

Санкт-Петербургский политехнический университет Петра Великого  
Высшая школа прикладной математики и вычислительной физики  
Физико-механический институт

# **Курсовая работа**

## **по дисциплине «Автоматизация научных исследований»**

# Генерация вводной и теоретической части научного исследования в виде статьи на тему бакалаврской ВКР

## Тема ВКР – «Аппаратная реализация фильтра для повышения резкости видеоизображения»

## Выполнил

студент гр. № 5040102/50201

Колесников Е. В.

## Преподаватель:

Новиков Ф. А.

# Санкт-Петербург 2026

## **СОДЕРЖАНИЕ**

**АННОТАЦИЯ**

**ABSTRACT**

**ВВЕДЕНИЕ.....Ошибка! Закладка не определена.**

**ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ ФИЛЬТРОВ НА ПЛИС .....Ошибка!  
Закладка не определена.**

**1.1 Типы фильтров обработки видеоизображений на ПЛИС.....Ошибка!  
Закладка не определена.**

**1.2 Применения ПЛИС в реальных устройствахОшибка! Закладка не  
определенна.**

**1.3 Преимущества использования ПЛИС для обработки видеопотока  
.....Ошибка! Закладка не определена.**

**1.4 Фильтры обработки видео на видеокамерахОшибка! Закладка не  
определенна.**

**ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О ПОВЫШЕНИИ РЕЗКОСТИ  
ВИДЕОИЗОБРАЖЕНИЯ.....Ошибка! Закладка не определена.**

**2.1Формат видео UYVY .....Ошибка! Закладка не определена.**

**2.1.1 Краткая история появления видеоформатаОшибка! Закладка не  
определенна.**

**2.1.2 Структура формата.....Ошибка! Закладка не определена.**

**2.2 Сравнение и выбор лучшего для ПЛИС методаОшибка! Закладка не  
определенна.**

**ЗАКЛЮЧЕНИЕ**

## **АННОТАЦИЯ**

В работе рассматривается аппаратная реализация фильтра повышения резкости видеоизображения на платформе FPGA для видеопотоков в формате UYVY (YUV 4:2:2). Исследование включает анализ существующих алгоритмов фильтрации, выбор оптимального метода и проектирование конвейерной архитектуры с эффективной организацией буферов строк для обработки яркостной компоненты Y. Особое внимание уделено рациональному использованию логических ресурсов и DSP-блоков ПЛИС, минимизации задержки и обеспечению работы в реальном времени. Предложенное решение демонстрирует высокую производительность при потоковой обработке HD и Full HD видео и может быть применено в интеллектуальных камерах, системах видеонаблюдения и встроенных видеоустройствах.

## **ABSTRACT**

The paper addresses the hardware implementation of a video image sharpening filter on an FPGA platform for UYVY (YUV 4:2:2) video streams. The study includes an analysis of existing filtering algorithms, the selection of an optimal method, and the design of a pipelined architecture with efficient line buffer organization for processing the luminance component Y. Special attention is given to the rational utilization of FPGA logic resources and DSP blocks, minimization of latency, and ensuring real-time operation. The proposed solution demonstrates high performance in streaming HD and Full HD video and can be applied in smart cameras, video surveillance systems, and embedded video devices.

## **ВВЕДЕНИЕ**

Современные системы обработки видеоизображений находят широкое применение в задачах технического зрения, промышленной автоматизации, системах видеонаблюдения, медицинской визуализации и бортовых комплексах реального времени. Повышение требований к качеству визуальной информации, достоверности распознавания объектов и устойчивости алгоритмов анализа изображений обуславливает необходимость эффективной предварительной обработки видеопотока. Одной из ключевых операций такой обработки является фильтрация повышения резкости, направленная на усиление локальных перепадов яркости и восстановление визуально значимых границ объектов.

Алгоритмы повышения резкости традиционно реализуются программно на центральных (CPU) или графических процессорах (GPU). Однако в задачах реального времени с жёсткими ограничениями по задержке, энергопотреблению и пропускной способности канала передачи данных программные решения не всегда обеспечивают требуемую производительность. Встраиваемые системы, включая интеллектуальные камеры и специализированные видеомодули, предъявляют требования к обработке видеопотока без буферизации целого кадра, с минимальной латентностью и гарантированной детерминированностью. В этих условиях особую актуальность приобретает аппаратная реализация алгоритмов обработки изображений на базе программируемых логических интегральных схем (FPGA).

