

*Печин Н.В., магистрант
институт информационных технологий
МИРЭА — Российский Технологический Университет
РФ, г. Москва*

*Научный руководитель: Плотников С.Б., к.т.н.
МИРЭА — Российский Технологический Университет
РФ, г. Москва*

ЗАДАЧА РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ КРУГОВОГО ОБЗОРА

Аннотация. В работе исследуется задача разработки информационной системы кругового обзора для строительной техники. Обосновывается недостаточность существующих решений, которые, как правило, обеспечивают либо многокамерную визуализацию, либо ограниченные функции анализа сцены, но не сочетают низкую задержку, панорамное представление и программную гибкость. Рассматривается система кругового обзора как программно-аппаратное решение, обеспечивающее прием видеопотоков с нескольких камер, их геометрическую коррекцию, проекцию на опорную плоскость, сшивку и отображение в формате birdview. Результаты анализа применяются к постановке задачи разработки системы для строительной техники с возможностью последующего обнаружения людей и препятствий в рабочей зоне.

Ключевые слова: круговой обзор, строительная техника, birdview, видеопоток, компьютерное зрение, Camera2 API, OpenCV, Android, детекция объектов, промышленная безопасность.

Pechin N.V., student
Department of Information Systems
MIREA — Russian Technological University
Russia, Moscow

Scientific Supervisor: Plotnikov S.B., Candidate of Sciences in Technology
MIREA — Russian Technological University
Russia, Moscow

DEVELOPMENT OF AN INTELLIGENT CIRCULAR REVIEW SYSTEM: A TASK STATEMENT ARTICLE

Abstract. The paper addresses the problem of developing an around-view information system for construction machinery. It substantiates the insufficiency of existing solutions, which typically provide either multi-camera visualization or limited scene analysis capabilities, but do not combine low latency, panoramic representation, and software flexibility. The around-view system is considered as a hardware-software solution that provides acquisition of multiple video streams, their geometric correction, projection onto a reference plane, stitching, and birdview visualization. The results of the analysis are applied to the problem statement for developing a system for construction machinery with the possibility of subsequent detection of people and obstacles in the working area.

Keywords: around-view system, construction machinery, birdview, video stream, computer vision, Camera2 API, OpenCV, Android, object detection, industrial safety.

Введение.

Повышение безопасности эксплуатации строительной техники остается одной из наиболее значимых задач на стыке промышленной безопасности, мобильных вычислений и компьютерного зрения. Работа экскаваторов, погрузчиков, автокранов, самосвалов и другой крупногабаритной техники ведется в среде, где оператор вынужден принимать решения в условиях неполной визуальной информации. Даже при наличии зеркал и стандартных камер заднего вида значительная часть пространства вокруг машины остается вне непрерывного контроля, особенно вблизи корпуса, по бокам и в зонах, закрываемых навесным оборудованием. В результате растет вероятность столкновений с объектами инфраструктуры, другой техникой или персоналом, находящимся в опасной близости к траектории движения машины.

С технической точки зрения эта проблема не сводится к простой установке нескольких камер. Многокамерное видеонаблюдение само по себе не гарантирует уменьшения риска, если изображение поступает разрозненно, имеет выраженные геометрические искажения, воспринимается оператором с высокой когнитивной нагрузкой или не сопровождается автоматическим обнаружением опасных объектов. Поэтому современная система кругового обзора должна решать не одну, а сразу несколько взаимосвязанных задач: обеспечивать непрерывный прием видеопотоков, выполнять геометрическую коррекцию изображений, приводить их к единой системе координат, формировать обобщенное представление окружающего пространства и, при необходимости, выделять объекты, представляющие непосредственную угрозу.

Android-интерфейс Camera2 предоставляет детальный контроль над камерой и предназначен для сложных сценариев работы с видеоданными [1]. Геометрическая часть такой системы естественным образом опирается

на средства OpenCV, в том числе на модель fisheye-камер и процедуры калибровки [2], [3].

