

DOI: <https://doi.org/10.30898/1684-1719.2024.3.8>

УДК: 004.93:004.032.26

## МЕТОД СИНТЕЗА ДАННЫХ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Д.А. Шлёнских<sup>1</sup>, М.Л. Белокопытов<sup>1</sup>, Д.В. Анохин<sup>2</sup>, И.Г. Иванов<sup>3</sup>

<sup>1</sup> Военно-космическая академия имени А.Ф. Можайского,  
197198, Санкт-Петербург, ул. Ждановская, 13

<sup>2</sup> Санкт-Петербургский государственный университет  
аэрокосмического приборостроения,  
190000, Санкт-Петербург, ул. Большая Морская, 67

<sup>3</sup> Главный испытательный космический центр им. Г.С. Титова,  
143090, Краснознаменск, ул. Октябрьская, 3

Статья поступила в редакцию 05 февраля 2024 г.

**Аннотация.** Статья посвящена вопросам формирования исходного набора данных для обучения нейронных сетей. Представлен апробированный метод создания синтетических данных, построенный на базе графического процессора с использованием графического конвейера. Отличительной особенностью данного метода является его модульная архитектура, позволяющая легко модифицировать, удалять или добавлять отдельные этапы в конвейер генерирования синтетических изображений. Проведено обучение одноэтапного автоматического обнаружителя на базе свёрточной нейронной сети типа *Yolo*, а также произведена оценка качества обученной модели и работы алгоритма распознавания. Сделаны выводы относительно возможности применения такого подхода при создании репрезентативных выборок большого объёма и их дальнейшего использования для обучения нейронных сетей распознаванию образов.

**Ключевые слова:** нейронная сеть, обучение, синтетические данные, графический конвейер.

**Автор для переписки:** Белокопытов Марк Львович, [Hommer1990@mail.ru](mailto:Hommer1990@mail.ru)

## Введение

Обнаружение объектов в сложных условиях обстановки является фундаментальной задачей для любой системы технического зрения. Современные архитектуры свёрточных нейронных сетей (НС), такие как *ResNet*, *VGG*, *GoogleNet*, *DenseNet*, *Faster R-CNN*, *SSD*, *Yolo*, *RetinaNet*, достигли очень впечатляющих результатов в области распознавания образов. Однако обучение таких моделей с миллионами параметров требует огромного количества (десятки тысяч) маркированных обучающих данных для достижения конкурентных результатов [1].

Данные – один из самых ценных в наше время ресурсов. Однако большие временные затраты, связанные с обработкой и сбором реальных данных, необходимых для качественного обучения НС, зачастую ограничивают практическую реализацию систем технического зрения. В подобных случаях для подготовки моделей машинного обучения хорошей альтернативой могут стать синтетические данные (СД), т.е. искусственные данные, имитирующие наблюдения за объектами реального мира.

К обучению НС с использованием СД прибегают в тех случаях, когда получение реальных данных невозможно из-за сложности их сбора или дороговизны самого процесса добычи. Синтетические данные принято разделять на два типа, полные и частичные [2]. Полный тип состоит исключительно из СД, полученных в результате 3D моделирования. Примером такого типа данных может служить сгенерированное изображение квадрокоптера в симулируемом окружении (рисунок 1а). В свою очередь частичный тип включает в себя как синтетические, так и реальные данные, полученные путём наблюдения и измерения. Примером этого типа данных

является сгенерированное изображение, вставленное в реальную фотографию пейзажа (рисунке 1б).



1a



1б

Рис. 1. Пример синтетических данных.

На сегодняшний день существует огромное количество библиотек программирования для генерации СД, однако практически ни одна из них не позволяет эффективно и быстро реализовать процесс генерации искусственных изображений. Предлагаемый авторами в данной работе метод генерации СД построен на базе графического процессора (*GPU*) с использованием графического конвейера (*Pipeline*).

## 1. Исходные данные и постановка задачи

Графический конвейер является ключевым компонентом графической архитектуры, главная задача которого заключается в обработке данных и отображении полученных результатов [3]. Графический конвейер (ГК) представляет собой последовательность этапов обработки изображений, включая вершинный, растровый и пиксельный шейдеры, которые выполняют различные вычисления (растеризацию) для создания конечного изображения.

