

Содержание

1	Литературный обзор	2
1.1	Определения	2
1.2	Обзор алгоритмов сегментации	2
2	Глава 2	7
2.1	Набор данных	7
2.2	Алгоритмы сегментации в экспериментах	15
2.3	Метрики оценки качества сегментации	18
2.4	Конвейер основных экспериментов	26
3	Глава 3	32
3.1	Правило останова	32

1 Литературный обзор

1.1 Определения

В рамках текущей работы, изображением размера $n = (n_1, n_2, n_3)$ будем называть отображение $I_n^m : P_3 \rightarrow G$, в котором элементы области определения $P \subset Z^3$ называются пикселями (вокселями), а $n_i \in N, i = 1, 2, 3$ и $(G, +)$ - аддитивная абелева группа [1].

Бинарной маской изображения I_n^m будем называть отображение $\tilde{I}_n^m : P_3 \rightarrow \{0, 1\}$, множество определения которого совпадает с множеством определения I_n^m .

TODO - add definition of множествами объекта и чувствительность метрик.

1.2 Обзор алгоритмов сегментации

В рамках данной работы рассматривается поведение сегментации изображения реконструкции в процессе томографии под контролем реконструкции. При этом выбор алгоритмов сегментации является важным шагом для дальнейшего проведения экспериментов.

С целью выбора алгоритма необходимо провести обзор доступных алгоритмов сегментации.

В научной периодике доступно большое количество публикаций на тему алгоритмов сегментации. Исследование [2] показало устойчивый рост количества ежегодно предложенных алгоритмов с 1995 года по 2006 года.

Систематизация и классификация доступных алгоритмов необходима для выбора наиболее подходящих для проведения экспериментов.

Хорошей отправной точкой в изучении доступных алгоритмов сегментации является рассмотрение схем их классификации.

Существует множество различных схем классификации алгоритмов сегментации. Например, часто используется классификация по одному признаку, такому как способ обработки изображения или степень участия человека в процессе сегментации [3].

В работе [4] также предложена схема обобщённой классификации,

объединяющая несколько одно-признаковых подходов в единую структуру.

В рамках проведённого литературного обзора поиск алгоритмов сегментации был проведен на основе их классификации по принципу работы.

Первым классом алгоритмов, встречающихся в литературе, является семейство пороговых алгоритмов.

Пороговые алгоритмы бинаризации выполняют классификацию вокселей изображения на основе заданного порогового значения интенсивности. Воксели с интенсивностью ниже порога относятся к фону, тогда как воксели с интенсивностью выше или равной порогу классифицируются как принадлежащие объекту.

Существует большое количество разнообразных пороговых алгоритмов, основные из которых изложены в работы [3].

Например, часть алгоритмов устанавливают порог глобально, другие - устанавливают его индивидуально для каждого пикселя.

Другим широко используемым подходом к сегментации является метод роста области (region growing), основанный на объединении вокселей с близкими характеристиками по мере распространения от заданных начальных точек [5].

Этот метод является полуавтоматическим, то есть он требует участия пользователя в процессе своей работы. В частности требуется ввести набор начальных точек.

В литературе встречается множество модификаций этого метода, включая автоматический вариант алгоритма, не требующий указания начальной точки [6].

Несмотря на такое множество модификаций, метод роста области достаточно трудно реализуем в контексте задачи сегментации под контролем реконструкции.

Полуавтоматический характер метода и ресурсоёмкость его автоматической модификации делают его непрактичным для использования в рамках данного исследования.

Следующим направлением, широко представленным в литературе, являются методы кластеризации. Схожесть формальных постановок

задач сегментации и кластеризации способствует применению кластеризационных алгоритмов в контексте сегментации.

В частности, алгоритмы K-means [7] и mean shift [8] нередко применяются при решении задач сегментации.

Применимость алгоритмов кластеризации к сегментации изображений реконструкции требует дополнительного анализа, поскольку реализация их вычислительно эффективных версий представляет собой нетривиальную задачу.

Представленные ранее классы алгоритмов во многом однородны по своей структуре и принципам работы. В то время как последующие группы объединяют существенно более разнородные методы, классифицированные по более общим признакам.

Такой группой алгоритмов являются методы на базе теории графов [9].

Принципы работы этих методов достаточно разнообразны. Некоторые работы применяют алгоритмы поиска разрезом в графе [10; 11], другие работы адаптируют алгоритм поиска максимального потока под задачу сегментации [12].

Также встречаются и нестандартные подходы к сегментации в этой группе алгоритмов.

Один из таких нестандартных подходов предложен в работе [13], где реализован алгоритм, концептуально близкий к Unseeded Region Growing, но основанный на представлении изображения в виде взвешенного графа.

Следующей группой являются вероятностные алгоритмы. Принцип работы методов в этой группе основан на некотором априорном предположении о распределении значений пикселей в рамках изображения.

На пример методы, изложенные в работах [14; 15] основаны на предположении, что значение пикселей объектов в изображении имеет распределение вейбюля.

Последней крупной группой алгоритмов, встречающейся в литературе, это методы на основе машинного обучения.

К алгоритмам на основе машинного обучения относятся как нейросетевые методы, применяемые в общей задаче сегментации изображений

[16; 17], так и подходы, адаптированные под изображения реконструкции в рамках томографии [18].

Также встречаются менее распространённые методы, включая клеточные нейронные сети [19] и отдельные примеры полуавтоматических алгоритмов, использующих классические модели, такие как метод опорных векторов [20; 21].

Некоторые выявленные в литературе методы не вписываются в описанные выше категории и базируются на оригинальных, зачастую уникальных подходах.

Алгоритмы этого типа представлены в небольшом числе работ и не формируют обособленного направления.

Одним из таких алгоритмов является Полуавтоматический SegMo [22], разработанный для сегментации изображений реконструкции индустриальной томографии.

Сам алгоритм достаточно сложный в реализации и требует активного участия пользователя, соответственно не подходит для исследования томографии под контролем реконструкции, однако авторы заявляют достаточно высокое качество сегментации на выходе.

Среди редких методов также встречаются алгоритмы на основе множеств уровня, использующие эволюцию поверхностей для выделения объектов в объёме. В работе [23] такой подход применён для трёхмерной сегментации сосудистой системы на данных магнитно-резонансной ангиографии.

В рамках литературного обзора были изучены пороговые алгоритмы, методы роста области, алгоритмы на основе кластеризации, теории графов, вероятностные подходы, а также методы, использующие машинное обучение и нестандартные оригинальные принципы.

Из большого разнообразия доступных алгоритмов была выбрана группа пороговых методов.

Причина выбора пороговых методов заключается в их высокой степени изученности. В литературе представлено множество работ, посвящённых их модификации, анализу и практическому применению, обладающих высокой цитируемостью.

Применение алгоритмов других групп в контексте томографии под

контролем реконструкции сопряжено с рядом ограничений.

Многие из них являются полуавтоматическими и требуют участия пользователя, что делает их непригодными для включения в итеративный процесс реконструкции.

Классы алгоритмов с высокой вычислительной сложностью, такие как методы кластеризации и алгоритмы на основе множеств уровня, затруднительно использовать в силу необходимости их повторного запуска на каждой итерации томографического сканирования.

Методы, основанные на нейронных сетях, требуют большого объёма размеченных данных, которые на текущий момент недоступны для рассматриваемой задачи.

Следующим этапом после выбора класса является формирование перечня конкретных алгоритмов, из которых будет производиться отбор для последующего эксперимента.

2 Глава 2

2.1 Набор данных

В целях исследования итеративного процесса томографии под контролем реконструкции с точки зрения качества сегментации необходимы следующие компоненты:

1. Набор данных, на которых проводятся эксперименты.
2. Алгоритмы сегментации.
3. Метрики для оценки качества сегментации.

Рассмотрим первую компоненту - входной набор данных.

Под набором данных, в рамках данной работы, подразумевается множество из n изображений $(I_m^n)_1, (I_m^n)_2, \dots, (I_m^n)_n$.

Основной целью работы является исследование правила останова в итеративной томографии под контролем реконструкции. Поэтому набор данных должен моделировать различные варианты соотношения между углами сканирования и угловой структурой изображений.

Большинство изображений в наборе данных являются синтетическими и были сгенерированы с помощью языка программирования Python.

Набор данных состоит из 10 изображений, сгруппированных по 2 на основе схожести моделируемой угловой структуры.

Первые 4 пары изображений имеют одинаковый размер - 256 срезов, каждый из которых 512×512 пикселей. Объем каждого изображение соответственно имеет размер 512×256 .

Первая пара представляет собой два изображения реконструкции бетонных плит, предоставленные в работе [24].

Изображение "Бетон-1" содержит характерные горизонтально ориентированные дефекты, интерпретируемые как трещины. На рисунке 1 приведен срез этого изображения и его трехмерная визуализация.