Платформы FPGA, такие как решения семейства Xilinx и Intel, предоставляют широкие возможности для параллельной обработки потоковых данных, реализаций конвейерных архитектур и оптимизации вычислений на уровне аппаратной логики. В отличие от универсальных процессоров, FPGA позволяют проектировать специализированные вычислительные структуры, адаптированные под конкретный алгоритм фильтрации, что обеспечивает

высокую пропускную способность при относительно низком энергопотреблении. Это делает их предпочтительной платформой для реализации видеофильтров в системах с высокими требованиями к реальному времени.

Особый интерес представляет обработка видео в формате UYVY (YUV 4:2:2), широко применяемом в интерфейсах цифрового видеозахвата и передачи данных. Формат UYVY характеризуется раздельным представлением яркостной (Y) и цветоразностных (U, V) компонент, при этом хроминансные компоненты субдискретизированы по горизонтали. Такая организация данных требует учёта особенностей пространственного расположения компонент при проектировании фильтра. В задачах повышения резкости целесообразно выполнять обработку преимущественно над яркостной составляющей, поскольку именно она определяет пространственную структуру изображения и воспринимаемую чёткость, в то время как избыточное усиление цветовых компонент может приводить к появлению цветовых артефактов.

Фильтры повышения резкости, как правило, основываются на операциях пространственной свёртки с ядрами, усиливающими высокочастотные составляющие сигнала. Классические методы включают использование лапласиана, высокочастотных масок, а также алгоритмов типа unsharp masking. Аппаратная реализация таких методов требует эффективной организации буферов строк (line buffers), формирования скользящего окна обработки и оптимизации операций умножения и сложения с учётом ограниченных ресурсов FPGA (логических элементов, блоков памяти и DSP-модулей). Кроме того, при работе с потоковым видео необходимо обеспечить корректную обработку граничных областей кадра и соблюдение временных параметров интерфейса передачи данных.

Несмотря на наличие большого количества публикаций, посвящённых алгоритмам повышения резкости, вопросы их оптимальной аппаратной реализации для потокового UYVY-видео остаются актуальными. Требуется комплексный подход, учитывающий особенности формата данных,

требования к пропускной способности (вплоть до HD и Full HD разрешений), ограничения по ресурсам конкретной FPGA и необходимость интеграции фильтра в существующий видеотракт.

В настоящей работе рассматривается разработка и исследование аппаратной реализации фильтра повышения резкости видеоизображения в формате UYVY на FPGA. Основное внимание уделяется выбору алгоритма фильтрации, архитектуре потоковой обработки, организации памяти для формирования двумерного окна свёртки, а также оценке использования аппаратных ресурсов и временных характеристик. Целью исследования является создание эффективного и масштабируемого решения, обеспечивающего обработку видеопотока в реальном времени с минимальной задержкой и рациональным использованием логических и вычислительных ресурсов программируемой логики.

## **АКТУАЛЬНОСТЬ ИССЛЕДОВАНИЯ**

Стремительное развитие систем компьютерного зрения, интеллектуального видеонаблюдения и автономных технических комплексов обуславливает рост требований к качеству и информативности видеоизображений в реальном времени. Повышение резкости является одной из ключевых операций предварительной обработки, непосредственно влияющей на точность последующих алгоритмов детектирования, сегментации и распознавания объектов. В условиях ограниченных вычислительных ресурсов и жёстких требований к задержке обработки программные методы не всегда обеспечивают необходимую производительность и энергоэффективность. Аппаратная реализация фильтра повышения резкости на FPGA позволяет организовать параллельную потоковую обработку видеоданных с детерминированной латентностью и минимальной буферизацией кадров.

Особую значимость приобретает обработка видео в формате UYVY (YUV 4:2:2), широко применяемом в промышленных и встроенных видеосистемах, где требуется учитывать особенности представления яркостных и цветоразностных компонент. Разработка специализированной архитектуры фильтра с учётом структуры данного формата позволяет повысить эффективность использования аппаратных ресурсов и избежать избыточных вычислений. Полученные в работе решения могут быть использованы при создании интеллектуальных камер, бортовых видеосистем, медицинских и измерительных комплексов. Кроме того, результаты исследования способствуют развитию методов аппаратной оптимизации алгоритмов цифровой обработки изображений и расширяют практику применения FPGA в задачах высокоскоростной видеопереработки.

