

Оптимизация алгоритмов фильтрации и поиска пика видеосигнала на программируемых логических интегральных схемах (ПЛИС) для задач лазерной резки

А.А. Молотков^{1,А}, Ю.А. Сальков^{2,В}, О.Н. Третьякова^{3,А}, Д.Н. Тужилин^{4,В}

^А ФГБОУ ВО Московский авиационный институт (национальный исследовательский университет) МАИ, Москва, 125993, Россия

^В ООО «Лаборатория промышленных исследований» группы компаний «Лазеры и аппаратура ТМ», Москва, 124498, Россия

¹ ORCID: 0000-0002-9335-5219, karacerr@gmail.com

² ORCID: 0009-0002-4928-3171, salkov@laser-app.ru

³ ORCID: 0000-0003-0256-4558, tretiyakova_olga@mail.ru

⁴ ORCID: 0000-0002-8570-1732, tuzhilin@laserapr.ru

Аннотация

В процессе лазерной резки необходимо точно отслеживать расстояние между поверхностью обрабатываемой заготовки и линзой, фокусирующей лазерный луч. Для отслеживания расстояния была разработана система оптического слежения за зазором (СОС). Для работы данной оптической схемы необходимо с большой частотой определять пик, создаваемый лазерным лучом на кадре с камеры.

Для высокоскоростной обработки видео была взята система на кристалле (SoC) Zynq 7000 от компании Xilinx. Для определения положения пика на кадре были разработаны алгоритмы фильтрации и поиска центра масс пика. Эти алгоритмы были адаптированы для работы на ПЛИС.

Статья является продолжением работ по реализации и внедрению алгоритмов машинного зрения на ПЛИС. В данной статье представлены результаты реализации алгоритмов, а также результаты работы алгоритмов при разных параметрах.

Ключевые слова: ZYNQ, машинное зрение, обработка видео, ПЛИС, фильтрация картинки, moving average, скользящее среднее, HLS.

Введение

Настоящая статья является продолжением серии работ [1–2], посвящённых разработке и реализации алгоритмов машинного зрения на программируемых логических интегральных схемах (ПЛИС). В предыдущих публикациях основное внимание уделялось общей архитектуре систем обработки видеосигналов и принципам построения аппаратных решений. В данной работе акцент сделан на углублённом математическом анализе используемых фильтров и алгоритмов, оценке их эффективности с точки зрения потребляемых аппаратных ресурсов, а также на количественном анализе качества фильтрации.

Особое внимание уделяется реализации модифицированного фильтра скользящего среднего и пороговой фильтрации, предназначенных для обработки видеопотока в системе оптического слежения за зазором (СОС) при лазерной резке. Алгоритмы реализованы с использованием средств высокоуровневого синтеза (Vitis HLS) и адаптированы для работы на SoC-платформе Zynq 7000 от Xilinx.

Для оценки качества фильтрации в статье рассматривается метрика PSNR, вычисленная при различных размерах скользящего окна. PSNR (пиковое отношение сиг-

нал/шум) широко применяется в задачах цифровой обработки изображений и видеосигналов благодаря своей простоте, интерпретируемости и возможности сравнения с эталонным сигналом. В отличие от более сложных метрик, таких как SSIM (структурное сходство), VMAF (Video Multi-Method Assessment Fusion) или VIF (Visual Information Fidelity), PSNR позволяет быстро оценить эффективность подавления шума. Выбор данной метрики обусловлен тем, что цель фильтрации — не визуальная точная передача картинки, а уверенное определение ярко выраженного пика сигнала.

Ряд работ по видеообработке на ПЛИС [3-5, 12-16] демонстрируют эффективные конвейеры для задач реального времени: от минималистичных трактов «камера-ПЛИС-дисплей» с бюджетными CMOS-интерфейсами до реализаций классических фильтров (например, Собеля) и сглаживающих фильтров, оптимизированных под ресурсы кристалла (включая аппроксимации коэффициентов степенями двойки для снижения использования DSP-блоков). Эти подходы подтверждают уместность ПЛИС как платформы для высокочастотной видеоаналитики, но в основном ориентированы либо на общие контуры визуализации/детектирования границ, либо на архитектурные оптимизации фильтрации без жесткой привязки к задаче быстрого и устойчивого поиска пика в узкополосном изображении технологической зоны лазерной обработки.