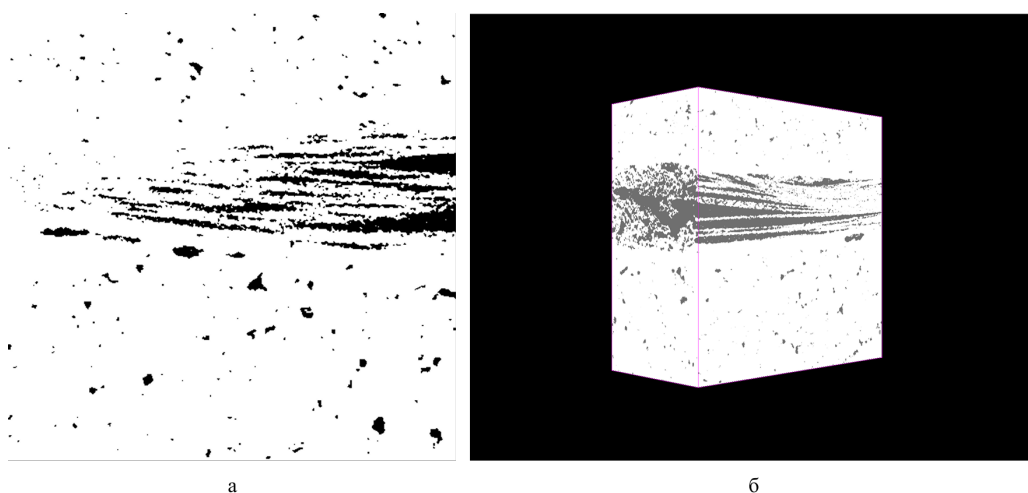


Рисунок 1 Изображение реконструкции "Бетон-1"; а - срез 32, б - трехмерная визуализация

Изображение "Бетон-2" представляет собой бетонную плиту с дефектом типа раковина - пустое пространство внутри плиты.

На рисунке 2 приведен срез изображения "Бетон-2" и соответствующая трехмерная модель.

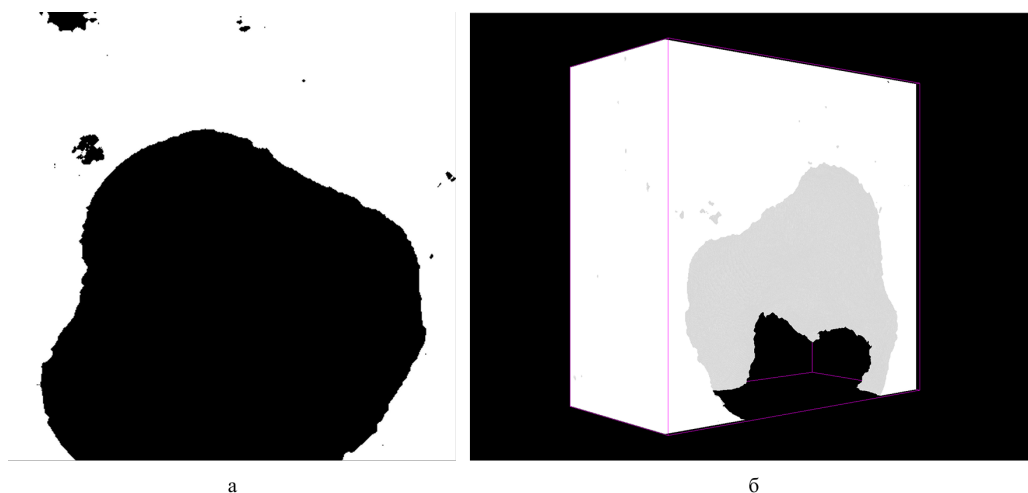


Рисунок 2 Изображение реконструкции "Бетон-2"; а - срез 32, б - трехмерная визуализация

В этих изображениях стоит отметить преобладание объекта над фоном по количеству пикселей.

Следующей парой объектов в наборе данных являются синтетические изображения трёхмерной решётки, сформированной пересекающимися семействами параллельных плоскостей.

В одном случае решётка строго ортогональна координатным осям. Это изображение в дальнейшем обозначается как "Решётка". На рисун-

ке 3 представлен срез её реконструкции и объемная модель.

Такой объект включён в набор данных как модель идеального случая: углы ориентации структур внутри объекта полностью совпадают с начальными углами сканирования в процессе томографии под контролем реконструкции, соответственно процесс должен уже остановиться при небольшом количестве проекций.

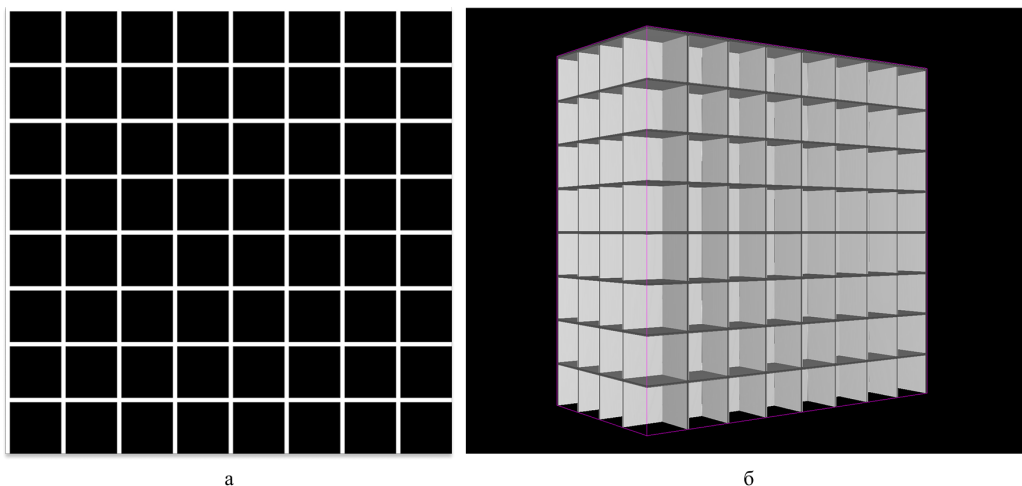


Рисунок 3 Изображение реконструкции "Решётка"; а - срез 32, б - трехмерная визуализация

Вторая решётка, обозначаемая как "Наклонная решётка" формируется по тому же принципу, что и предыдущая, но все её плоскости наклонены относительно координатных осей.

Каждое семейство плоскостей в этой решётке изначально было ортогонально одному из направлений декартовой системы координат. Затем к каждому семейству был применён поворот вокруг одной из осей: первое семейство — на угол 1,31 радиан вокруг оси X , второе — на угол 0,13 радиан вокруг оси Y , третье — на угол 1,31 радиан вокруг оси Z .

Срез изображения и трехмерная визуализация приведены на рисунке 4.

Данный объект моделирует случай, когда начальный набор углов сканирования не совпадает с ориентацией внутренних структур объекта. При этом число таких ориентаций ограничено: решётка состоит из трёх групп плоскостей, каждая из которых наклонена под фиксированным углом.

Ожидается, что при включении в процесс реконструкции проекций под соответствующими углами произойдёт резкое улучшение качества

бинаризации.

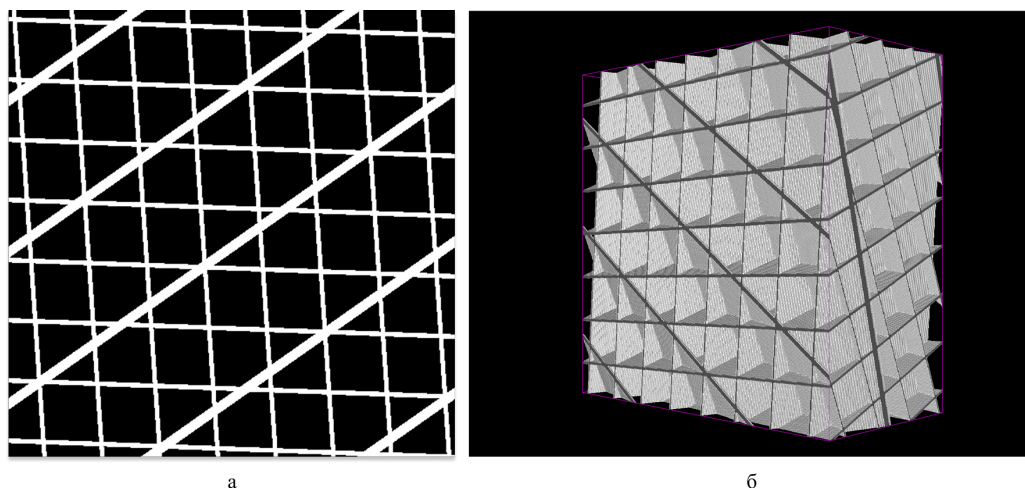


Рисунок 4 Изображение реконструкции "Наклонная решётка"; а - срез 32, б - трехмерная визуализация

Далее рассмотрим пару синтетических изображений, содержащих объекты с плавными очертаниями, без резких переходов или острых углов.

В отличие от структур с резкими границами, для них не ожидается резкого скачка качества при включении определённых углов сканирования — качество улучшается постепенно с включением большего количества проекций.

Первое изображение из этой пары - "Эллипсы". Оно представляет собой объём, разделённый на 27 параллелепипедов, в каждый из которых вписан трёхмерный эллипсоид. Размеры эллипсоидов варьируются, и в случае, если фигура не помещается в отведённую область, её выходящие за границы части отсекаются.

Срез и трехмерная визуализация этого изображения представлена на рисунке 5.