В практическом смысле задача разработки системы кругового обзора для строительной техники определяется противоречием между эксплуатационными требованиями и возможностями доступных решений. С одной стороны, пользователю необходимы низкая задержка, полнота обзора, устойчивость к тяжелым условиям работы и возможность последующего наращивания функций. С другой стороны, значительная часть представленных на рынке систем ориентирована либо на закрытую промышленную архитектуру, либо на легковой транспорт и парковочные сценарии, что ограничивает их применимость в строительной среде.

Цель настоящей работы – формулирование Технического задания для создания системы гибкой программно-расширяемой платформы кругового обзора строительной техники, которая должна опираться на параметры, поддающиеся оценке: полноту обзора, задержку визуализации, частоту кадров, точность геометрического преобразования, точность детекции объектов и степень зависимости программной части от конкретной аппаратной конфигурации.

1. Анализ существующего уровня решений и постановка противоречия

Существующие решения в области кругового обзора условно можно разделить на две группы. К первой относятся промышленные системы, ориентированные на тяжелую технику и эксплуатацию в неблагоприятных условиях. Ко второй — потребительские и полупрофессиональные системы birdview. Их сравнительный анализ позволяет точнее определить границы проектируемого объекта.

Характерным представителем промышленного класса является система Orlaco SurroundView [4]. Основное достоинство решения

заключается в надежной многокамерной визуализации и его ориентации на эксплуатацию в составе тяжелой техники. Однако сама логика такой системы ориентирована прежде всего на надежный обзор, а не на глубокую программную переработку видеоданных. Следовательно, при всех преимуществах в части готовности к эксплуатации подобный подход не в полной мере отвечает задаче создания гибкой программно-расширяемой платформы.

Другой класс решений представляет система «Цифра. Обзор 360», позиционируемая как программно-аппаратное решение для контроля пространства вокруг горнотранспортной техники и предотвращения столкновений. Официальное описание подчеркивает, что система устраняет слепые зоны, оповещает оператора об опасных сближениях, фиксирует действия оператора, формирует видеоархив и используется как элемент активной безопасности предприятия [5]. Такой подход значительно ближе к задачам промышленной аналитики, поскольку в нем круговой обзор уже сочетается с предупреждениями и накоплением событийных данных.

Третью группу образуют системы birdview, распространенные в автомобильном сегменте. Например, в описании 3D CAN Birdview 360 указывается, что изображения с четырех камер преобразуются в единую картинку вида сверху, а сама система поддерживает несколько режимов отображения [6]. Такие решения важны тем, что они демонстрируют практическую работоспособность схемы «четыре камеры — геометрическое преобразование — birdview-визуализация». Однако именно автомобильная специализация ограничивает их применимость в строительной отрасли.

Проведенный анализ показывает, что на рынке уже существуют отдельные технологические компоненты, необходимые для построения системы кругового обзора: многокамерный захват, объединение

изображений, предупреждение об опасных сближениях, регистрация событий. Однако эти компоненты редко объединяются в рамках единой открытой и масштабируемой архитектуры. Именно в этом заключается основная цель: требуется система, которая не повторяет уже имеющиеся продукты, а сочетает их сильные стороны без воспроизведения их архитектурных ограничений.

2. Обоснование актуальности разработки и целевых параметров

Актуальность разработки современной системы кругового обзора для строительной техники определяется не только общим тезисом о необходимости повышения безопасности, но и конкретными техническими параметрами, которые требуется улучшить по сравнению с существующей практикой. Прежде всего речь идет о полноте визуального контроля окружающего пространства. Для тяжелой машины важен не просто факт наличия камер, а формирование целостного представления рабочей зоны без критически выраженных слепых зон в непосредственной близости от корпуса. Это означает, что система должна обеспечивать согласованное использование не менее четырех широкоугольных камер, каждая из которых покрывает свою часть сцены и после преобразования вносит вклад в единое панорамное изображение.