# **ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ ФИЛЬТРОВ НА ПЛИС**

Аппаратная реализация фильтров для обработки видеоизображений на FPGA — активно развивающаяся область, в которой на основе программируемой логики создаются специализированные модули для выполнения пространственных и временных преобразований видеопотока. FPGA (Field-Programmable Gate Array) предоставляет архитектуру с высокой степенью параллелизма, позволяющую проектировать потоковые алгоритмы свёртки и анализировать пиксели «на лету», что критично для задач реального времени при HD/Full-HD обработке.

## **1.1 Типы фильтров обработки видеоизображений на ПЛИС**

### **1) Линейные пространственные фильтры (2D FIR)**

Линейные пространственные фильтры, включая конечные импульсные отклики (2D FIR), широко применяются для сглаживания, повышения резкости и выделения границ. Эти фильтры реализуются через двумерные свёртки с ядрами фиксированных размеров (например,  $3\times 3$ ,  $5\times 5$ ), где каждый выходной пиксель является линейной комбинацией соседних входных пикселей. Оптимизация таких свёрток на ПЛИС включает развёртывание оконной логики и параллельные умножения с акцентом на минимизацию использования DSP-блоков.

### **2) Фильтры повышения резкости**

Фильтры повышения резкости усиливают высокочастотные компоненты изображения, делая границы объектов более чёткими. Одним из подходов является использование масок на основе лапласиана или высокочастотных усилителей, которые выделяют перепады яркости и подавляют медленно изменяющиеся области. В аппаратных решениях это достигается 2D-свёртками с соответствующими коэффициентами ядра. ([Eco-Vector Journals Portal][3]) Практические примеры коммерческих IP-ядров включают нерезкое маскирование (unsharp masking) с буферизацией строк и программируемым

коэффициентом усиления резкости, предназначенные для интеграции в видеотракты FPGA/SoC.

### 3) Фильтры выделения границ

Операторы типа Собеля и других градиентных фильтров реализуются на FPGA для выделения контуров на изображении. Они вычисляют приближённые производные по горизонтали и вертикали, что позволяет получить карту краёв. На FPGA такие алгоритмы оформляются как свёрточные блоки с малыми ядрами (обычно  $3 \times 3$ ), реализованными в конвейерной архитектуре для достижения высоких частот тактирования.

### 4) Медианные и морфологические фильтры

Нелинейные фильтры, такие как медианный, используются для подавления шума без значительного размытия границ. Медианный фильтр требует сортировки значений в окне, что усложняет аппаратную реализацию, но возможно через специализированные структуры сравнения и множественных регистров. Аналогичным образом реализуются морфологические преобразования (эрозия, дилатация), которые анализируют максимум/минимум в окне и применяются к бинарным или градационным изображениям для выделения структурных особенностей.

### 5) Комбинированные фильтры (шум и повышение резкости)

Существуют решения, комбинирующие шумоподавление и повышение резкости, где сначала применяется медианный или иной фильтр для удаления шумов, а затем 2D FIR-фильтр для усиления краёв. Такие конвейерные структуры особенно полезны при аппаратной обработке потоков с повышенными требованиями к визуальному качеству.

### 6) Библиотеки и генераторы IP-блоков

Современные инструменты разработки, такие как Vision HDL Toolbox и сопутствующие генераторы HDL, предоставляют готовые алгоритмы фильтрации и повышения резкости с возможностью автоматической генерации синтезируемого VHDL/Verilog кода для FPGA. Это значительно упрощает

разработку аппаратных модулей обработки видео, позволяя использовать отладенные библиотеки и адаптировать ядра под конкретные требования.

Таким образом, совокупность линейных и нелинейных фильтров, оптимизированных для потоковой обработки видеосигнала на FPGA, обеспечивает богатый инструментарий для повышения качества видеоизображения, в том числе повышения резкости, с возможностью адаптации под заданные требования по ресурсам и пропускной способности.

## 1.2 Применение ПЛИС в реальных устройствах

Программируемые логические интегральные схемы (FPGA) широко применяются в реальных устройствах видеообработки благодаря способности обеспечивать высокую пропускную способность при детерминированной задержке обработки. В отличие от универсальных процессоров, архитектура FPGA позволяет реализовывать глубоко конвейерные и параллельные структуры, обрабатывающие каждый пиксель видеопотока в такт системной частоте, что особенно важно для систем реального времени.