Использование *Pipeline* и *GPU* имеет ряд важных преимуществ. Прежде всего, графический процессор обладает огромным параллельным вычислительным потенциалом, что позволяет эффективно обрабатывать большое количество графических данных одновременно. Это свойство *GPU*

особенно актуально при рендеринге (визуализации) трехмерных сцен, в тех случаях, когда требуется обработка огромного массива графических элементов и текстур.

Кроме того, реализация параллельных вычислений на графическом процессоре позволяет значительно ускорить процесс обработки графических данных, что делает их более эффективными в данной предметной области по сравнению с общими центральными процессорами (*CPU*).

На рисунке 2 представлена упрощенная схема графического конвейера.

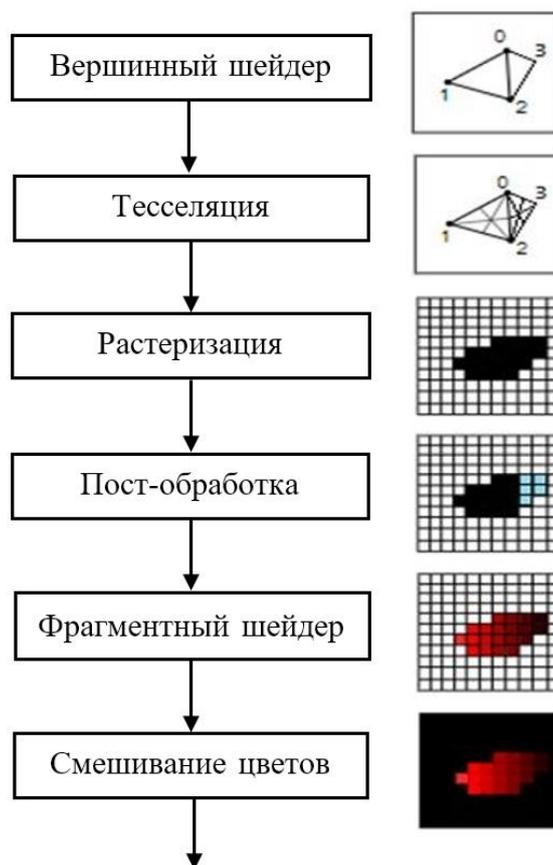


Рис. 2. Упрощенная схема графического конвейера.

Рассмотрим более подробно схему работы графического конвейера.

На первом этапе вершинный шейдер вызывается для каждой вершины и используется, в основном, для трансформации вершин из локальной системы координат в экранную. Далее с помощью шейдеров тесселяции производится разбиение примитивов (треугольников, квадров, полигонов) на меньшие геометрические формы. Данная технология позволяет создавать объекты сцены с более достоверными и прорисованными формами.

На этапе растеризации каждый примитив раскладывается на множество фрагментов, которые представляют собой пиксельные элементы, в последующем заполняющие примитивы во фреймбуфере – буфере памяти, который содержит данные обо всех пикселях полного кадра. Все фрагменты, которые выходят за пределы экрана, отсекаются, а атрибуты, переданные вершинным шейдером, интерполируются по фрагментам, как показано на рисунке 2.

На этапе пост-обработки применяются шейдеры, которые обрабатывают изображения после всех основных этапов рендеринга. Фрагментный шейдер вызывается для каждого фрагмента и определяет в какие фреймбуферы и с каким значением цвета и глубины записываются фрагменты.

На заключительном этапе происходит смешивание различных фрагментов, относящихся к одному и тому же пикселю во фреймбуфере. Фрагменты могут перекрывать друг друга или смешиваться в зависимости от прозрачности.

Главной особенностью *Pipeline* является то, что он работает параллельно и в многопоточном режиме, что позволяет обрабатывать большое количество графических данных одновременно и достигать высокой производительности при рендеринге 3D сцен.

После генерации конечного изображения для выделения объекта и создания его маски (матрицы принадлежности пикселей текущему объекту), используются так называемые пользовательские трафареты и материалы постобработки.

