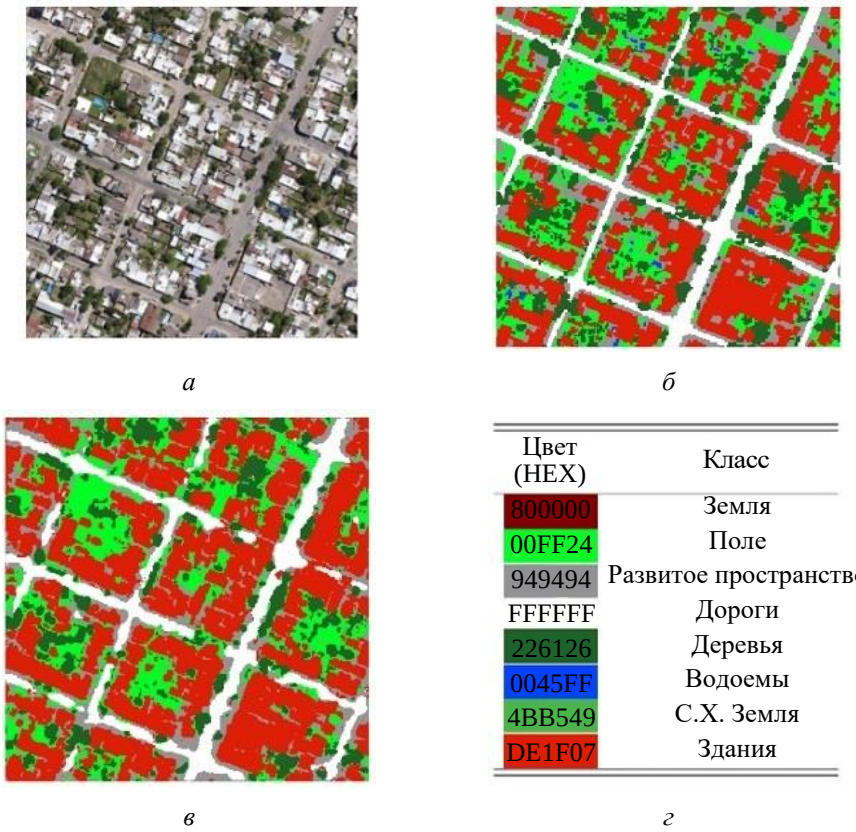


Рис. 3. Изображения из набора данных «OpenEarthMap»:
а — исходное изображение спутникового снимка;
б — разметка сегментов из набора «OpenEarthMap»;
в — результат сегментации свёрточной нейронной сетью;
д — цветовая маркировка сегментов

Необходимо отметить, что сегментация нейросетью не всегда выдает точный результат, что может отразиться на результатах тестирования. Именно поэтому тестирования будут проводиться на двух видах входных данных.

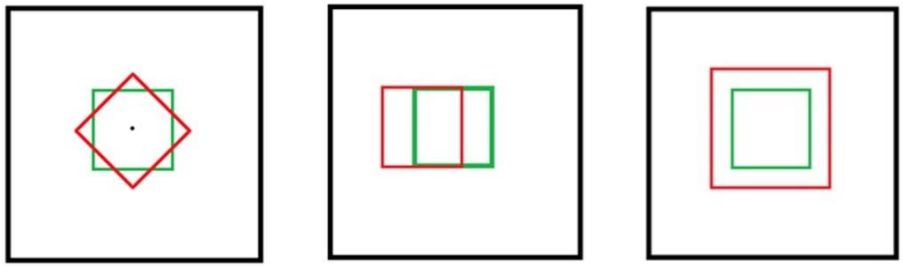
Каждый тип тестирования предполагает наличие оригинального и измененного определенным образом изображений, которые являются частью более крупного изображения. Если тестирование поводится на «чистых» данных, то эти изображения вырезаются из данных разметки набора «OpenEarthMap». Если тестирование поводится на «реальных» данных, то эти изображения вырезаются из снимков набора «OpenEarthMap», а затем сегментируются нейросетью.

Оба изображения хешируются тестируемым алгоритмом, затем полученные хеши изображений сравниваются по расстоянию Хэмминга. Будут проводится тесты с использованием следующих преобразований.

Поворот изображения на угол n*:* изображение вырезается из той же области, что и оригинальное и при помощи аффинных преобразований поворачивается на угол n вокруг центра изображения (рис. 4, *а*).