В промышленной практике FPGA используются в интеллектуальных камерах машинного зрения, где первичная обработка изображения (фильтрация, коррекция, выделение признаков) выполняется непосредственно на борту устройства. Подобные решения реализуются на базе ПЛИС компаний Xilinx и Intel, а также в составе однокристальных систем (SoC FPGA), совмещающих программируемую логику и процессорное ядро. Это позволяет распределять задачи между аппаратной (пиксельная обработка) и программной (управление, передача данных) частями системы.

В области видеонаблюдения FPGA применяются в регистраторах и IP-камерах для выполнения операций масштабирования, шумоподавления, повышения резкости, цветокоррекции и кодирования видеопотока. Аппаратная обработка снижает нагрузку на центральный процессор и позволяет обрабатывать видеопотоки высокой чёткости (HD, Full HD и выше) без потери кадров. Потоковая архитектура обеспечивает минимальную латентность —

обработка начинается сразу после поступления первых строк изображения, без ожидания полного кадра.

В телевизионной и профессиональной видеотехнике FPGA используются в устройствах захвата, видеомикшерах, системах постобработки и преобразования форматов. Высокая скорость работы достигается за счёт параллельной реализации арифметических операций, использования встроенных DSP-блоков и блоков памяти для организации буферов строк. Это делает возможной обработку видеосигналов с частотами десятки и сотни мегагерц при сохранении точности вычислений.

В фото- и видеокамерах специализированного назначения (медицинская визуализация, научная съёмка, тепловизионные системы) FPGA применяются для реализации алгоритмов улучшения контраста и резкости, компенсации искажений оптики и коррекции сенсорных шумов. Практичность таких решений обусловлена возможностью адаптации архитектуры под конкретный сенсор и интерфейс передачи данных, а также возможностью обновления конфигурации без изменения аппаратной платформы.

Отдельного внимания заслуживает энергоэффективность FPGA по сравнению с GPU при выполнении узкоспециализированных потоковых операций. При фиксированном алгоритме (например, двумерной свёртке) логическая реализация позволяет достичь высокой производительности на ватт, что критично для встраиваемых и автономных систем. Кроме того, детерминированный характер обработки упрощает сертификацию устройств, используемых в промышленности и транспорте.

Таким образом, применение FPGA в реальных устройствах видеообработки демонстрирует их практическую эффективность в задачах высокоскоростной фильтрации и улучшения качества изображения, включая аппаратную реализацию фильтров повышения резкости, где требуется сочетание высокой производительности, низкой задержки и рационального использования аппаратных ресурсов.

### **1.3 Преимущества использования ПЛИС для обработки видеопотока**

Аппаратная обработка видеопотока на FPGA обладает рядом ключевых преимуществ, которые делают этот подход оптимальным для реализации фильтров повышения резкости в реальном времени. Эти преимущества тесно связаны с архитектурными особенностями FPGA и специфическими требованиями к потоковой обработке видео.

#### **1. Высокая параллельность вычислений**

FPGA позволяет реализовать широкие конвейерные и параллельные структуры, обрабатывающие множество пикселей одновременно. В отличие от CPU и GPU, где параллельность ограничена количеством ядер и потоков, FPGA предоставляет возможность организовать обработку на уровне отдельных операций, включая умножение и сложение, что особенно важно при реализации 2D-свёрток для повышения резкости. Это обеспечивает обработку видеопотоков HD и Full HD с высокой пропускной способностью без потери кадров.

#### **2. Минимальная задержка обработки (детерминированность)**

Потоковая архитектура FPGA позволяет начинать обработку пикселя сразу после его поступления, без необходимости буферизации всего кадра. Это критично для приложений с жёсткими требованиями к времени отклика, таких как системы видеонаблюдения, автомобильные камеры и интеллектуальные сенсорные комплексы. Повышение резкости при потоковой обработке на FPGA выполняется практически без задержки, что невозможно достичь на стандартных CPU при тех же разрешениях и частотах кадров.

#### **3. Энергоэффективность и оптимальное использование ресурсов**

Специализированная логическая реализация свёрток и фильтров позволяет достигать высокой производительности на ватт. Встроенные блоки памяти и DSP-модули используются эффективно, минимизируя энергопотребление по сравнению с GPU, которые требуют значительных

ресурсов для выполнения одинаковых операций. Это особенно важно для встраиваемых и автономных систем с ограниченным энергопотреблением.