Связанные работы

Статьи [1-2] описывают особенности реализации рассматриваемых алгоритмов машинного зрения на ПЛИС. Данная статья является продолжением работ по разработке и внедрению алгоритмов машинного зрения на ПЛИС.

В документе [3] представлена аппаратно-эффективная система обработки видео в реальном времени с использованием ПЛИС с интерфейсом камеры OV7670 и выходом VGA. Система разработана на ПЛИС Spartan-6 и Artix-7, для реализации используются Xilinx Vivado и Verilog HDL. Работа направлена на оптимизацию использования аппаратных ресурсов при одновременном обеспечении возможностей обработки видео в реальном времени. В этой статье описывается подключение и обработка видеопотока с камеры.

В исследовании [4] разрабатывается система обработки видео в реальном времени на ПЛИС с использованием алгоритма обнаружения границ при помощи Собель-фильтра. Система объединяет CMOS-камеру, процессор Microblaze и дисплей DVI. Реализованный на ПЛИС Spartan-3A DSP, этот проект использует инструменты System Generator и Embedded Development Kit (EDK) для оптимизации производительности обработки изображений во встроенных средах. В статье описывается видеотрансляция для определения границ. Принципы этой видеотрансляции мы можем использовать в нашей работе.

В исследовании [5] основное внимание уделяется разработке гауссовского сглаживающего фильтра, оптимизированного для реализации ПЛИС, с использованием коэффициентов степени двойки. В исследовании предлагается алгоритм аппроксимации коэффициентов фильтра Гаусса, снижающий сложность аппаратного обеспечения при сохранении эффективности подавления шума. Реализованная на ПЛИС Xilinx Artix-7, система обеспечивает эффективную фильтрацию в реальном времени, улучшая соотношение сигнал/шум при минимальных вычислительных затратах.

Эти работы дают ценную информацию об обработке видео в реальном времени на базе ПЛИС, охватывая такие аспекты, как разработка гауссовского сглаживания, эффективное использование оборудования и оптимизированные методы фильтрации, но при разработке алгоритма фильтрации для СОС необходимо было учитывать аппаратные особенности и требования к скорости работы фильтра, поэтому многие подходы к реализации фильтров, описанные в вышеуказанных статьях, неприменимы к нашей разработке.

Аппаратная конфигурация и инструменты разработки

Для реализации СОС была использована отладочная плата AMD Zynq 7000SoC ZC702 от компании Xilinx. Данная плата является системой на кристалле (SoC).

При разработке контроллера СОС использовался инструмент высокоуровневого синтеза Vitis-HLS. Инструменты высокоуровневого синтеза позволяют разрабатывать отдельные функциональные блоки путем написания кода на языках высокого уровня (C++, Python и т.д.), а не на языках описания аппаратуры (Verilog, HDL и т.д.). Эти инструменты позволяют разработчику функциональных блоков абстрагироваться от уровня регистровых передач (RTL) и разрабатывать код на уровне обработки данных.

Алгоритм работы контроллера

Входные данные контроллера – монохромный видеопоток в формате CameraLink, каждый пиксель кодируется 8 битами.

Выходные данные после обработки видео – положение и ширина пика на кадре в пикселях.

Алгоритм работы контроллера представляет собой:

1. Получение потокового видео и конвертация его из формата CameraLink в формат AXI-stream
2. Предварительная обработка видеопотока с помощью модифицированного фильтра скользящего среднего:

$$y_{ij} = \frac{1}{2N} \sum_{m=0}^1 \sum_{n=-N}^0 x_{i+m, j+n}$$

3. Фильтрация видеопотока с помощью порогового фильтра:

$$y_i = \begin{cases} 1, & \text{if } x_i \geq T \\ 0, & \text{if } x_i < T \end{cases}$$

4. Поиск пика на кадре при помощи формулы определения масс-центра:

$$y_{ij} = \frac{\sum_{i=k}^{k+n} (r_i * m_i)}{\sum_{i=k}^{k+n} (m_i)}$$

После определения масс-центра пика на кадре контроллер генерирует управляющие сигналы для серводрайвера и некоторых других модулей. Далее в статье будут описаны особенности реализации алгоритмов фильтрации и поиска пика на ПЛИС.