Сдвиг изображения на n *пикселей:* Область из которого вырезается измененное изображение сдвинута относительно области оригинального изображения на n пикселей по горизонтали (рис. 4, *б*).

Масштабирование изображения: Область из которого вырезается измененное изображение сдвинута относительно области оригинального изображения на n пикселей по вертикали и горизонтали, а также имеет размер по вертикали и горизонтали на $2n$ больше чем у оригинального (рис. 4, *в*).



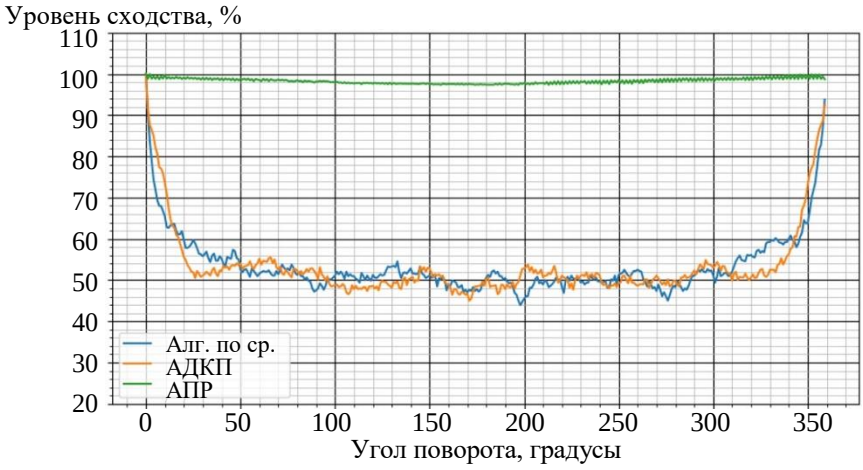
а *б* *в*

Рис. 4. Визуализация областей оригинального (зеленого) и измененного (красного) изображений при тестировании:
а — на поворот изображения; *б* — на сдвиг изображения;
в — на масштабирование изображения

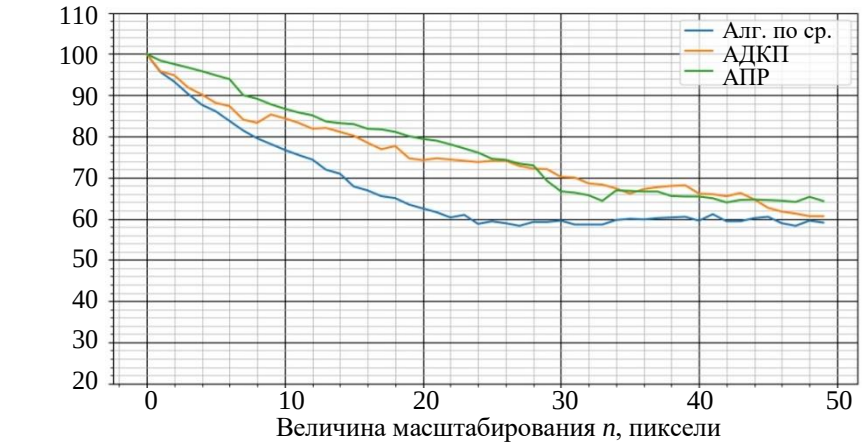
Тестирования проводились с использованием 10-ти снимков/разметок из «OpenEarthMap». Перечислим названия примеров из которых была сформирована выборка: abancay_22; abancay_46; kyoto_4; kyoto_66; melbourne_23; melbourne_49; tyrolw_7; tyrolw_37; vienna_24; vienna_31.

Результаты моделирования. Далее приведены графики, отражающие на сколько процентов согласно определенному алгоритму совпадают изображения при определенном параметре преобразования. Для краткости записи введены сокращения: АДКП — алгоритмом с использованием дискретно-косинусного преобразования, АПР — Алгоритм с использованием Преобразования Радона.