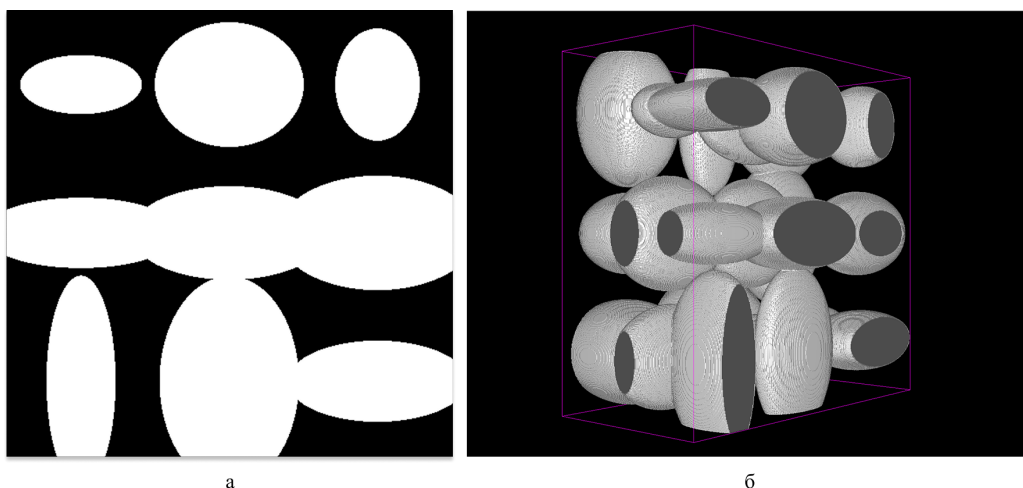


Рисунок 5 Изображение реконструкции "Эллипсы"; а - срез 32, б - трехмерная визуализация

Вторым изображением из данной пары является "Гауссиана". Оно представляет собой синтетическую трёхмерную структуру, состоящую из двух объёмных образований, форма которых аппроксимирует гауссовы колокола.

Первая фигура расположена в нижней части объёма, начинаясь с плоскости $z = 0$, и ориентирована так, что её максимум находится вблизи центра, а ось симметрии направлена вверх. Вторая фигура начинается с верхней части объёма, от плоскости $z = 255$, и ориентирована в противоположную сторону — вниз.

Обе фигуры подвергнуты повороту: первая — вокруг оси x , вторая — вокруг оси y . Угол поворота в обоих случаях составляет $\frac{\pi}{22}$ радиан.

Рисунок 6 иллюстрирует изображение «Гауссиана»: слева показан его срез, справа — трёхмерная визуализация.

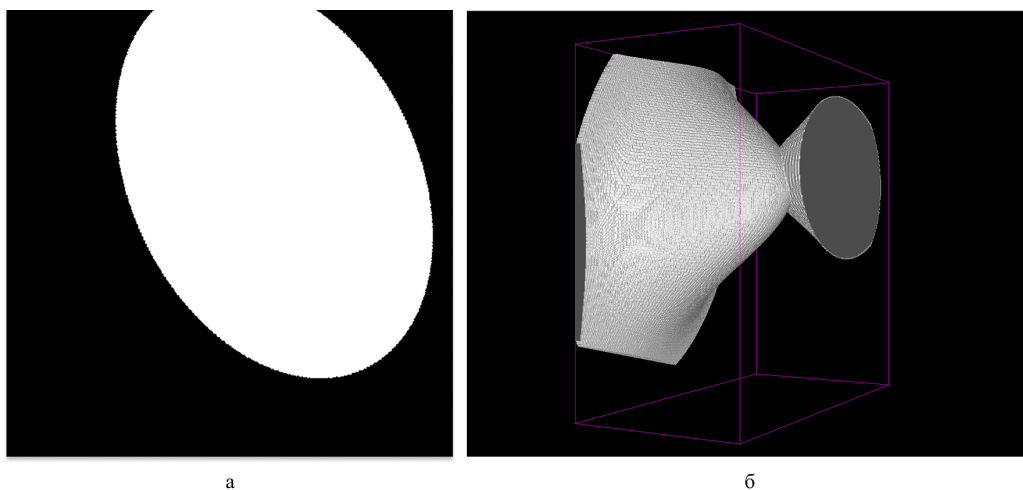


Рисунок 6 Изображение реконструкции "Гауссиана"; а - срез 32, б - трехмерная визуализация

Следующая пара синтетических изображений является противоположностью предыдущей - состоит из резких углов и ровных линий.

У таких изображений ожидаются скачки в качестве реконструкции при включении проекций под соответствующими углами. В отличие от пары "Решётка" и "Наклонная решётка" таких скачков может быть несколько, поскольку изображение содержит несколько трёхмерных объектов с различными ориентациями.

В состав изображения входят две пересекающиеся пирамиды: одна с прямоугольным основанием, другая — с треугольным. Оси симметрии пирамид ориентированы по-разному: первая повернута вокруг оси z , вторая — вокруг оси y .

В дополнение к ним, в свободное пространство между пирамидами встроен трёхмерный гексагон, не пересекающийся с другими объектами. Все фигуры размещены в объёме случайным образом, что исключает регулярность и добавляет сложности реконструкции.

В случае первого изображения "Полигоны-1" приведенного на рисунке 7, пирамида с прямоугольным основанием имеет поворот вокруг оси z на угол 1.005 радиан, а пирамида с треугольным основанием имеет угол поворота вокруг оси y равный 2.89 радиан.

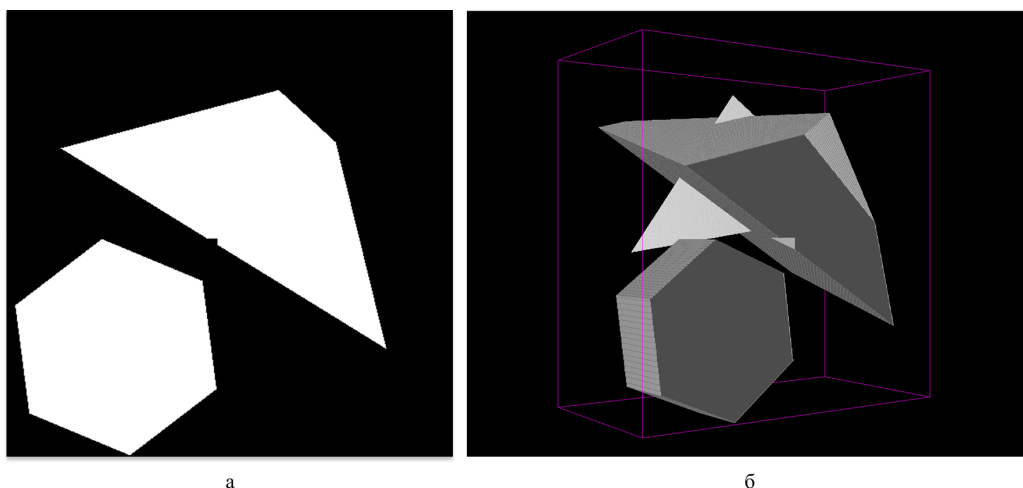


Рисунок 7 Изображение реконструкции "Полигоны-1"; а - срез 32, б - трехмерная визуализация

Второе изображение - "Полигоны-2" проиллюстрировано на рисунке 8. В нём пирамида с прямоугольным основанием повёрнута вокруг оси z на угол 4.71 радиан, а пирамида с треугольным основанием — вокруг оси y на угол 2.89 радиан.

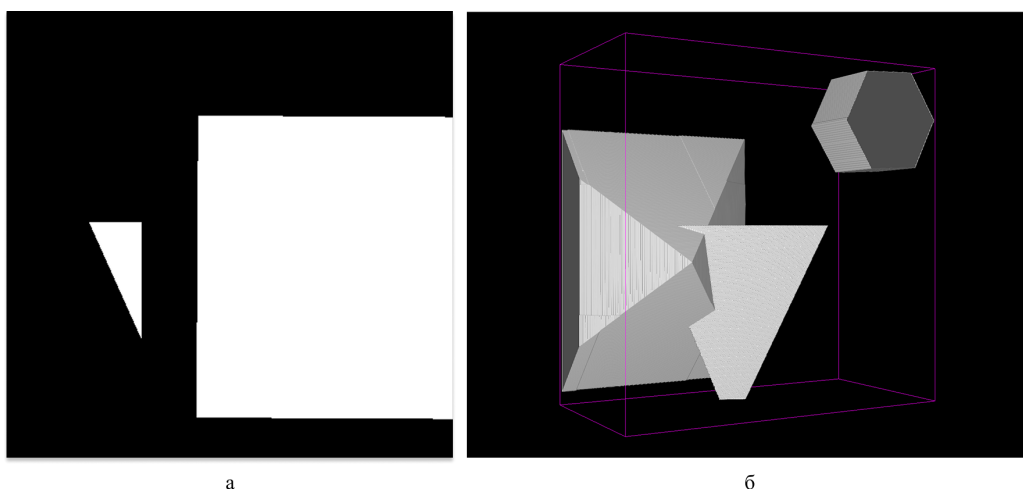


Рисунок 8 Изображение реконструкции "Полигоны-2"; а - срез 32, б - трехмерная визуализация

Последней парой изображений являются трехмерные модели построенные из изображений реконструкции при сканировании объектов с помощью сканера Cyberware 3030 MS в лаборатории компьютерной графики Стэнфордского университета [25].

Данные изображения представляют собой сканы реальных объектов с высокой геометрической детализацией, что позволяет оценить поведение правила останова на данных, приближённых к практическим

условиям применения.