Разработка алгоритма фильтрации

Как упоминалось выше, контроллер СОС должен быстро принимать и обрабатывать входящий видеопоток. Перед поиском пика на кадре, необходимо отфильтровать входящий видеопоток. Для фильтрации кадра был использован модифицированный алгоритм скользящего среднего в сочетании с пороговой фильтрацией. Эти алгоритмы были выбраны в связи с тем, что пик на кадре с камеры является явно выраженным и необходимо лишь убрать сильные шумы, чтобы правильно определять масс-центр пика.

Разностное уравнение для модифицированного фильтра скользящее среднее можно записать следующим образом:

$$y_{ij} = \frac{1}{2N} \sum_{m=0}^1 \sum_{n=-N}^0 x_{i+m, j+n}, \quad (1)$$

где N – размер окна.

Разностное уравнение порогового фильтра имеет вид:

$$y_i = \begin{cases} 1, & \text{если } x_i \geq T \\ 0, & \text{если } x_i < T' \end{cases} \quad (2)$$

где T – пороговое значение.

Математическая модель алгоритмов очень проста, но при реализации этих алгоритмов на ПЛИС необходимо учитывать некоторые особенности и стараться максимально распараллеливать процессы для достижения наилучшей производительности.

Сложность алгоритмов фильтрации и количество затраченных ресурсов

При реализации алгоритмов с использованием инструмента Vitis HLS существует несколько параметров, которые в совокупности определяют временную сложность реализуемого алгоритма. Такие параметры, как *Latency*, *iteration latency*, *interval*, составляют общую временную сложность алгоритма. Все параметры измеряются в тактах, так как один и тот же реализованный алгоритм может работать на разных частотах. Использование ресурсов ПЛИС рассчитывается во время синтеза алгоритма. Каждая ПЛИС обладает ограниченным количеством ресурсов, которые могут быть использованы для реализации любых алгоритмов. В таблице 1 приведены все значимые временные параметры сложности и оценка использования ресурсов ПЛИС после синтеза алгоритма.

Таблица 1. Оценка эффективности и ресурсов

Modules and loops	Latency (cycles)	Iteration latency	Interval	Pipe-lined	BRAM	DSP	FF	LUT
VideoProcessingSystem	4175	-	4176	no	2 (~0%)	6 (2%)	5846 (5%)	7399 (13%)
Pipeline	4143	-	4143	no	-	6 (2%)	4747 (4%)	4532 (8%)
Loop	4141	47	1	yes	-	-	-	-

Параметр *Latency* показывает общую временную сложность алгоритма в тактах, т.е. параметр показывает, сколько тактов потребуется для обработки входных данных. Параметры *iteration latency* и *interval* показывают задержку в тактах между итерациями алгоритма, это необходимо в некоторых случаях, поскольку при реализации алгоритмов Vitis HLS последовательно разделяет весь написанный код на некоторое число состояний конечного автомата. В каждом состоянии выполняются конкретные операции с данными. Таким образом, мы можем упрощенно сказать, что Vitis HLS использует конечный автомат для перевода высокоуровневого кода в RTL-представление.

Основным параметром ресурсов ПЛИС является количество *LUT* (LookUp Table), которая указывает, насколько масштабный проект может быть реализован на конкретной модели ПЛИС. Параметр *FF* указывает, сколько триггеров будет использоваться. Параметр *DSP* указывает, сколько блоков DSP используется на ПЛИС, эти блоки представляют собой аппаратно реализованные функции, такие как умножение или деление двух чисел с плавающей запятой и т.д.. Эти блоки позволяют уменьшить количество использованных ресурсов ПЛИС, за счет готовой реализации некоторых распространенных функций.

Исходя из данных, представленных в таблице, можно сказать, что реализация алгоритма прошла успешно. Ресурсы ПЛИС расходуются в небольшом количестве, а временная сложность алгоритма приемлемая.

При реализации алгоритма инструмент Vitis HLS детализирует временную сложность, в отчетах после синтеза можно увидеть общую временную сложность алгоритма. При реализации алгоритмов фильтрации была достигнута задержка в 4175 тактов. При запуске алгоритма на частоте 100 МГц временная задержка составит ~42 мкс. Следует отметить, что временная сложность алгоритма не зависит от размера окна, это связано с особенностью реализации алгоритма, описанной выше.