#### 4. Гибкость и настраиваемость под конкретные алгоритмы

FPGA позволяет оптимизировать фильтры повышения резкости под конкретные требования: размер окна свёртки, коэффициенты усиления, обработку граничных областей, работу с различными цветовыми форматами (например, UYVY, RGB). Такая настройка обеспечивает максимальное качество изображения при рациональном расходе аппаратных ресурсов.

#### 5. Поддержка высоких разрешений и частот кадров

FPGA справляется с обработкой видео 1080р и 4К при 60 и более кадрах в секунду благодаря конвейерной организации свёрток и прямому доступу к потокам данных. Реализация фильтров повышения резкости на таких скоростях позволяет получать визуально чёткое изображение в реальном времени, что особенно востребовано в промышленных и научных приложениях.

### **1.4 Фильтры обработки видео на видеокамерах**

Современные видеокамеры (как профессиональные, так и потребительского уровня) уже в момент съёмки выполняют широкий спектр фильтрации видеосигнала, обеспечивая улучшение качества изображения, коррекцию дефектов сенсора и подготовку видео к последующей обработке или воспроизведению. Такие фильтры реализуются в виде аппаратных блоков обработки изображения (ISP – Image Signal Processor), специализированных микросхем или программных модулей внутри камеры.

#### 1. Типы фильтров на видеокамерах

1) Пре-обработка сигнала (Raw-обработка) — преобразование «сырых» данных с сенсора камеры, включая коррекцию плотности цветовых фильтров, компенсацию дефектов пикселей и линейность яркости.

2) Баланс белого и цветокоррекция — алгоритмы автоматического определения и корректировки цветовой температуры, компенсации оттенков для получения натуральной цветопередачи.

3) Шумоподавление (Denoise) — пространственные и временные фильтры подавления шума, возникающего из-за высоких ISO, низкой освещённости или теплового шума сенсора. Часто используются медианные или ориентированные на видео аддитивные фильтры, сохраняющие границы при подавлении шума.

4) Повышение резкости (Sharpening / Edge Enhancement) — усиление локального контраста на границах объектов для визуального повышения чёткости изображения. Этот фильтр повышает акцент на перепадах яркости и тонких деталях, улучшая восприятие деталей кадра.

5) Коррекция оптики (Lens Correction) — фильтры, компенсирующие дисторсию линз, аберрации и виньетирование для более точной геометрии и цвета по всему полю кадра.

6) Фильтры цвета и тона (Look-Up Table, LUT) — предустановленные или программируемые таблицы коррекции цветов и тонального распределения, применяемые для художественного оформления или стилизации изображения (например, кинематографические профили).

7) Фильтры частоты кадров и интерполяции движения — сглаживание движения между кадрами, увеличение частоты обновления, использование временных фильтров для подавления дрожания и «трепета» объектов.

## 2. Реальные примеры устройств

Цифровые зеркальные камеры и беззеркальники: камеры от Nikon, Canon, Sony содержат выделенные ISP-процессоры (например, Nikon Expeed, Canon DIGIC), которые реализуют всю цепочку обработки: шумоподавление, цветокоррекцию, повышение резкости и компенсацию оптических дефектов.

- Профессиональные видеокамеры ARRI, RED, Blackmagic Design оснащены мощными аппаратными модулями обработки изображения,

способными выполнять многокадровое шумоподавление и динамическую коррекцию тёмных/светлых областей, сохраняя широкие динамические диапазоны.

- Потребительские и смартфоны: матричные процессоры ISP в смартфонах выполняют сложные фильтры HDR, подавление шума, алгоритмическую резкость, компенсацию дрожания и реализацию LUT-профилей для видео прямо при записи.
- IP-камеры видеонаблюдения: встроенные фильтры повышения резкости, подавления шума и компенсации задней засветки обеспечивают более четкое распознавание объектов и лиц в условиях низкой освещённости.

### 3. Постпродакшн-фильтры

Постпродакшн-фильтры применяются уже после съёмки видеопотока на этапе монтажа или цветокоррекции. Они выполняются в виде программных модулей в видеоредакторах и могут содержать:

- Расширенное шумоподавление — более сложные методы, учитывающие движение между кадрами, спектральные признаки шума и другие контексты.
- Тональная и цветовая градация (Color Grading) — тонкие настройки цветов, контраста и освещённости для художественной стилизации кадра.
- Деблокирование и интерполяция — удаление артефактов сжатия, восстановление деталей, сглаживание «ступенчатых» переходов (фильтры типа Deblocking используются, например, при восстановлении видео после кодеков сжатия).
- Специальные эффекты — размытость по Гауссу, имитация глубины резкости, эффекты свечения, кинематографические стили и другие художественные преобразования.