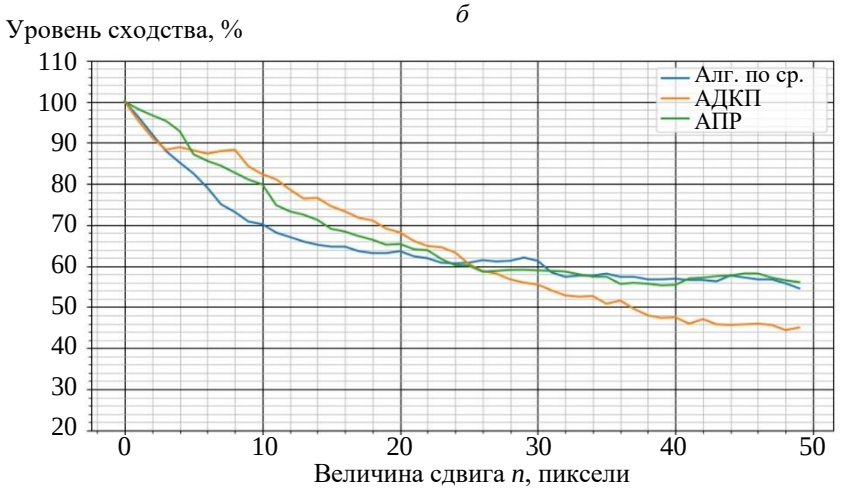
На рис. 5 приведены результаты тестирования на чистых данных, на рис. 6 — на реальных данных.