Первым изображением из данной пары, которое назовем "Кролик" является скан фигурки кролика и состоит из 100 срезов размером 128×129 , соответственно размер объема $128 \times 129 \times 100$.

Один из срезов изображения и его трехмерная визуализация представлены в рисунке 9.

Фигурка является полой, соответственно особенность данного изображения является преобладание объекта над фоном. Это делает данное изображение противоположностью изображений "Бетон-1" и "Бетон-2".

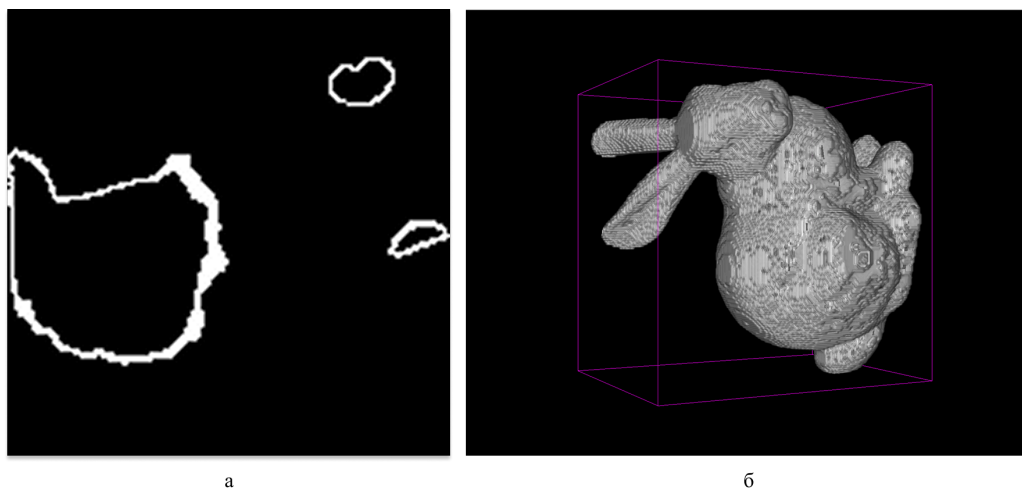


Рисунок 9 Изображение реконструкции "Кролик"; а - срез 32, б - трехмерная визуализация

Последним изображением является скан небольшой тайской статуэтки и представлено на рисунке 10. Трехмерное изображение формируется из 132 срезов размером 257×153 пикселей, соответственно полный объем имеет размер $257 \times 153 \times 132$ пикселя.

Её особенностью являются достаточно сложные границы объекта. В отличие от изображения "Кролик" данное изображение, которое назовем "Тайская статуэтка" не является полым.

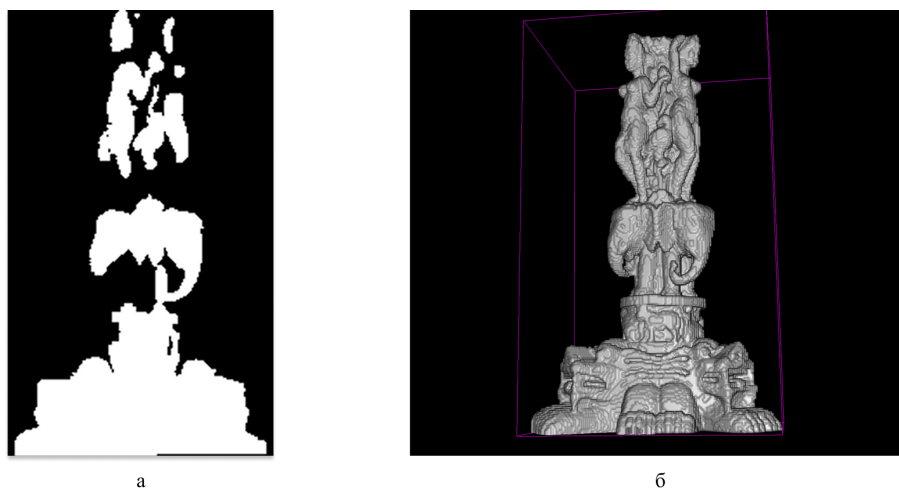


Рисунок 10 Изображение реконструкции "Тайская статуэтка"; а - срез 32, б - трехмерная визуализация

Собранный набор данных, состоящий из десяти изображений с разнообразной структурой, охватывает широкий спектр сценариев, возникающих при томографической реконструкции: от регулярных и идеально выровненных структур до объектов с произвольной ориентацией и сложной геометрией.

Такой набор данных формирует основу исследования, поскольку правило останова может считаться надёжным лишь в случае его устойчивой работы на разнообразных и сбалансированных по характеристикам данных.

Далее будем рассматривать остальные компоненты, необходимые для конвейера основных экспериментов.

2.2 Алгоритмы сегментации в экспериментах

С учётом необходимости многократного применения в процессе итеративной реконструкции, для экспериментов были выбраны пороговые алгоритмы, как простые, устойчивые и не требующие сложной настройки.

Основная идея пороговых алгоритмов достаточно проста.

Пусть дано изображение I_m^n размера $n = (n_1, n_2, n_3)$.

Выходом алгоритма будет сегментация \tilde{I}_m^n изображения I_m^n , совпадающая размером с исходным изображением.

Значение каждого пикселя сегментации \tilde{I}_m^n определяется по следующей формуле:

$$\tilde{I}_m^n(i, j, k) = \begin{cases} 1, I_m^n(i, j, k) \geq t \\ 0, I_m^n(i, j, k) < t \end{cases} \quad (1)$$

Соответственно в сегментации пиксель (i, j, k) классифицируется как объект если значение исходного изображения в этом пикселе имеет значение больше или равное некоторому порогу t , иначе этот пиксель классифицируется как фон.

В зависимости от характера порога t выделяют глобальные и локальные пороговые алгоритмы.

В глобальном случае порог не зависит от пикселя, соответственно все пиксели сравниваются с одним значением порога.

Таковыми алгоритмами являются классический пороговый алгоритм и алгоритм Отсу.

В классическом пороговом алгоритме порог является параметром, соответственно он требует его априорной оценки.

Алгоритм Отсу [26] определяет оптимальное значение порога t , максимизируя межклассовую дисперсию:

$$\sigma^2(t) = \omega_0(t)\omega_1(t) [\mu_0(t) - \mu_1(t)]^2, \quad (2)$$

где $\omega_0(t)$ и $\omega_1(t)$ — вероятности (доли) фона и объекта при пороге t , а $\mu_0(t)$ и $\mu_1(t)$ — соответствующие средние значения интенсивности.

Алгоритм перебирает возможные значения t и выбирает то, при котором значение $\sigma^2(t)$ максимизируется.

Локальные пороговые алгоритмы определяют порог для каждого пикселя, соответственно порог t становится функцией от пикселя $t = t(i, j, k)$.

Распространенным локальным пороговым алгоритмом является метод Ниблэка [27].

Алгоритм Ниблэка определяет порог t для пикселя (i, j, k) по следующей формуле:

$$t(i, j, k) = \mu_r(i, j, k) + k\sigma_r(i, j, k) \quad (3)$$

где $\mu_r(i, j, k)$ и $\sigma_r(i, j, k)$ - среднее и среднеквадратичное отклоне-

ние интенсивности в окрестности r пикселя (i, j, k) .

В общем случае среднее и среднеквадратичное отклонение в окрестности r пикселя (i, j, k) рассчитываются по следующим формулам:

$$\mu_r(i, j, k) = \frac{1}{(2r + 1)^3} \sum_{x=i-r}^{i+r} \sum_{y=j-r}^{j+r} \sum_{z=k-r}^{k+r} I_m^n(x, y, z) \quad (4)$$

$$\sigma_r(i, j, k) = \sqrt{\frac{1}{(2r + 1)^3} \sum_{x=i-r}^{i+r} \sum_{y=j-r}^{j+r} \sum_{z=k-r}^{k+r} (I_m^n(x, y, z) - \mu_r(i, j, k))^2} \quad (5)$$

Соответственно окном является куб с длиной сторон $2r + 1$.

Проблемной частью формул 4 и 5 является ситуация, когда окно частично выходит за границы изображения.

В таких случаях применяются следующие стратегии обработки:

1. Обрезка окна по границам. Вычисления проводятся только по той части окна, которая полностью попадает внутрь изображения.
2. Задание фиксированного значения. За пределами изображения значения пикселей считаются равными фиксированной константе, например, нулю или среднему значению изображения.
3. Отражение по границе. Отсутствующие значения заполняются за счёт зеркального отражения пикселей относительно соответствующей границы изображения.

У каждой стратегии есть свои преимущества и недостатки. Её выбор зависит от характера входных изображений.

У классического алгоритма Ниблэка существует множество модификаций.

Таковыми модификациями являются, например, алгоритмы Сауволя [28] и Фансалкара [29].

В рамках данной работы выбран аффинный вариант алгоритма Ниблэка, описанный в работе [30].

Метод аффинного Ниблэка добавляет в уравнение 3 дополнительный параметр *beta*, который является глобальной оценкой шума.

$$t(i, j, k) = \mu_r(i, j, k) + \sigma_r(i, j, k) + \beta \quad (6)$$

Такая модификация является вычислительно эффективной так как к формуле добавляется константа, однако она добавляет дополнительный параметр.