Эти фильтры зачастую выполняют задачу доведения визуального качества до профессионального уровня, что невозможно достичь только средствами встроенной обработки камеры. Постпродакшн-фильтры работают уже с полным видеопотоком кадр за кадром, используя более мощные вычислительные ресурсы, чем доступно «на борту» устройства.

Один из хороших примеров визуализации эффекта повышения резкости — рисунок 1, демонстрирующий применение unsharp masking до и после.



Рисунок 1 – До и после применения фильтра unsharp masking

## **ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ О ПОВЫШЕНИИ РЕЗКОСТИ ВИДЕОИЗОБРАЖЕНИЯ**

### **2.1 Формат видео UYVY**

#### **2.1.1 Краткая история появления видеоформата**

Формат UYVY, относящийся к семейству YUV (или YCbCr), возник в контексте цифрового видео как способ представить цветное изображение, отделив яркостную компоненту (luma, Y) от цветоразностных компонент (chrominance, U и V). Изначально модели YUV были разработаны для аналоговых телевизионных стандартов PAL/NTSC с целью передавать цветное изображение в совместимости с чёрно-белым ТВ, где яркость (Y) играла ключевую роль в формировании воспринимаемого сигнала, а цветовые компоненты (U, V) представляли дополнительные сведения о цвете.

С распространением цифрового видеообразования и стандартов сжатия (MPEG-2, JPEG, DV), были разработаны форматы субдискретизации цветовой информации, такие как 4:4:4, 4:2:2 и 4:2:0, позволяющие уменьшать объём данных, хранящихся и передаваемых при минимальных потерях визуального качества. Формат UYVY — это один из стандартов представления видео в цветовой модели YUV 4:2:2, где каждый пиксель характеризуется полными данными яркости и совместно используемыми компонентами цветности для соседних пикселей. Он стал широко применяться в интерфейсах передачи видеоданных (например, SDI, USB видеозахват, камеры промышленного класса), поскольку позволяет экономить пропускную способность, сохраняя достаточное цветовое качество для большинства приложений. Пример изображения цветового пространства YUV можно увидеть рисунке 2.

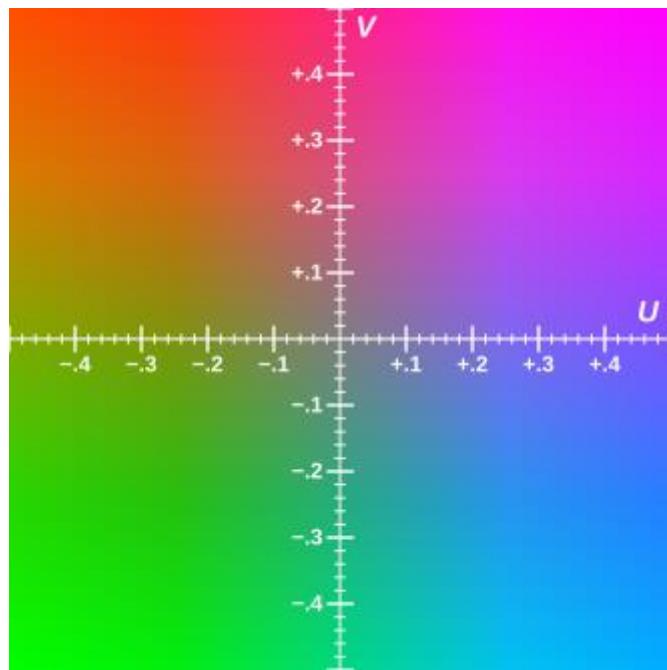


Рисунок 2 – Визуализация компонентов цветности при фиксированном значении яркости

### 2.1.2 Структура формата UYVY

Формат UYVY представляет данные видеопотока как последовательность байт, где каждая группа из четырёх байт кодирует два пикселя. Это отражает стратегию субдискретизации 4:2:2 — яркостная компонента (Y) присутствует для каждого пикселя, а цветоразностные компоненты U и V чередуются между ними (таблица 1).

Таблица 1 – Структура видеоформата UYVY в памяти

Byte №	Компонента	Обозначение
0	U	Chrominance (Cb) — для двух пикселей
1	$Y_0$	Luma первого пикселя
2	V	Chrominance (Cr) — для двух пикселей
3	$Y_1$	Luma второго пикселя

Подробное объяснение:

- U (Cb) — цветоразностная компонента синего: задаёт отличие от яркостной компоненты, влияет на оттенок цвета в направлении синего.