а

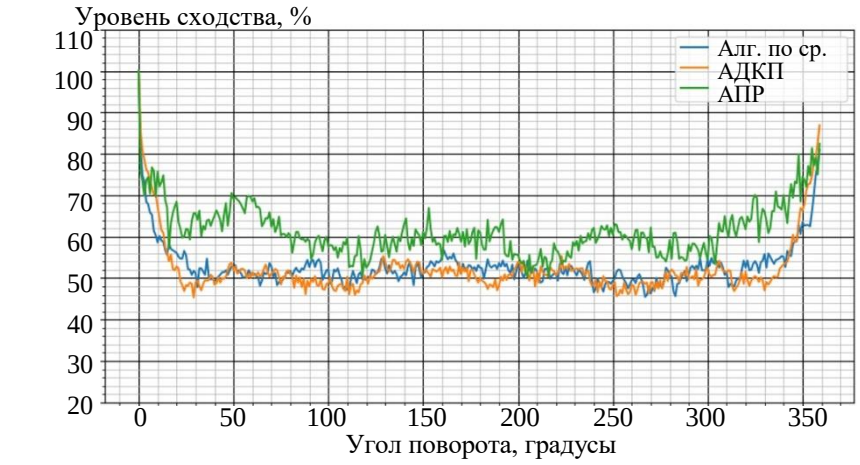


б

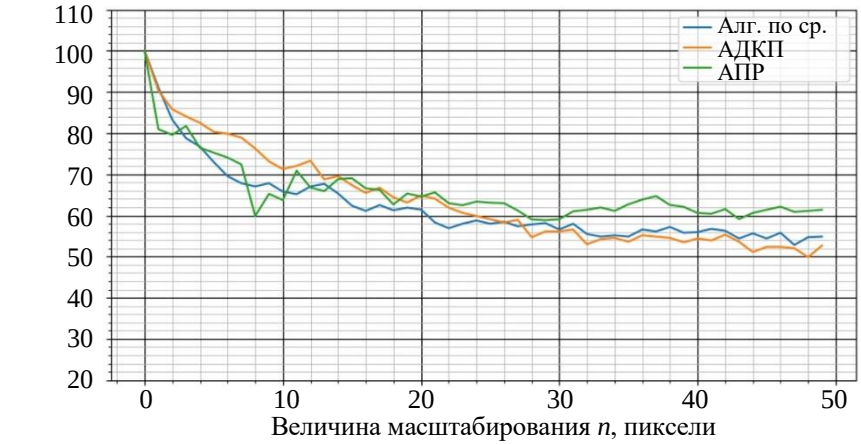


в

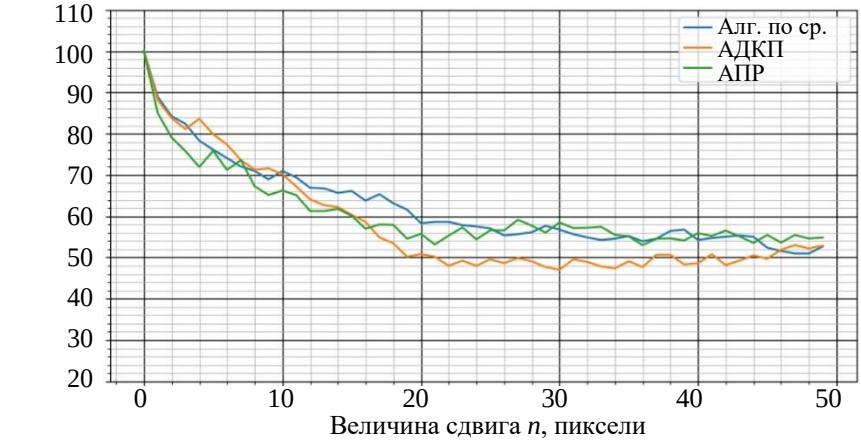
Рис. 5. Результаты тестирования на чистых данных (отображены средние значения по всем тестам): *а* — вращение; *б* — масштабирование; *в* — сдвиг



а



б



в

Рис. 6. Результаты тестирования на реальных данных (отображены средние значения по всем тестам): *а* — вращение; *б* — масштабирование; *в* — сдвиг

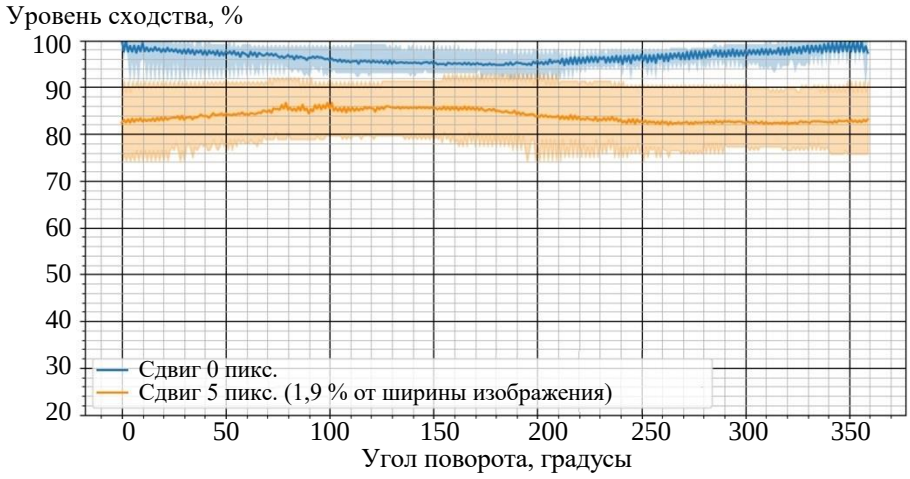
Анализируя графики (рис. 5 — 6) можно сделать следующие выводы:

1) АПР, в отличие от других алгоритмов, оказался почти полностью инвариантным к преобразованиям на вращение (рис. 5);

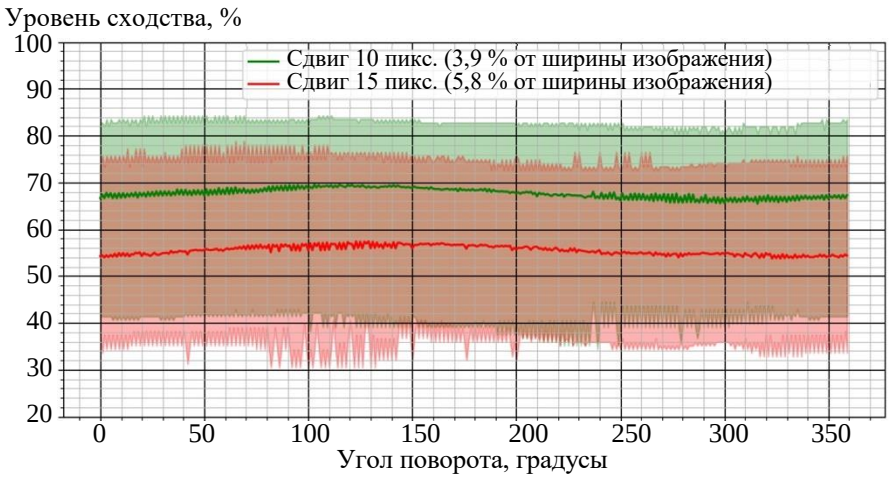
2) АПР оказался очень чувствительным к качеству сегментации, что заметно при сравнении графиков вращения на рис. 5 и 6;

3) все алгоритмы выдают примерно одинаковые результаты при тестировании на сдвиг и масштабирование.