Вторым ключевым параметром является минимизация задержки вывода изображения. Для оператора промышленной техники обзор имеет ценность только тогда, когда наблюдаемое изображение соответствует текущему положению объектов, а не их состоянию с заметным опозданием. Поэтому целевая задержка панорамной визуализации должна быть ограничена значением порядка 150 мс, а режим работы — поддерживать не менее 25–30 кадров в секунду.

Третьим параметром становится геометрическая достоверность результирующего изображения. В условиях использования

широкоугольной или fisheye-оптики недостаточно лишь принять видеопоток и вывести его на экран. Необходимо компенсировать оптические искажения, оценить внутренние параметры камер и выполнить преобразование изображений к общей плоскости наблюдения. В OpenCV для этого предусмотрены как процедуры классической калибровки, так и средства работы с fisheye-моделью, где вычисления основываются на внутренних и внешних параметрах камеры и соответствиях между пространственными точками и точками изображения [2], [3]. Следовательно, одним из проектных требований становится обеспечение корректной геометрической подготовки изображений до этапа сшивки.

Наконец, для современных систем наблюдения все большее значение приобретает интеллектуальный анализ сцены. Оператору недостаточно просто видеть окружение; ему необходимо оперативно понимать, где именно находится источник опасности. В официальной документации Ultralytics объектная детекция определяется как задача идентификации расположения и класса объектов на изображении или в видеопотоке [7], а в документации Google AI Edge подчеркивается, что детекторы объектов способны определять наличие известных классов объектов и предоставлять информацию об их положении в кадре или видеоряде [9]. В контексте строительной техники это означает возможность автоматического выделения людей, техники или иных препятствий в опасной зоне и формирования предупреждения до того, как оператор успеет визуально обнаружить их самостоятельно. Отсюда вытекает требование к включению в архитектуру модуля компьютерного зрения, рассчитанного на работу на edge-устройстве.

3. Постановка задачи разработки

С учетом рассмотренных предпосылок задача разработки может быть сформулирована следующим образом. Требуется создать

информационную систему кругового обзора для строительной техники, обеспечивающую одновременный прием видеопотоков не менее чем с четырех широкоугольных камер, их синхронную обработку, коррекцию дисторсии, преобразование к общей опорной плоскости и формирование единого панорамного изображения окружающего пространства в реальном времени. Система должна функционировать на универсальной программной платформе, поддерживать визуализацию для оператора и предоставлять основу для последующего анализа сцены средствами компьютерного зрения.

Из этой формулировки непосредственно следуют частные подзадачи. Первая подзадача связана с организацией устойчивого многокамерного приема видеоданных и выбором интерфейса, позволяющего управлять конфигурацией камер на низком уровне. Здесь оправдано использование Camera2 API, поскольку данный пакет Android предназначен для работы с отдельными устройствами камеры и обеспечивает детализированный контроль, необходимый для сложных сценариев захвата [1]. Вторая подзадача заключается в калибровке камер и компенсации оптических искажений, без чего последующая проекция на плоскость и сшивка приводили бы к накоплению геометрической ошибки. Третья подзадача состоит в разработке механизма формирования birdview-представления, уменьшающего когнитивную нагрузку на оператора по сравнению с отображением четырех независимых изображений. Четвертая подзадача — интеграция модуля обнаружения людей и препятствий в видеопотоке. Пятая — разработка архитектуры, при которой перечисленные подсистемы остаются модульными и допускают развитие без полной переработки приложения.

С инженерной точки зрения разрабатываемый объект должен отличаться от существующих аналогов согласованностью параметров. Требуется получить платформу, где многокамерная визуализация,

геометрическая обработка, рендеринг и интеллектуальный анализ работают как единая цепочка.