Таким образом, итоговый набор алгоритмов, используемых в конвейере экспериментов, включает в себя:

1. Классический пороговый алгоритм
2. Алгоритм Отсу
3. Алгоритм аффинного Ниблека

Выбор данных алгоритмов обусловлен их высокой степенью изученности в литературе и активным применением в задачах компьютерной томографии.

Поскольку алгоритм бинаризации запускается на каждой итерации томографии под контролем реконструкции, его вычислительная эффективность напрямую влияет на общую скорость всего процесса.

В целях оценки результата работы алгоритмов сегментации необходим набор метрик, позволяющих сформировать наиболее целостное представление о полученных бинарных масках.

2.3 Метрики оценки качества сегментации

Последним необходимым элементом экспериментального конвейера является набор метрик оценки качества сегментации.

Метрики должны быть подобраны так, чтобы в совокупности охватывать ключевые аспекты качества бинарных масок — например, точность границ и степень перекрытия с эталоном.

Такой набор метрик должен выявлять различные типы ошибок и предоставлять более объективную оценку результата.

Хотя в области машинного обучения и компьютерного зрения существует множество метрик, не все из них применимы к задачи сегментации изображений.

Например, метрики, ориентированные на числовые значения, текстовые последовательности или графовые структуры, не учитывают пространственные особенности изображений и потому неинформативны в контексте сегментации.

Даже среди метрик, предназначенных для работы с изображениями, нередко встречается избыточность: несколько показателей могут быть чувствительны к одним и тем же видам ошибок, и их совместное использование не даёт дополнительной информации.

В конвейер экспериментов требуется сформировать минимально необходимый набор метрик, который позволит обнаружить распространённые типы ошибок.

Чтобы выбрать такой набор метрик обоснованно, необходимо провести серию экспериментов на модельных данных.

Такие эксперименты позволят изучить поведение метрик при различных типах ошибок.

Для реализации таких экспериментов необходимы следующие шаги:

1. Собрать начальный набор метрик для исследования.
2. Определить набор ошибок, которые должны обнаружить метрики.
3. Сформировать модельные изображения для набора ошибок

Часть метрик для начального набора были найдены в работе [31]. Авторы данного исследования изложили распространённые метрики, применяемые в оценке качества сегментации изображений реконструкции в медицине.

Из таких метрик в начальный набор данных будут включены метрики DICE, Intersection Over Union (IOU), Mean Square Error (MSE).

Пусть даны эталонная бинарная маска $(\tilde{I}_m^n)_{gt}$ изображения и его сегментация $(\tilde{I}_m^n)_{seg}$.

Обозначим как $(\tilde{I}_m^n)_{gt} \cap (\tilde{I}_m^n)_{seg}$ множество пикселей, на которых значение сегментации и эталонной бинарной маски совпадают и равно 1: $(\tilde{I}_m^n)_{gt} \cap (\tilde{I}_m^n)_{seg} := |\{x \in P | (\tilde{I}_m^n)_{gt}(x) = (\tilde{I}_m^n)_{seg}(x) = 1\}|$.

Определим также $(\tilde{I}_m^n)_{gt} \cup (\tilde{I}_m^n)_{seg}$ как множество пикселей, на которых значение сегментации или эталонной бинарной маски равно 1:
 $(\tilde{I}_m^n)_{gt} \cup (\tilde{I}_m^n)_{seg} := |\{x \in P | (\tilde{I}_m^n)_{gt}(x) = 1 \vee (\tilde{I}_m^n)_{seg}(x) = 1\}|$.

Тогда метрика DICE определяется как следующей формулой:

$$DICE((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = \frac{2 \cdot |(\tilde{I}_m^n)_{gt} \cap (\tilde{I}_m^n)_{seg}|}{|(\tilde{I}_m^n)_{gt}| + |(\tilde{I}_m^n)_{seg}|} \quad (7)$$

где $|(\tilde{I}_m^n)|$ является мощностью множества пикселей, на которых маска (\tilde{I}_m^n) имеет значение 1: $|(\tilde{I}_m^n)| = |\{x \in P | (\tilde{I}_m^n)(x) = 1\}|$.

Метрика DICE принимает значения в отрезке от 0 до 1, где 0 - полное несовпадение изображений, а 1 - их полное совпадение.

Индекс DICE является одной из классических метрик перекрытия — класса метрик, которые измеряют степень пересечения двух множеств (в данном случае бинарных масок).

Другой распространенной метрикой перекрытия является IOU, также известная как метрика Жаккарда. Она определяется следующей формулой:

$$IOU((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = \frac{|(\tilde{I}_m^n)_{gt} \cap (\tilde{I}_m^n)_{seg}|}{|(\tilde{I}_m^n)_{gt} \cup (\tilde{I}_m^n)_{seg}|} \quad (8)$$

Аналогично метрики DICE, значения IOU находятся в отрезке от 0 до 1, где 0 - полное несовпадение изображений, а 1 - их полное совпадение.

Последняя метрика перекрытия несколько менее распространенная - Symmetric Boundary DICE (SBD), описанной в работе [32].

Чтобы определить эту метрику необходимо ввести понятия r окрестности пикселя x и границы изображения.

Окрестностью радиуса r пикселя $x \in P$ маски \tilde{I}_m^n будем называть множество

$$M_r(x) = \left\{ x' \in P \left| \max_{i \in \{0,1,2\}} |x_i - x'_i| \leq r \right. \right\}, \quad (9)$$

то есть все пиксели, находящиеся в дискретной окрестности Мура радиуса r от x .

Размер этой окрестности напрямую влияет на чувствительность метрики: при малом r метрика фиксирует только локальные несоответ-

ствия границ, а при большом — начинает учитывать отклонения на большем расстоянии, сглаживая мелкие ошибки.

Границей $\partial \tilde{I}_m^n$ маски \tilde{I}_m^n будем называть множество пикселей, окрестность радиуса r которых содержит как пиксели фона, так и пиксели объекта:

$$\partial I_m^n = \{x \in P | \exists x' \in M_r(x) : \tilde{I}_m^n(x') = 0 \wedge \exists x'' \in M_r(x) : \tilde{I}_m^n(x'') = 1\} \quad (10)$$

Заметим, что так как метрики сравнивают бинарную маску $(\tilde{I}_m^n)_{seg}$ и эталонную маску $(\tilde{I}_m^n)_{gt}$ одного и того же исходного изображения, то у них одинаковая область определения P .

Это позволяет выбрать одну окрестность радиуса r пикселя x и сравнивать значения пикселей из этого множества в обоих изображениях.

Обозначим через $DICE(M_r(x))$ значение метрики dice на множестве $M_r(x)$:

$$DICE(M_r(x)) = DICE[(\tilde{I}_m^n)_{seg}(M_r(x)), (\tilde{I}_m^n)_{gt}(M_r(x))] \quad (11)$$

Определив необходимые понятия можно перейти к определению метрики Symmetric Boundary DICE:

$$SBD((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = \frac{\sum_{x \in \partial(\tilde{I}_m^n)_{seg}} DICE(M_r(x)) + \sum_{x \in \partial(\tilde{I}_m^n)_{gt}} DICE(M_r(x))}{|\partial(\tilde{I}_m^n)_{gt}| + |\partial(\tilde{I}_m^n)_{seg}|} \quad (12)$$

В числителе формулы суммируются значения метрики Dice, вычисленные в окрестностях радиуса r каждого граничного пикселя как сегментированной маски, так и эталонной. Знаменатель представляет собой общее количество таких граничных пикселей, обеспечивая нормировку результата. Параметр r задаёт радиус окрестности, используемой для локального сравнения, и определяется ранее в формуле (9).

Следующая метрика в начальном наборе - метрика Normalized Hausdorff - отличается от предыдущих двух по принципу сравнения.

Данная метрика основана не на площади перекрытия, а на расстоя-

нии между множествами объекта в сегментации и в эталонной бинарной маске.

Чтобы определить метрику Normalized Hausdorff, определим расстояние между пикселем $x \in P$ и множеством $(\tilde{I}_m^n) = \{x | (\tilde{I}_m^n)(x) = 1\}$.

$$d(x, (\tilde{I}_m^n)) = \inf_{y \in (\tilde{I}_m^n)} (\sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2 + (x_2 - y_2)^2}) \quad (13)$$

Операясь на формулу 13 можно определить расстояние Хаусдорфа:

$$HD((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = \max(\sup_{x \in (\tilde{I}_m^n)_{seg}} d(x, (\tilde{I}_m^n)_{gt})) \quad (14)$$

Соответственно значением расстояния Хаусдорфа является максимум из минимумов расстояний от пикселя до множества объектов.

Определив все необходимые формулы можно определить метрику Normalized Hausdorff Distance:

$$NHD((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = 1 - \frac{\max(HD((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}), HD((\tilde{I}_m^n)_{gt}, (\tilde{I}_m^n)_{seg}))}{\sqrt{n_0^2 + n_1^2 + n_2^2}} \quad (15)$$

Максимальное значение расстояния Хаусдорфа в пределах изображения не превышает его диагонали, поэтому длина диагонали является разумным нормализующим коэффициентом.

Чем меньше значение расстояния Хаусдорфа, тем ближе бинарные маски друг к другу.