Пользовательский трафарет – это трафарет, который используется для определения того, какие пиксели визуализируются при растеризации (например, при обрисовке объектов с определенным ограничивающим контуром). Пользовательский трафарет может быть трафаретом, созданным для реализации конкретных эффектов или фильтров в графике. Обработка таких трафаретов обычно включается в стадию растеризации в *Pipeline*.

Материал постобработки – это компьютерная графика, реализуемая в виде наложения различных эффектов и фильтров на изображения

после их рендеринга. Применение компьютерной графики позволяет изменить визуальное восприятие изображения или видео. Постобработка применяется, как правило, в конце графического конвейера, после выполнения основных этапов рендеринга 3D сцены. Данный этап носит название «постпроцессинг».

В реализованном авторами методе генерации СД пользовательский трафарет используется для выделения объекта сегментации и последующего создания маски с помощью материала постобработки.

В результате работы графического конвейера получается конечное изображение, а также текстура маски объекта сегментации, которые могут быть отображены на экране (рисунок 3).

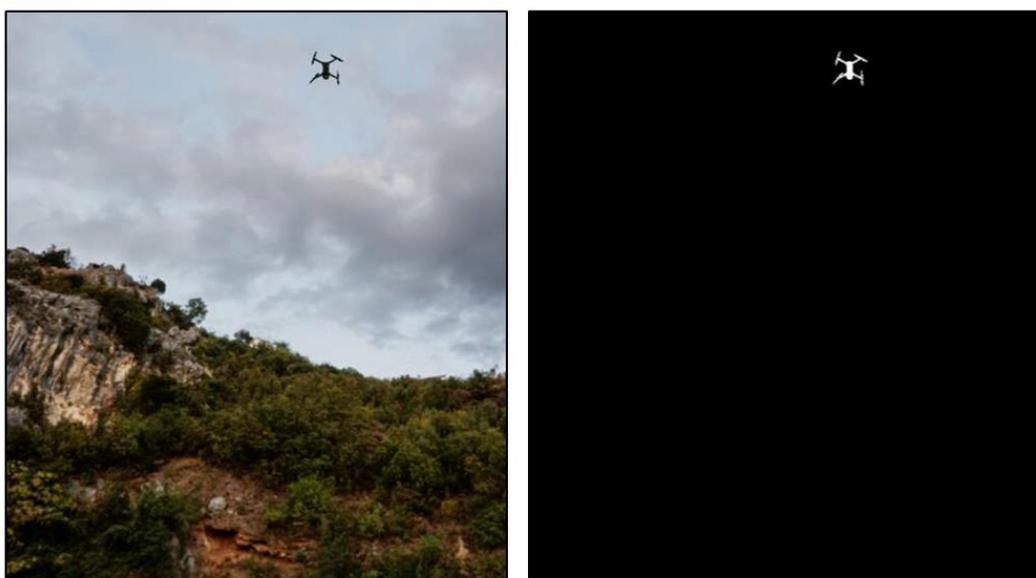


Рис. 3. Пример конечного изображения и маски объекта, полученные с использованием графического конвейера.

Для повышения эффективности обучения НС с помощью сгенерированных СД необходимо реализовать процедуру преобразования маски в контур (осуществить кодирование контуров), что позволит максимизировать точность дальнейшей работы нейронной сети по распознаванию объектов.

Одним из наиболее распространённых методов кодирования контуров является цепной код Фримена [4]. Цепные коды применяются для конвертации границы в последовательность прямых определенной длины и указателя направления. В основе данной конвертации лежит 2-х или 3-х битная решетка

на 4 и 8 направлений соответственно. Длина определяется разрешением решетки, а направление – указателем из решетки (рисунок 4).