4. Технологическая основа решения

В качестве программной платформы целесообразно использовать Android, поскольку эта среда предоставляет зрелый стек для работы с камерами, графикой и локальными вычислениями на мобильных и встраиваемых устройствах. Для управления захватом видеоданных рационально использовать Camera2 API [1], а для графического вывода — OpenGL ES, который в официальной документации Android рассматривается как стандартный механизм высокопроизводительной двумерной и трехмерной графики для встраиваемых устройств [8].

Геометрическая обработка изображений должна опираться на OpenCV, поскольку эта библиотека предоставляет необходимые средства для калибровки, коррекции дисторсии, работы с широкоугольной оптикой и вычисления преобразований, связывающих изображение с наблюдаемой плоскостью [2], [3]. За счет этого достигается не только улучшение визуального качества, но и повышение метрической согласованности результирующего изображения.

Подсистема анализа видеоданных должна строиться на моделях детекции объектов, пригодных для работы на edge-устройствах в режиме, близком к реальному времени. Семейство YOLO широко используется именно в сценариях, где требуется сочетание скорости и приемлемой точности, а официальная документация Ultralytics прямо указывает на ориентацию детекторов на обработку изображений и видеопотока с выдачей ограничивающих рамок и классов объектов [7]. Для практического встраивания таких моделей в конечную систему оправдано использование средств запуска on-device-моделей, ориентированных на edge-сценарии [9].

Принципиально важным архитектурным требованием становится модульность. Подсистема захвата, подсистема геометрической обработки, подсистема сшивки, подсистема детекции и пользовательский интерфейс должны проектироваться как относительно независимые компоненты. Только при таком подходе можно изменять модель детекции, тип камеры или способ визуализации без полной перестройки всей системы. Именно модульность обеспечивает возможность масштабирования решения на различные типы техники, наборы камер и целевые вычислительные платформы.

5. Реализация функции вывода изображения

Одним из наиболее важных аспектов в разработке является реализация модуля вывода изображения и его захвата для передачи в дальнейшую обработку. Для работы с камерами в проекте используется встроенная в ОС Андроид библиотека Camera2 API [1]. По сравнению с другими различными решениями, такими как Camera1 или CameraX [10], она дает возможность наиболее гибкой настройки аппаратной части камеры, а также, более оптимизированные подходы в работе с захваченными изображениями с камер, что дает нам конкурентное преимущество в сравнении с другими системами.

Далее, рассмотрим основные фрагменты моей реализации, которые играют ключевую роль в получении и передаче изображения дальше в обработку.

Открытие камеры и выбор параметров происходят в методе `openCamera()`. Он выполняет выбор камеры и её конфигурацию: определяется идентификатор камеры, считываются характеристики устройства, выбирается разрешение видеопотока, инициируется асинхронное открытие камеры (Листинг 5.1).

Листинг 5.1 – Метод открытия камеры и выбора ее параметров

```
private fun openCamera() {  
    try {
```

```

        val cameraId = cameraManager.cameraIdList[targetCameraId]
        val characteristics =
cameraManager.getCameraCharacteristics(cameraId)
        val map =
characteristics.get(CameraCharacteristics.SCALER_STREAM_CONFIGURAT
ION_MAP)
        previewSize =
map?.getOutputSizes(TextureView::class.java)?.maxByOrNull {

```

Продолжение Листинга 5.1

```

        it.width * it.height
    }
    cameraManager.openCamera(cameraId, cameraStateCallback,
handlerCamera)
    // Log available sizes and fps ranges for diagnostics

    val sizes =
map?.getOutputSizes(ImageFormat.YUV_420_888) val ranges =
characteristics.get(CameraCharacteristics.CONTROL_AE_AVAILABLE_TAR
GET_FPS_RANGES)
}

```

Для извлечения кадров используется объект ImageReader, настроенный на формат YUV_420_888, который является стандартом для дальнейшей обработки изображений (Листинг 5.2).