Для удобства интерпретации нормализованное расстояние вычитается из единицы, поэтому метрика Normalized Hausdorff Distance принимает значения от 0 (полное несовпадение) до 1 (полное совпадение масок).

Последней метрикой в начальном наборе является MSE - средне-квадратичное расстояние между пикселями сравниваемых масок.

$$MSE((\tilde{I}_m^n)_{seg}, (\tilde{I}_m^n)_{gt}) = \frac{1}{n_0 \cdot n_1 \cdot n_2} \sum_{x \in P} ((\tilde{I}_m^n)_{seg}(x) - (\tilde{I}_m^n)_{gt}(x))^2 \quad (16)$$

Таким образом, начальный набор метрик составлен из различных по принципу оценивания метрик.

Следующим шагом в формировании итогового набора метрик является составление набора эталонных изображений для оценки чувствительности начального набора к различным видам ошибок.

Будем рассматривать в качестве типовых ошибок рябь, соответствие границ объекта, выброс и ошибки при преобладании объекта над фоном.

Рябь будем определять как наличие фоновых пикселей, ошибочно попавших в область объекта.

Ошибка соответствия границ возникает в тех случаях, когда сегментированная область в целом совпадает с объектом, но границы объекта смещены относительно эталона.

Следующим типом ошибок является выброс - некоторое небольшое количество пикселей ошибочно классифицированные как объект, при этом находящихся на расстоянии от области объекта.

Последней ситуацией, на которой будем проводить опыты над метриками, является преобладание объекта над фоном.

В такой ситуации некоторые метрики могут быть менее чувствительными к ошибкам так как в абсолютных цифрах большинство изображений является объектом. На пример это может привести к хорошей оценки нечувствительной метрикой если просто классифицировать все изображение как объект.

Модельные изображения, на которых будут проводиться эксперименты с метриками, приведены в рисунке 11.

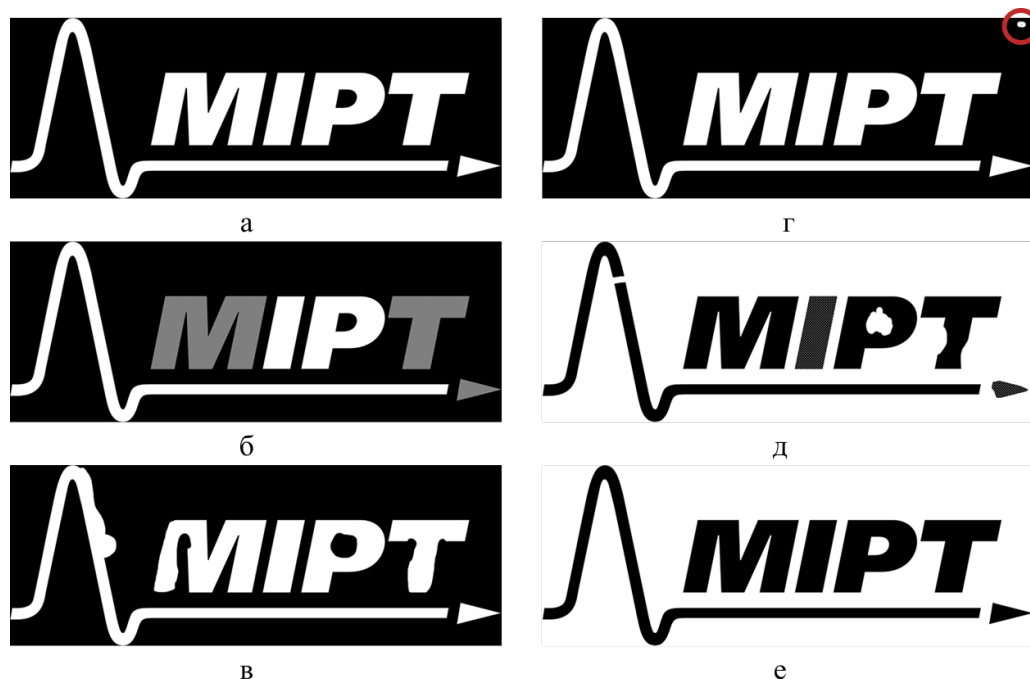


Рисунок 11 Тестовое изображения для экспериментов по оценке чувствительности метрик; а — эталонная маска, б — тест «Рябь», в — тест «Соответствие границ», г — тест «Выброс», д — тест «Преобладание объекта над фоном», е — эталонная маска для теста «Преобладание объекта над фоном»

Собрав начальный набор метрик, определив типы ошибок и модельные изображения, можно провести эксперименты по оценке чувствительности метрик.

Эксперименты реализованы с помощью языка программирования Python с использованием библиотек numpy для работы с изображениями как с многомерным массивом, PIL для операций чтения и записи изображений в файловой системе, а также matplotlib для визуализации результатов.

Настройки экспериментов читаются с файла формата JSON, который представляет собой массив из объектов, описывающих эксперимент.

Каждый объект описание эксперимента содержит его название и пути к модельному и эталонному изображениям.

Программа сравнивает по каждой метрике модельные изображения с эталонными и записывает результат в таблицу, столбцы которой являются значениями метрик, а строки соответствуют экспериментам.

Заполнив таблицу с результатами программа выводит ее визуализацию в виде табличной тепловой карты.

Получившаяся в результате экспериментов тепловая карта представлена на рисунке 12.

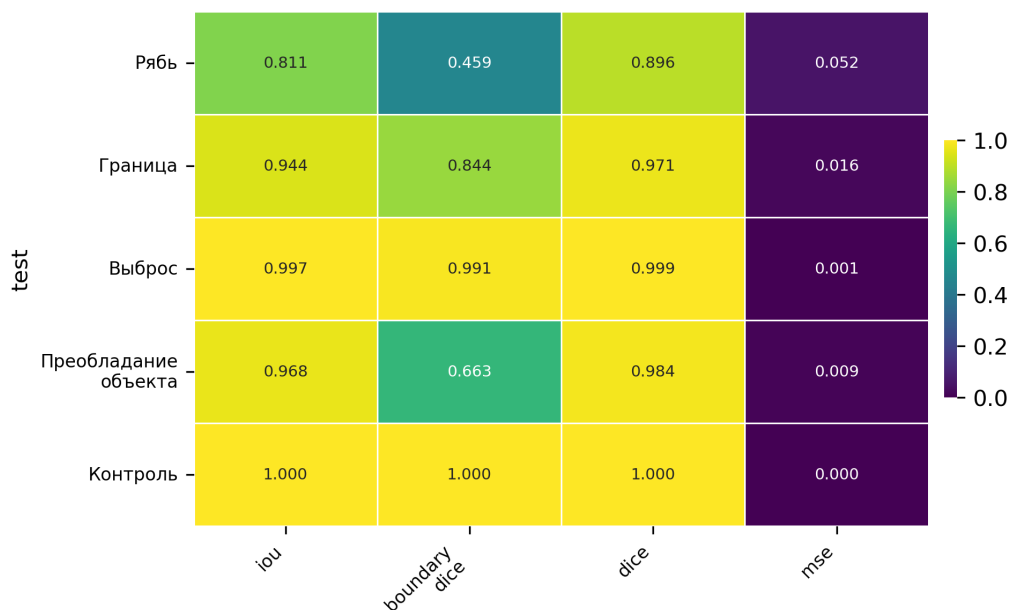


Рисунок 12 Тепловая карта значений метрик для модельных изображений с различными типами ошибок. Цвет и числа отражают чувствительность метрик к отклонениям от эталона.

Сравнение значений метрик показывает, что IOU и DICE в целом демонстрируют схожее поведение на различных типах ошибок, однако IOU в большинстве случаев оказывается более чувствительной: она сильнее снижается при наличии отклонений от эталона. Соответственно в итоговый набор метрик имеет смысл включить IOU и исключить DICE.

Symmetric Boundary DICE показывает высокую чувствительность к ошибкам соответствия границ, ряби и ошибкам при преобладании объекта над фоном.

Метрика MSE реагирует на все виды ошибок, однако её показания в совокупности дублируются метриками IOU и Symmetric Boundary DICE, что делает её избыточной в составе итогового набора.

Для ошибки типа выброс все метрики показывают сравнительно низкую чувствительность, однако наилучшие результаты наблюдаются у метрик IOU и Symmetric Boundary DICE.

В итоговый набор метрик включены IOU и Symmetric Boundary DICE. Первая представляет собой компактную и широко используемую метрику перекрытия, служащую ориентиром для оценки общей точности

сегментации, а вторая — наиболее чувствительна к локальным погрешностям и искажениям границ.

Собрав набор данных, алгоритмы и метрики можно приступить к проектированию и реализации конвейера основных экспериментов.

2.4 Конвейер основных экспериментов

Проектирование и реализация эффективного программного обеспечения начинается с чёткой постановки требований, поэтому первым шагом в разработке конвейера экспериментов является их формулировка.

Первым требованием к конвейеру является устойчивость к прерыванию.

Система должна обеспечивать возможность продолжения экспериментов с места остановки, поэтому результаты уже выполненных вычислений необходимо сохранять, чтобы при последующем запуске можно было продолжить выполнение без необходимости пересчёта уже обработанных данных.

Следующим требованием к конвейеру экспериментов является модульность — система должна позволять легко добавлять новые данные, алгоритмы и метрики без необходимости переработки остального кода.