Теперь для АПР проведем смешанные тесты на вращение и сдвиг одновременно, т. е. центр области изменяемого изображения будет сдвинут относительно центра области оригинального на некоторое значение (рис. 7).



а



б

Рис. 7. Результат смешанного тестирования (вращение + сдвиг) на чистых данных; линиями отображены средние значения по всем тестам, полупрозрачными областями разброс значений от минимальных до максимальных: *а* — сдвиг 0 пикс., сдвиг 5 пикс.; *б* — сдвиг 10 пикс., сдвиг 15 пикс.

Из результатов на рис. 7 видно, что инвариантность к вращению в целом остается, однако можно заметить, что область разброса значений увеличивается. Это говорит о том, что АПР теряет стабильность по мере удаления центра области изменяемого изображения от центра области оригинального.

Далее Алгоритм с применением низкочастотного фильтра рассматриваться не будет, т.к. его показатели аналогичны или хуже, чем показатели АДКП. Сравним АПР с АДКП по разбросу результатов (рис. 8).

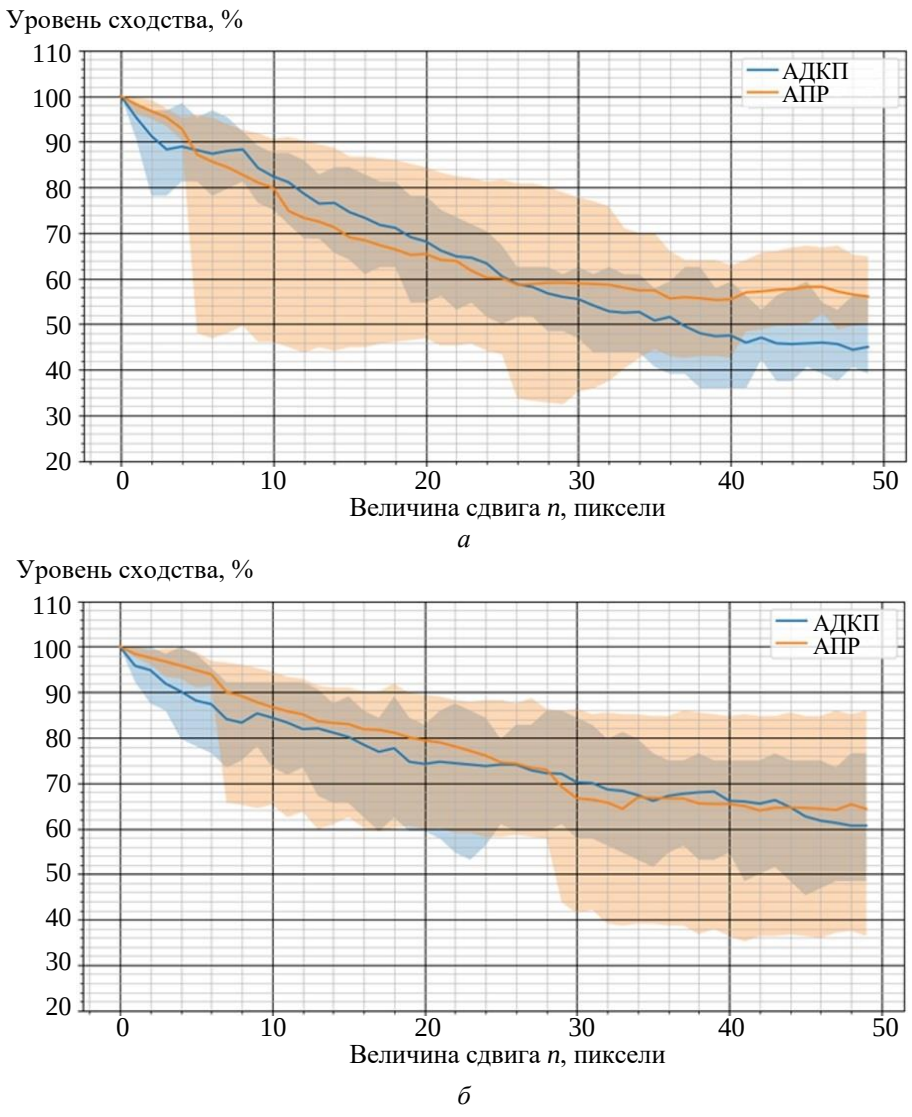


Рис. 8. Результаты тестирования сдвигом и масштабированием на чистых данных; линиями отображены средние значения по всем тестам, полупрозрачными областями разброс значений от минимальных до максимальных:
а — сдвиг; *б* — масштабирование

Из рис. 8 можно заметить, что АПР имеет больший разброс значений по сравнению с АДКП, это означает, что АПР менее стабилен и может чаще чем АДКП выдавать ложные результаты сходстве или различии изображений.