Результаты работы алгоритмов фильтрации

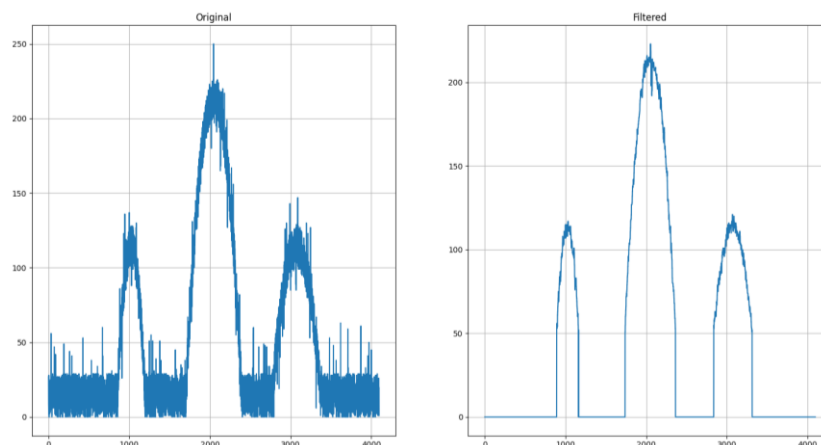


Рисунок 5. Результат работы алгоритмов фильтрации, размер окна 6

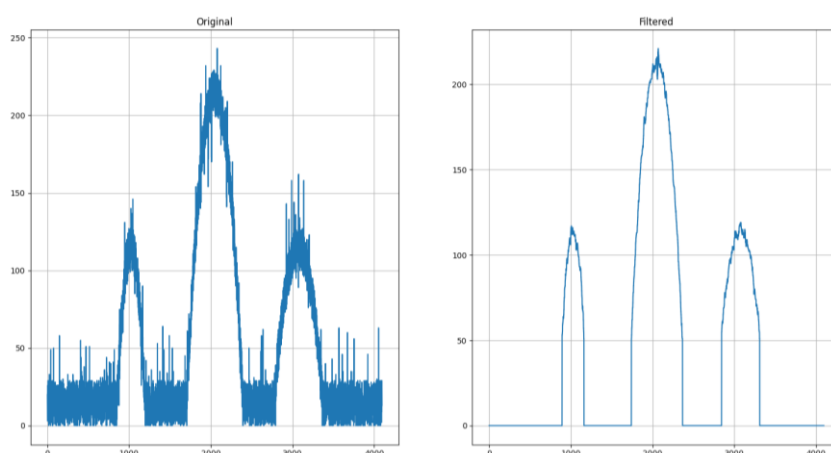


Рисунок 6. Результат работы алгоритмов фильтрации, размер окна 10

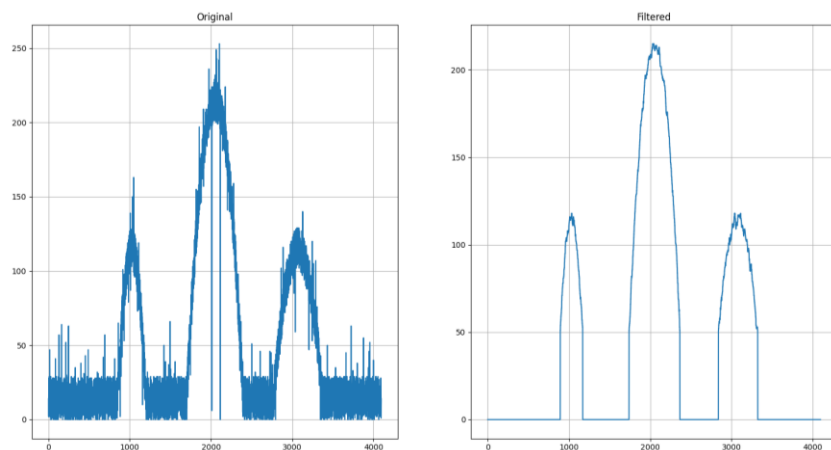


Рисунок 7: Результат работы алгоритмов фильтрации, размер окна 14

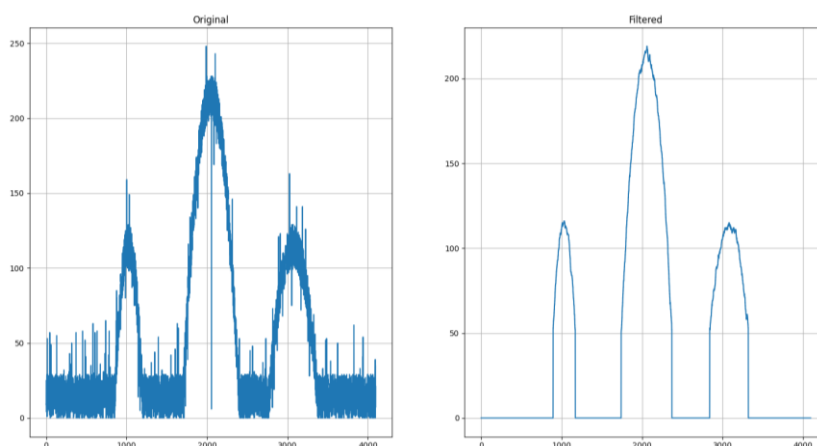


Рисунок 8. Результат алгоритмов фильтрации, размер окна 18

При генерации входных данных была использована стандартная библиотека C++ "random" для генерации шума и выбросов на входном изображении. При генерации пика, в дополнение к основному пику, также создаются два пика вдвое меньшей амплитуды и ширины.

Пиковое отношение сигнал/шум (PSNR) является широко используемым показателем для измерения качества восстановленного или отфильтрованного изображения/сигнала по сравнению с оригиналом. Оно выражается в децибелах (дБ), а более высокие значения PSNR указывают на лучшее качество (меньше искажений или шума).