Такая гибкость особенно важна при проведении серии экспериментов, где может потребоваться быстро изменять состав компонентов для оценки их влияния на итоговый результат.

Последним требованием к конвейеру является поддержка вычислений на графическом процессоре.

Многие используемые алгоритмы и метрики допускают параллельную обработку, поэтому реализация с возможностью использования дискретной видеокарты позволяет существенно ускорить вычисления и сократить общее время экспериментов.

Конвейер экспериментов был спроектирован с учетом вышесформулированных требований.

Система конвейера экспериментов будет состоять из пяти модулей:

1. Модуль реконструкции
2. Модуль бинаризации

3. Модуль оценки результатов

4. Модуль визуализации результатов

Рассмотрим подробнее каждый из этих модулей и сформулируем их входные данные, выход и особенности реализации.

Модуль реконструкции отвечает за симуляцию томографии под контролем реконструкции.

На вход этому модулю требуется трёхмерное изображение в виде набора двумерных срезов, а также файл с настройками, содержащий название изображения, путь к эталонной маске и параметры для выполнения реконструкции.

Кроме того, модулю передаются общие параметры эксперимента: стратегия выбора углов проекций и их максимальное количество.

К числу параметров реконструкции относятся, в частности, размеры проекционного детектора, задаваемые числом строк и столбцов, а также шаг детектора по осям, определяющий физический размер пикселя. Эти параметры влияют на масштаб и разрешение моделируемых проекционных данных. Также указывается используемый алгоритм реконструкции; если он является итеративным, дополнительно задаётся количество итераций, определяющее глубину восстановления.

На выходе модуль формирует реконструкции для каждого набора углов проекций и сохраняет их в формате TIFF, а также некоторые параметры и значение, которые понадобятся для следующего модуля.

Таковыми параметрами являются путь к эталонной маске, среднеквадратичное отклонение уровня шума в реконструкции, название исходного изображения, индекс набора углов и массив с углами в радианах.

Результаты сохраняются в директории с установленной структурой: для каждого входного изображения создаётся отдельная папка, имя которой совпадает с названием изображения. Внутри неё размещаются подкаталоги, пронумерованные от 0 до n , где n - количество наборов углов. Каждый из этих подкаталогов содержит изображение реконструкции, полученное на основе соответствующего набора углов.

С целью обеспечения устойчивости к прерыванию каждая рекон-

струкция сохраняется сразу после завершения расчёта. При повторном запуске модуль проверяет наличие соответствующего результата в папке и, если он уже существует, пропускает данный набор углов.

Модуль считывает входные данные из заданной директории, в которой должны располагаться подкаталоги — по одному на каждое изображение. Каждый такой подкаталог содержит срезы изображения и файл с его настройками.

Такой способ организации данных позволяет легко добавлять или удалять входные изображения без изменения исходного кода, что соответствует требованию модульности.

Реализация модуля выполнена на языке Python. Для моделирования процесса томографии под контролем реконструкции используется библиотека `astra-toolbox` [33], поддерживающая, в том числе, выполнение реконструкции на графическом процессоре.

Модуль бинаризации реализует бинарную сегментацию каждого входного изображения с помощью заданного набора алгоритмов.

Входными данными для модуля являются реконструкции, полученные на предыдущем этапе, а также параметры, необходимые для настройки алгоритмов бинаризации.

Работа модуля организована следующим образом: производится обход директории с результатами модуля реконструкции. Для каждой вложенной папки — соответствующей конкретному изображению и набору углов — считываются файл реконструкции в формате TIFF и сопутствующие параметры, сохранённые на этапе реконструкции.

Затем для каждой полученной реконструкции последовательно применяются все алгоритмы из заданного набора.

Набор алгоритмов формируется из папки `algorithms`, в которой должны быть файлы с реализацией алгоритма сегментации. Реализация должна содержать метод `segment`, принимающий на вход изображение реконструкции и эталонную маску. На выходе функция `segment` должна вернуть бинаризацию и объект с значениями параметров, используемых алгоритмом бинаризации.

Таким образом, требование модульности выполняется: для добавления или удаления алгоритма достаточно поместить или удалить соот-

ветствующий файл из директории `algorithms`, без необходимости внесения изменений в остальной код конвейера.

После выполнения бинаризации полученная маска сохраняется в формате TIFF в папке, путь к которой включает название изображения, индекс набора углов и имя алгоритма. В той же папке дополнительно сохраняется JSON-файл, содержащий параметры, использованные при бинаризации, путь к соответствующей эталонной маске, а также список с набором углов проекций, использованных при реконструкции входного изображения.

В целях обеспечения устойчивости алгоритма к прерыванию, при запуске производится проверка на существования пути к результатам и наличия там уже готовой бинаризации. При обнаружении таких файлов изображение пропускается.

Модуль реализован на языке Python с использованием библиотеки CuPy [34], предоставляющей реализации базовых численных операций, таких как свёртки и тензорные преобразования, на графическом процессоре с поддержкой CUDA.

Благодаря этому становится возможным эффективно использовать графический процессор для ускорения алгоритмов бинаризации, поскольку многие из них хорошо поддаются параллелизации.

Далее рассмотрим модуль оценки результатов. Этот модуль отвечает за вычисление значений метрик, характеризующих качество полученных бинаризаций.

Работа модуля организована как последовательный обход структуры папок, сформированной предыдущим этапом. В каждой вложенной директории он поочерёдно обрабатывает результаты бинаризации: загружает бинарную маску и соответствующий JSON-файл с параметрами, содержащими путь к эталонной маске, параметры алгоритма бинаризации и набор углов проекций.

После загрузки данных модуль рассчитывает значения всех метрик из заданного набора и сохраняет результат в CSV-файл. Структура путей при этом сохраняется аналогично результатам бинаризации, но с другой корневой директорией, соответствующей модулю оценки.

Обработка выполняется поэтапно: каждая бинаризация обрабаты-

вается отдельно, результаты сохраняются сразу после расчёта.

При этом, перед расчетом метрик, программа также проверяет наличие CSV файла в директории с результатами, соответствующей текущей входной бинарной маске. Если таковой существует - то программа пропускает данное изображение.

Такой подход обеспечивает устойчивость к прерыванию и позволяет избежать повторной обработки уже обработанных данных.

Аналогично предыдущему модулю, набор метрик формируется на основе содержимого директории `metrics`, в которой располагаются файлы с их реализациями. Название каждой метрики соответствует имени соответствующего файла. Каждый такой файл должен содержать функцию `evaluate`, принимающую на вход бинарную маску и эталонную маску, и возвращающую численное значение метрики.

Стоит также отметить, что метрика `Symmetric Boundary DICE` требует параметра r - радиуса окрестности граничных пикселей (12).

В целях простоты реализации в качестве значения параметра было выбрано $r = 5$.

Модуль был написан на языке Python с использованием библиотеки `CuPy`, позволяющей ускорить вычисления метрик с помощью графического процессора.

Последним модулем конвейера экспериментов является модуль визуализации. Его задача — агрегировать полученные результаты и представить их в наглядной графической форме.

Модуль считывает все CSV-файлы, сформированные на предыдущем этапе, из директории с результатами. В отличие от других модулей, он не обрабатывает файлы по одному, а загружает все данные сразу. Для удобства доступа и группировки реализован интерфейс `ResultData`, обеспечивающий получение результатов в различных представлениях — например, значения метрик по алгоритмам, по изображениям и другим признакам.

Такая конструкция позволяет программе соответствовать требованию модульности: для добавления новой визуализации достаточно разместить соответствующий файл с реализацией в директории `visualizers`, без необходимости внесения изменений в основной код.

Модуль реализован на языке python с применением библиотек numpy для работы с массивами данных и matplotlib для формирования графиков.

Применение графического процессора в рамках данного модуля избыточно, соответственно все операции проходят на центральном процессоре.

В реализации конвейера также присутствует контрольный модуль, который последовательно запускает другие модули. Он необходим так как все другие модули спроектированы как отдельные программы.

Таким образом, в данной главе были рассмотрены все основные компоненты, необходимые для построения эксперимента по исследованию итеративной томографии под контролем реконструкции: входной набор данных, алгоритмы сегментации, метрики оценки качества и программный конвейер. Конвейер представляет собой основу эксперимента, реализуя последовательную обработку изображений — от реконструкции до оценки и визуализации результатов.

В следующей главе будет сформулировано правило останова процесса томографии под контролем реконструкции на основе анализа бинаризаций. С помощью разработанного конвейера экспериментов будет проведена его проверка и оценена эффективность предложенного правила.

3 Глава 3

3.1 Правило останова

Томография под контролем реконструкции является итеративным процессом, соответственно в каждой итерации необходимо принять решение об остановке процесса на текущей реконструкции или добавления большего количества проекций.

В рамках текущей работы изучается автоматическое принятие такого решения заранее сформулированным правилом останова на основе анализа бинаризации.

Для изучения данного вопроса требуется сформулировать такое правило останова и рассмотреть его эффект на среднее качество бинаризации по изображениям на количество углов с помощью реализованного конвейера экспериментов.

В реальных условиях применения правила останова в процессе томографии под контролем реконструкции отсутствует доступ к эталонным данным, соответственно правило останова, основанное на объективной оценке качества, является неприменимым в данном контексте.