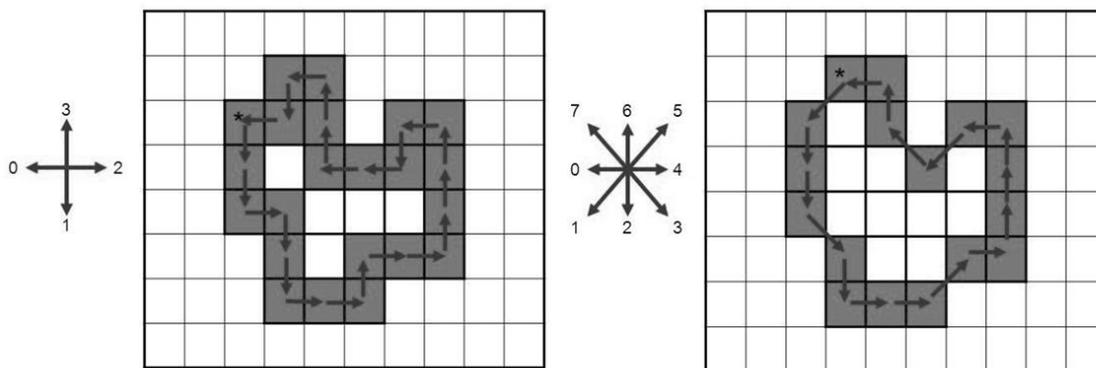


Рис. 4. Пример работы цепного кода Фримена на 4 и 8 направлений (на рисунке «звёздочкой» отмечена начальная точка, стрелкой – элементарный вектор).

На контуре фиксируется точка, которая называется начальной. Затем контур обходится, и каждый вектор смещения записывается комплексным числом  $a + ib$ , где  $a$  – смещение точки по оси  $X$ , а  $b$  – смещение по оси  $Y$ . Пример записи координат элементарного вектора при комплекснозначном кодировании представлен на рисунке 5.

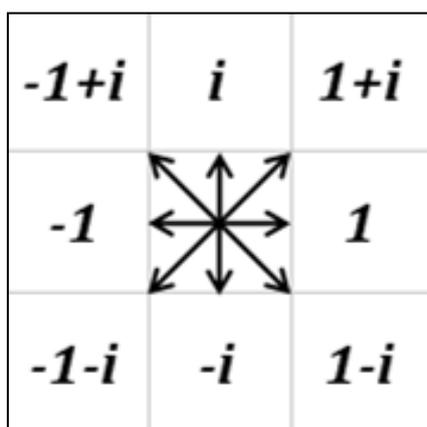


Рис. 5. Координаты элементарного вектора при комплекснозначном кодировании.

Смещение берется относительно предыдущей точки. При таком подходе контур определяется как совокупность элементарных векторов, представленных своими двумерными координатами. Изменение начальной точки приведет

к циклическому сдвигу вектор-контура, а изменение масштаба изображения можно рассматривать как умножение элементарного вектора на масштабный коэффициент.

Затем производится сжатие горизонтальных, вертикальных и диагональных сегментов до их конечных точек с целью уменьшения избыточности информации и ускорения дальнейшей обработки.

В последующем для обучения НС результаты работы кода Фримена необходимо перевести в формат метки набора данных путем нормализации координат точек контура и назначением им класса объекта.

## 2. Обучение нейронных сетей на синтетических данных

В настоящей работе авторами проводилось обучения одноэтапного автоматического обнаружителя на базе свёрточной нейронной сети типа *Yolo*. Обучение производилось без учителя, в условиях, когда испытываемая система спонтанно обучалась выполнять поставленную задачу без вмешательства со стороны экспериментатора. Такой тип обучения пригоден только для задач, в которых известны описания множества объектов (обучающей выборки) и требуется обнаружить внутренние взаимосвязи, зависимости и закономерности, существующие между объектами [5].

Для формирования обучающей и тестовой выборок набор сгенерированных описанным выше методом СД был разбит в соотношении 80% на 20% соответственно.

Для оценки качества обученной модели и работы алгоритма использовались метрики: *precision*, *recall*, *mAP@0.5* и *F*-метрика [6].

В таблице 1 приведены значения полученных метрик, характеризующих качество обученной нейросетевой модели.

Таблица 1. Оценка качества обученной нейросетевой модели.

Метрика	<i>Precision</i>	<i>Recall</i>	<i>mAP@0.5</i>	<i>F1</i>
Значение точности	0.857	0.89	0.912	0.87

Из таблицы видно, что точность модели на тестовом наборе данных находится на приемлемом уровне, модель показывает хорошее обобщение на новых данных.