Выводы. Из результатов моделирования и анализа результатов можно сделать следующие выводы.

1) АПР показал хорошую инвариантность к преобразованиям на вращение, но оказался чувствительным к качеству сегментации, что может привести к ложным результатам сходства изображений. Кроме того, АПР теряет стабильность по мере удаления центра области изменяемого изображения от центра области оригинального.

2) АДКП и Алгоритм с применением НЧ фильтра оказались более стабильными и имеют меньший разброс значений по сравнению с АПР, что означает, что эти алгоритмы менее склонны к ошибкам при сравнении изображений. Однако, эти алгоритмы могут оказаться неприменимыми, если по условию задачи одинаковые изображения повернуты на определенный угол. Выбирая из двух алгоритмов, большее предпочтение нужно отдать АДКП, т.к. его показатели оказались лучше.

Таким образом, наиболее применимыми можно считать алгоритмы АПР и АДКП. Для выбора конкретного алгоритма в задаче поиска максимально схожего изображения, необходимо учитывать конкретные условия и ограничения задачи, а также требования к качеству сравнения изображений.

ЛИТЕРАТУРА

[1] Достовалова А.М. Моделирование локально-однородных радиолокационных изображений при использовании различных статистических критериев. *Математическое моделирование и численные методы*, 2021, № 4, с. 103–120.

[2] Достовалова А.М. Применение модели смеси вероятностных распределений в обработке радио локационных изображений. *Математическое моделирование и численные методы*, 2020, № 3, с. 117–130.

[3] Hadmi A., Puech W., Es Said B.A., Ait A. P*erceptual Image Hashing. Watermarking — Volume 2*. InTech, 2012. DOI: 10.5772/37435

[4] Сэломон Д. *Сжатие данных, изображений и звука*. Москва, Техносфера, 2004, 368 с.

[5] Марчевский И.К., Пузикова В.В. Анализ эффективности итерационных методов решения систем линейных алгебраических уравнений. *Математическое моделирование и численные методы*, 2014, № 4, с. 37–52.

[6] Loeffler C., Ligtenberg A., Moschytz G.S. Practical fast 1-D DCT algorithms with 11 multiplications. *International Conference on Acoustics, Speech, and Signal Processing*, 1989, vol. 2, pp. 988–991. DOI: 10.1109/ICASSP.1989.266596

[7] Nguyen D.Q., Weng L., Preneel B. Radon Transform-Based Secure Image Hashing. *CMS 2011: Communications and Multimedia Security*, 2011, p. 186–193. DOI: 10.1007/978-3-642-24712-5_1

[8] Standaert F.-X., Lefebvre E., Rouvroy G., Macq B., Quisquater J.-J., Legat J.-D., Practical evaluation of a radial soft hash algorithm. *International Conference on Information Technology: Coding and Computing (ITCC'05)*, 2005, vol. 2, pp. 89–94. DOI: 10.1109/ITCC.2005.229

[9] Xia J., Yokoya N., Adriano B., Broni-Bediako C. *OpenEarthMap: A Benchmark Dataset for Global High-Resolution Land Cover Mapping*. arXiv.org, 2022. DOI: https://arxiv.org/abs/2210.10732

[10] Ronneberger O., Fischer P., Brox T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.org, 2015. URL: https://arxiv.org/abs/2210.10732

Статья поступила в редакцию 18.11.2023

Ссылку на эту статью просим оформлять следующим образом:

Валишин А.А., Запривода А.В., Цухло С.С. Моделирование и сравнительный анализ эффективности перцептивных хеш-функций для поиска сегментированных изображений. *Математическое моделирование и численные методы*, 2024, № 2, с. 46-67.