- $V$  ( $Cr$ ) — цветоразностная компонента красного: аналогично отвечает за оттенок в направлении красного спектра.
- $Y_0$  и  $Y_1$  — яркостные компоненты двух последовательных пикселей, определяющие воспринимаемую светимость и детали изображения.

Таким образом, каждая пара пикселей совместно использует один набор цветоразностных данных ( $U$  и  $V$ ), но имеет отдельные значения яркости. Это отражает компромисс между визуальным качеством и экономией памяти / пропускной способности.

Преимущество такого представления состоит в том, что большинство визуальной информации воспринимается через яркость, тогда как человеческий глаз менее чувствителен к мелким вариациям цветности. Формат 4:2:2 уменьшает объём цветовой информации без заметной потери качества изображения.

### 2.1.3 Другие форматы дискретизации

Помимо UYVY (YUV 4:2:2), широко применяются следующие форматы, указанные в таблице 2.

Таблица 2 – Разные форматы дискретизации изображения

<b>Формат</b>	<b>Подавление цветности</b>	<b>Описание</b>
<b>4:4:4</b>	Нет	Полные данные яркости и цветности для каждого пикселя — максимум качества и объёма данных
<b>4:2:2</b>	Субдискретизация по горизонтали	Как UYVY — два пикселя на один набор ( $U$ , $V$ )
<b>4:2:0</b>	Субдискретизация по горизонтали и вертикали	Ещё большая экономия данных (широко используется в MPEG/HEVC)

*Последние цифры отражают количество сэмплов яркости и цветности на группу из 4 пикселей: 4:4:4 — все компоненты, 4:2:2 — снижение цветовой плотности по горизонтали, 4:2:0 — по горизонтали и вертикали.*

Наглядную визуализацию форматов дискретизации можно увидеть на рисунке 3.

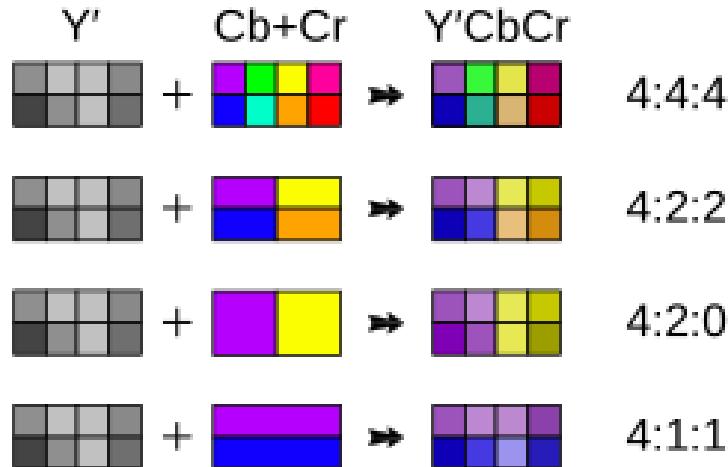


Рисунок 3 – Визуальное представление разных форматов YUV изображения

#### 2.1.4 Обоснование выбора формата UYVY для FPGA

##### 1) Использование на реальных устройствах

Формат UYVY долгое время стал де-факто стандартом в промышленной видеоаппаратуре:

- широко применяется в интерфейсах видеозахвата (USB, SDI, PCIe),
- поддерживается камерами промышленного класса и устройствами захвата (frame grabbers),
- используется в потоковых цифровых интерфейсах камер (*Camera Link, GigE Vision, USB3 Vision*).

Такое распространение означает, что обработка UYVY может осуществляться «в потоке» без предварительных преобразований форматов, что уменьшает сложность FPGA-архитектуры.

##### 2) Ограниченностъ ресурсов ПЛИС

FPGA обладает ограниченными ресурсами: логическими элементами, DSP-модулями, памятью. Цветовое представление **RGB** требует хранения трёх компонент (R, G, B) для каждого пикселя, что влечёт за собой:

- больший объём буферов для хранения строк и окон свёртки,

- увеличение числа арифметических операций (сумм, умножений) для трёх каналов,
- более высокую нагрузку на шины данных.

В UYVY же данные передаются как:

- отдельные компоненты яркости Y для каждого пикселя,
- одна пара цветоразностных компонентов на пару пикселей.