Однако собранный для конвейера экспериментов набор метрик позволяет всесторонне оценить схожесть двух входных изображений.

В процессе томографии под контролем реконструкции как правило в первой же итерации, с набором из 4 углов, маловероятно получить достаточно качественное изображения для остановки процесса.

Соответственно, для формулировки осмысленного правила останова требуется сравнение двух реконструкций — полученной на текущем наборе углов и реконструкции, соответствующей предыдущему набору.

Предполагается, что на каждой итерации доступны изображения $(I_m^n)_{i-1}$ и $(I_m^n)_i$, где $i = 1, \dots, n$ это индекс набора углов, а реконструкция, полученная на первом наборе, имеет индекс 0.

Правилом останова назовем отображение $S : R \rightarrow \{0, 1\}$, где значение 0 означает продолжение процесса томографии под контролем реконструкции, а 1 - завершение процесса на текущей реконструкции.

Сравнивая бинаризации двух соседних реконструкций можно получить информацию о степени изменения бинаризации с добавлением

большого количества проекций.

Аргументом для правила останова будет являться значение метрики M на бинаризациях $(\hat{I}_m^n)_{i-1}$ и $(\hat{I}_m^n)_i$.

Так как объективная оценка качества бинаризации не доступно при применении правило останова, его формулировка требует некоторых предположений и эвристики.

Первым предположением, лежащим в основе формулировки правила останова, является допущение о том, что при достаточно высокой степени схожести бинаризаций двух соседних реконструкций добавление новых проекций не приводит к значимым изменениям результата. Формально это выражается условием:

$$S((\hat{I}_m^n)_{i-1}, (\hat{I}_m^n)_i) = \begin{cases} 1, M((\hat{I}_m^n)_{i-1}, (\hat{I}_m^n)_i) \geq c \\ 0, M((\hat{I}_m^n)_{i-1}, (\hat{I}_m^n)_i) < c \end{cases} \quad (17)$$

где M — выбранная метрика сравнения бинарных масок, а $c = \text{const} \in [0, 1]$ — фиксированное константное пороговое значение.

Описанное правило уже позволяет получить хороший результат в случае когда качество реконструкции с увеличением количества углов плавно улучшается.

Примерами таких изображений в наборе данных конвейера экспериментов являются объёмы "Кролик" и "Статуэтка".

В случаях, когда наблюдается резкий скачок качества при включении проекций под углами, совпадающими с основными структурными элементами исследуемого объекта, правило останова, описанное в формуле (17), может привести к слишком раннему завершению процесса. Это связано с тем, что до момента скачка несколько соседних реконструкций могут оказаться достаточно схожими между собой, несмотря на то, что качество ещё не достигло оптимального уровня.

Таковыми свойствами в наборе данных обладают изображения "Решётка" и "Наклонная решётка". В первом случае, благодаря ортогональной структуре плоскостей, приемлемое качество достигается уже при первом наборе углов. Во втором случае качество улучшается скачкообразно — при добавлении углов, совпадающих с наклонами отдельных

семейств плоскостей.

Такие ситуации можно учесть в правиле останова добавив дополнительное условие на минимальное количество углов в наборе.

Хотя в текущем виде правило останова не может сработать на самом первом наборе углов — для расчёта метрики требуется как минимум две реконструкции — этого недостаточно для устранения проблемы преждевременной остановки. Чтобы её избежать, необходимо дополнительно ограничить минимальный индекс, с которого правило начинает применяться.

Список литературы

1. Николаев Д. П. Методы и алгоритмы Хаф-анализа изображений в системах технического зрения : дис. ... канд. / Николаев Д. П. — 2023.
2. Zhang Y.-J. Advances in image and video segmentation. — IGI Global, 2006.
3. Wirjadi O. Survey of 3D image segmentation methods. — 2007.
4. Ханыков И. Классификация алгоритмов сегментации изображений // Известия высших учебных заведений. Приборостроение. — 2018. — Т. 61, № 11. — С. 978—987.
5. Adams R., Bischof L. Seeded region growing // IEEE Transactions on pattern analysis and machine intelligence. — 1994. — Т. 16, № 6. — С. 641—647.
6. Lin Z., Jin J., Talbot H. Unseeded Region Growing for 3D Image Segmentation. // ACM International Conference Proceeding Series. Т. 9. — Citeseer. 2000. — С. 31—37.
7. Segmentation and classification of lung tumor from 3D CT image using K-means clustering algorithm / P. Sarker [и др.] // 2017 4th international conference on advances in electrical engineering (ICAEE). — IEEE. 2017. — С. 731—736.
8. Comaniciu D., Meer P. Mean shift: A robust approach toward feature space analysis // IEEE Transactions on pattern analysis and machine intelligence. — 2002. — Т. 24, № 5. — С. 603—619.
9. Camilus K. S., Govindan V. A review on graph based segmentation // International Journal of Image, Graphics and Signal Processing. — 2012. — Т. 4, № 5. — С. 1.
10. Boykov, Kolmogorov. Computing geodesics and minimal surfaces via graph cuts // Proceedings Ninth IEEE international conference on computer vision. — IEEE. 2003. — С. 26—33.

11. Peng Z., Qu S., Li Q. Interactive image segmentation using geodesic appearance overlap graph cut // Signal Processing: Image Communication. — 2019. — Т. 78. — С. 159—170.
12. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in N-D images / Y. Zeng [и др.] // Computer vision and image understanding. — 2008. — Т. 112, № 1. — С. 81—90.
13. Felzenszwalb P. F., Huttenlocher D. P. Efficient graph-based image segmentation // International journal of computer vision. — 2004. — Т. 59. — С. 167—181.
14. Volumetric segmentation using Weibull E-SD fields / J. Hu [и др.] // IEEE Transactions on Visualization and Computer Graphics. — 2003. — Т. 9, № 3. — С. 320—328.
15. Ayed I. B., Hennane N., Mitiche A. Unsupervised variational image segmentation/classification using a Weibull observation model // IEEE transactions on Image processing. — 2006. — Т. 15, № 11. — С. 3431—3439.
16. A 3D convolutional neural network for volumetric image semantic segmentation / H. Lu [и др.] // Procedia Manufacturing. — 2019. — Т. 39. — С. 422—428.
17. Ха Л. М. Свёрточная нейронная сеть для решения задачи классификации // Труды Московского физико-технического института. — 2016. — Т. 8, 3 (31). — С. 91—97.
18. Milletari F., Navab N., Ahmadi S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation // 2016 fourth international conference on 3D vision (3DV). — Ieee. 2016. — С. 565—571.
19. Liu C. J., Wu X. L. Industrial computerized tomography images segmentation based on cellular neural networks // Applied Mechanics and Materials. — 2011. — Т. 66. — С. 2228—2235.
20. Lang T., Sauer T. AI-Supported Segmentation of Industrial CT Data // 11th Conf. Ind. Comput. Tomogr. — Wels. 2022.

21. Semi-automatic Segmentation of MRI Brain Metastases Combining Support Vector Machine and Morphological Operators. / G. Gonella [и др.] // IJCCI. — 2019. — С. 457—463.
22. Nagai Y., Ohtake Y., Suzuki H. SegMo: CT volume segmentation using a multi-level Morse complex // Computer-Aided Design. — 2019. — Т. 107. — С. 23—36.
23. 3D volume segmentation of MRA data sets using level sets: image processing and display / A. A. Farag [и др.] // Academic radiology. — 2004. — Т. 11, № 4. — С. 419—435.
24. Wagner F., Maas H.-G. A Comparative Study of Deep Architectures for Voxel Segmentation in Volume Images // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. — 2023. — Т. 48. — С. 1667—1676.
25. Marc Levoy. The Stanford 3D Scanning Repository. — 2023. — URL: <https://graphics.stanford.edu/data/3Dscanrep/> (дата обр. 13.06.2025).
26. A threshold selection method from gray-level histograms / N. Otsu [и др.] // Automatica. — 1975. — Т. 11, № 285—296. — С. 23—27.
27. Niblack W. An introduction to digital image processing. — Strandberg Publishing Company, 1985.
28. Sauvola J., Pietikäinen M. Adaptive document image binarization // Pattern recognition. — 2000. — Т. 33, № 2. — С. 225—236.
29. Adaptive local thresholding for detection of nuclei in diversity stained cytology images / N. Phansalkar [и др.] // 2011 International conference on communications and signal processing. — IEEE. 2011. — С. 218—220.
30. Николаев Д. П., Сараев А. А. Критерии оценки качества в задаче автоматизированной настройки алгоритмов бинаризации // Труды Института системного анализа Российской академии наук. — 2013. — Т. 63, № 3. — С. 85—94.

31. Taha A. A., Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool // BMC medical imaging. — 2015. — Т. 15. — С. 1—28.
32. Yeghiazaryan V., Voiculescu I. Family of boundary overlap metrics for the evaluation of medical image segmentation // Journal of Medical Imaging. — 2018. — Т. 5, № 1. — С. 015006—015006.
33. Fast and flexible X-ray tomography using the ASTRA toolbox / W. Van Aarle [и др.] // Optics express. — 2016. — Т. 24, № 22. — С. 25129—25147.
34. CuPy Developers. CuPy: A NumPy-compatible array library accelerated by CUDA. — 2024. — URL: <https://cupy.dev/> (дата обр. 13.06.2025).