Валишин Анатолий Анатольевич — д-р. физ.-мат. наук, профессор кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана. e-mail: enf@mail.ru

Запривода Алексей Васильевич — студент кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана.

Цухло Сергей Сергеевич — студент кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана.

Modeling and efficiency analysis of perceptual hash functions for segmented image search

© A.A. Valishin, A.V. Zaprivoda, S.S. Tsukhlo

Bauman Moscow State Technical University, Moscow, 105005, Russia

This research paper explores the use of perceptual hash functions to improve the retrieval efficiency of aerial photography and satellite remote sensing images segmented by a convolutional neural network. This paper describes three hashing algorithms. The first algorithm is based on the use of a low-pass filter and is aimed at reducing image detail in order to highlight the most stable image features. The second algorithm uses a two-dimensional discrete cosine transform to create an image hash. The third algorithm is based on the Radon transform, which allows you to extract information about the directions of lines in the image, as well as provide maximum invariance to the rotation transformation of the input image. The article also tests these algorithms, including analysis of their invariance to transformations for rotation, scaling and shifting of the source image. Test results show that the algorithm based on the Radon transform exhibits good rotation invariance, but is sensitive to the quality of segmentation, which can lead to frequent collisions when searching for similar images. Algorithms using a two-dimensional discrete cosine transform and an algorithm using a low-pass filter turned out to be more stable and have a smaller spread of values. However, it should be noted that algorithms using a low-pass filter and 2D discrete cosine transform may not be applicable to rotated images. Based on the results of analysis and comparison of the

performance of the algorithms, it is recommended to give preference to either the second or third algorithm, because each of them has its own advantages and disadvantages, and the decision to use a specific algorithm in the task of finding the most similar image must take into account the specific conditions and limitations of the problem, as well as the requirements for the quality of image comparison.

Keywords: *computer vision, perceptual hashing, semantic segmentation, discrete cosine transform, Radon transform*

REFERENCES

[1] Dostavalova A.M. Simulation of locally homogeneous radar images using different statistical criteria. *Mathematical Modeling and Computational Methods*, 2021, no. 4, pp. 103–120.

[2] Dostavalova A.M. Application of a finite mixture model in radar image processing. *Mathematical Modeling and Computational Methods*, 2020, no. 3, pp. 117–130.

[3] Hadmi A., Puech W., Es Said B.A., Ait A. P*erceptual Image Hashing. Watermarking — Volume 2*. InTech, 2012. DOI: 10.5772/37435

[4] Salomon D. *Szhatie dannyh, izobrazhenij i zvuka* [Compression of data, images and sound]. Moscow, Technosphere Publ., 2004, 368 p.

[5] Marchevsky I.K., Puzikova V.V. Performance analysis of iterative methods of combined linear algebraic equations solution. *Mathematical Modeling and Computational Methods*, 2014, no. 4, pp. 37–52.

[6] Loeffler C., Ligtenberg A., Moschytz G.S. Practical fast 1-D DCT algorithms with 11 multiplications. *International Conference on Acoustics, Speech, and Signal Processing*, 1989, vol. 2, pp. 988–991. DOI: 10.1109/ICASSP.1989.266596

[7] Nguyen D.Q., Weng L., Preneel B. Radon Transform-Based Secure Image Hashing. *CMS 2011: Communications and Multimedia Security*, 2011, p. 186–193. DOI: 10.1007/978-3-642-24712-5_1

[8] Standaert F.-X., Lefebvre E., Rouvroy G., Macq B., Quisquater J.-J., Legat J.-D., Practical evaluation of a radial soft hash algorithm. *International Conference on Information Technology: Coding and Computing (ITCC'05)*, 2005, vol. 2, pp. 89–94. DOI: 10.1109/ITCC.2005.229

[9] Xia J., Yokoya N., Adriano B., Broni-Bediako C. *OpenEarthMap: A Benchmark Dataset for Global High-Resolution Land Cover Mapping*. arXiv.org, 2022. DOI: https://arxiv.org/abs/2210.10732

[10] Ronneberger O., Fischer P., Brox T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.org, 2015. URL: https://arxiv.org/abs/2210.10732

Valishin A.A., Dr. Sc. (Phys. — Math.), Professor, Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University. e-mail: enf@mail.ru

Zaprivoda A.V., student of Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University.

Tsukhlo S.S., student of Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University.