Это позволяет сократить объём обрабатываемых данных на уровне цветности в 2 раза по сравнению с RGB без ощутимой потери визуального качества, что критично для FPGA-реализаций, где количество логических блоков и памяти ограничено.

### 3) Отсутствие преобразований цветового пространства

При реализации фильтров повышения резкости в подавляющем большинстве случаев достаточно работать с яркостной компонентой (Y), поскольку:

- Y описывает интенсивность света, которая главным образом определяет визуальную чёткость и детали;
- цветоразностные компоненты U и V не несут пространственной информации, более чувствительной к резкости;
- фильтрация цветности может привести к нежелательным цветовым артефактам.

Для повышения резкости:

- операция свёртки и расчёт градиентов/масок применяется только к Y;
- U и V часто передаются «как есть», либо подвергаются минимальной обработке (или смешиваются с Y после фильтра).

Это означает, что FPGA не нужно выполнять преобразование из UYVY → RGB → Y и обратно, что экономит как ресурсы памяти и DSP, так и такты обработки.

## 2.2 Сравнение и выбор лучшего для ПЛИС метода

Таблица 1 - Сравнение методов повышения резкости видеоизображения для ПЛИС

Название метода	Сложность вычислений	Требования к памяти	Применимость к UYVY	Реализация на ПЛИС
<b>Unsharp Masking (нерезкое маскирование)</b>	Средняя: свёртки (размытие и вычитание) 1–2	Средняя: требуется буфер строк для свёртки	Высокая: работает только с Y, U/V можно оставить без обработки	Конвейерная реализация, DSP для умножения, буфер строк для окна свёртки
<b>Лапласиан (Laplacian Filter)</b>	Средняя: одна свёртка с ядром $3 \times 3$	Низкая—средняя: буфер строк 2–3	Высокая: легко применимо к Y	Простая реализация через суммирование и вычитание, минимальное использование DSP
<b>Высокочастотная маска (High-pass Filter)</b>	Средняя—высокая: свёртка масштабированное	Средняя: буфер строк для окна 1 +	Высокая: работает на Y	Конвейерная реализация с умножением и сложением, можно ограничиться Y

Название метода	Сложность вычислений	Требования к памяти	Применимость к UYVY	Реализация на ПЛИС
<b>Метод на основе градиента (Sobel / Prewitt)</b>	Средняя: свёртки (горизонтальная и вертикальная)	2 Средняя: буфер строк 2–3	Высокая: Y-компоненты	Реализуется конвейерно, требует сложения и сравнения, относительно DSP-независимый

С учётом особенностей FPGA и формата UYVY:

1. Фильтрация должна обрабатываться *только яркостную компоненту Y*, чтобы экономить память и вычислительные ресурсы.
2. Аппаратная реализация должна быть конвейерной, с минимальной задержкой и рациональным использованием DSP и логических элементов.
3. Метод *Unsharp Masking* оптимально сочетает простоту реализации, высокое визуальное качество и возможность работы «в потоке» на FPGA. Он требует средней вычислительной мощности и памяти (буфер строк для свёртки), полностью применим к UYVY и позволяет гибко регулировать коэффициент усиления резкости.

Таким образом, для потоковой аппаратной обработки видеопотока в формате UYVY на FPGA наиболее эффективным является метод *Unsharp Masking*, обеспечивающий визуально заметное улучшение чёткости при рациональном расходе ресурсов ПЛИС.

## **ЗАКЛЮЧЕНИЕ**

Таким образом, использование языковой модели (в частности ChatGPT) позволило ускоренным образом сгенерировать и составить вводную и теоретическую главы исследовательской работы по заданной теме «аппаратная реализация фильтра для повышения резкости видеоизображения». С незначительными поправками, исправлению неточностей и удалению маркировок форматирования, получилось написать приемлемую и содержательную текстовую часть научной работы. Также нейросеть смогла предоставить ссылки на картинки, подкрепляющие текст работы. Также получилось выявить пути улучшения текста исходной бакалаврской ВКР. При исследовании необходимо было явно указывать требуемый объем текста в предложениях и убирать избыточные пункты и тавтологические утверждения. Несмотря, на хорошую генерацию введения и актуальности, требуется внимательно следить за структурой и содержанием сгенерированных ответов. Важно приблизиться к максимально возможной формализации и подробному описанию промпта для достижения желаемого результата в генерации статьи. Помимо введения и актуальности, ChatGPT также хорошо справился с генерацией аннотации и её переводом.