Листинг 5.2 – Функция извлечения кадров для передачи в дальнейшую обработку

```

fun setupImageReader(w: Int, h: Int) {
    imageReader = ImageReader.newInstance(
        w,
        h,
        ImageFormat.YUV_420_888,
        3
    )
    imageReader.setOnImageAvailableListener({ reader ->
        val image = try {
            reader.acquireLatestImage()
        } catch (_: Exception) {
            null
        }
        if (image != null) {
            computingFPSCamera.handleCamera(image)
            image.close()
        }
    }, handlerCamera)
}

```

Таким образом, данная программная реализация служит основной всей системы кругового обзора, так как для работы системы нам необходимо получить, обработать и отобразить видеопоток с камер.

6. Заключение

Постановка задачи разработки информационной системы кругового обзора для строительной техники определяется необходимостью устранить ограниченность эксплуатируемых в настоящее время средств визуального контроля и обеспечить оператору целостное, оперативное и интерпретируемое представление об окружающем пространстве. Анализ существующих решений показывает, что современный рынок располагает отдельными зрелыми компонентами — многокамерным обзором, birdview-визуализацией, промышленной интеграцией и предупреждением о сближениях, — однако их сочетание в рамках единой программно-расширяемой архитектуры остается недостаточно проработанным.

Исходя из этого, целевая система должна обеспечивать прием видеопотоков с нескольких широкоугольных камер, коррекцию дисторсии, приведение изображений к общей системе координат, формирование панорамного изображения и автоматическое обнаружение опасных объектов в зоне работы машины. Ключевыми целевыми характеристиками выступают полнота обзора, низкая задержка визуализации, устойчивый режим реального времени, геометрическая достоверность результирующего изображения и возможность дальнейшего наращивания аналитических функций. Достижение этих характеристик связывается с использованием Camera2 API, OpenCV, OpenGL ES и детекторов объектов, ориентированных на edge-вычисления.

Таким образом, задача разработки состоит в проектировании интегрированной информационной системы, в которой многокамерный обзор становится основой для последующей интеллектуальной поддержки оператора. Именно это направление представляется наиболее перспективным для повышения безопасности эксплуатации строительной

техники и для дальнейшего развития мобильных промышленных систем компьютерного зрения.

Список литературы:

1. Android Developers. Camera2 overview [Электронный ресурс]. — URL: <https://developer.android.com/media/camera/camera2> (дата обращения: 10.04.2026).
2. OpenCV Documentation. Fisheye camera model [Электронный ресурс]. — URL: https://docs.opencv.org/4.x/db/d58/group__calib3d__fisheye.html (дата обращения: 10.04.2026).
3. OpenCV Documentation. Camera Calibration [Электронный ресурс]. — URL: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (дата обращения: 11.04.2026).
4. Stoneridge Orlando. SurroundView: 360° visibility for heavy machinery [Электронный ресурс]. — URL: <https://stoneridge-orlaco.com/en/systems/surroundview> (дата обращения: 12.04.2026).
5. ГК «Цифра». Обзор 360 [Электронный ресурс]. — URL: <https://www.zyfra.com/product/overview360/> (дата обращения: 13.04.2026).
6. BIRDDVIEW 360°. Система кругового обзора для авто 3D CAN BIRDDVIEW 360° [Электронный ресурс]. — URL: <https://car360view.ru/birdview-3d-can.html> (дата обращения: 13.04.2026).
7. Android Developers. OpenGL ES [Электронный ресурс]. — URL: <https://developer.android.com/develop/ui/views/graphics/opengl/about-opengl> (дата обращения: 15.04.2026).
8. Ultralytics Docs. Object Detection [Электронный ресурс]. — URL: <https://docs.ultralytics.com/tasks/detect/> (дата обращения: 18.04.2026).
9. Google AI Edge. Integrate object detectors [Электронный ресурс]. — URL:

https://www.tensorflow.org/lite/inference_with_metadata/task_library/object_detector (дата обращения: 18.04.2026).

10. Android Developers. CameraX overview [Электронный ресурс].
— URL: <https://developer.android.com/media/camera/cameras> (дата обращения: 19.04.2026).