После окончания процесса обучения была произведена валидация, т.е. проверка правильности работы (предсказательной способности) аналитической модели [7]. Она проводилась на независимом (то есть на не использовавшемся для обучения и тестирования) валидационном множестве, которое, в данном конкретном примере, представляло собой множество изображений квадрокоптеров, полученных натурным путём.

На рисунке 5 представлены результаты работы обученной нейронной сети по распознаванию квадрокоптеров.



Рис. 5. Результаты работы обученной нейронной сети на натуральных данных.

Натурные данные были получены с использованием видеокамеры *Megix DV 4K* и квадрокоптера типа *ZLRC Beast SG906 PRO*.

## **Заключение**

Автоматизация комплексной обработки исходных данных на этапах обнаружения и распознавания является в настоящее время важной и актуальной задачей при мониторинге того или иного объекта интереса. Одним из перспективных направлений такого рода автоматизации является

реализация процессов обработки получаемых изображений с использованием методов и технологий искусственного интеллекта на основе сверхточных нейронных сетей. При этом подготовка набора обучающих данных является одной из главных задач при реализации нейросетевой модели.

Однако ввиду ряда ограничений (сложность добычи, отсутствие необходимой аппаратуры съёмки и т.д.) зачастую использовать натурные данные для обучения нейронных сетей не представляется возможным. Для решения этой проблемы предлагается использовать синтетические данные, полученные искусственным путём.

В настоящей работе авторами представлен метод создания синтетических данных для обучения нейронных сетей, ключевым элементом которого является графический конвейер. Предлагаемый метод генерации позволяет создавать большие объёмы синтезированных изображений с достаточно высокой детализированностью реальных сцен и объектов.

Апробация данного метода была осуществлена на базе сверточной нейронной сети *Yolo*, которая была обучена (исключительно на синтезированных изображениях) распознаванию квадрокоптеров.

## Литература

1. Пчелинцев С.Ю., Юляшков М.А., Ковалева О.А. Метод создания синтетических наборов данных для обучения нейросетевых моделей распознаванию объектов // Информационно-управляющие системы. – 2022. – №. 3 (118). – С. 9-19. <http://i-us.ru/index.php/ius/article/view/15242>.
2. Сообщество IT-специалистов: Синтетические данные для машинного обучения: их природа, типы и способы генерации: сайт. – Москва, 2019. – URL: <https://habr.com/ru/articles/721170> (дата обращения: 21.11.2023). – Текст: электронный.

3. Гонахчян В.И. Модель производительности графического конвейера для однопроходной схемы рендеринга динамических трехмерных сцен // Труды Института системного программирования РАН. – 2020. – Т. 32. – №. 4. – С. 53-72.
4. Леденёв М.А. Программный модуль повышения качества изображений в задачах распознавания // Известия Транссиба. – 2015. – №. 2 (22). – С. 110-115.
5. Малыхина Г.Ф., Меркушева А.В. Элементы статистической концепции обучения нейронной сети и прогнозирование точности ее функционирования // Научное приборостроение. – 2005. – Т. 15. – №. 1. – С. 29-45.
6. Белокопытов М.Л. и др. Мониторинг объектов морского судоходства по аэрокосмическим данным дистанционного зондирования в СВЧ диапазоне с применением нейросетевых технологий // Журнал радиоэлектроники. – 2022. – №. 4. <https://doi.org/10.30898/1684-1719.2022.4.2>.
7. Носырев П.В. и др. Валидация аналитических методик: теория и практика (Часть I) // Ремедиум. Журнал о российском рынке лекарств и медицинской технике. – 2003. – №. 10. – С. 69-71.

**Для цитирования:**

Шлёнских Д.А., Белокопытов М.Л., Анохин Д.В. Иванов И.Г. Метод синтеза данных для повышения эффективности обучения нейронных сетей. // Журнал радиоэлектроники. – 2024. – №. 3. <https://doi.org/10.30898/1684-1719.2024.3.8>