$$PSNR = \frac{\max(X_f)}{MSE}$$

Где X_f - после фильтрации сигнала.

Результаты PSNR для разных размеров окна показаны на рисунке 5.

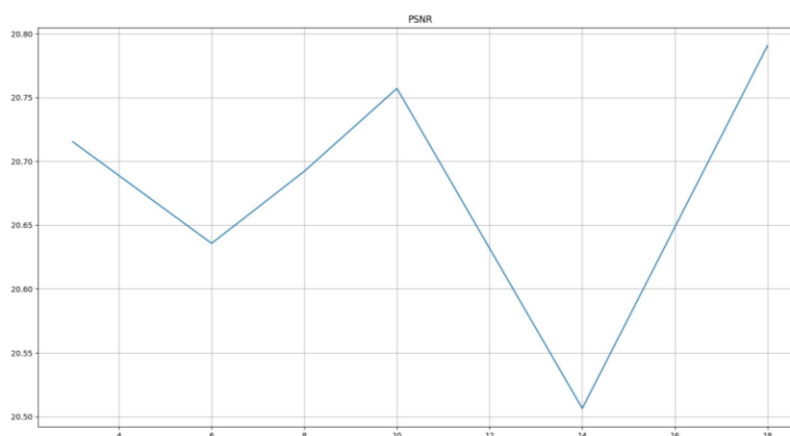


Рисунок 5. Результаты PSNR для разных размеров окон

Таблица 2. Интерпретация качества PSNR

PSNR (дБ)	Оценка качества
> 40 дБ	Превосходно (почти без потерь)
30 - 40 дБ	Хорошо (незначительный шум, незаметные искажения)
20 - 30 дБ	Приемлемо (видимый шум)
10 - 20 дБ	Плохое (значительное искажение, заметная деградация)
< 10 дБ	Очень плохо (сильный шум, непригодный для использования сигнал)

PSNR = 20 дБ находится на нижней границе "приемлемого" качества. Это означает, что в обработанном сигнале присутствует видимый шум. Для высококачественной об-

работки сигнала обычно предпочтительнее значение 30+ дБ. В шумных средах (например, при медицинской визуализации, видео при слабом освещении, обработке данных с датчиков) все еще может использоваться значение 20 дБ. Для целей этого проекта PSNR = 20 дБ является нормальным, потому что пик на изображении хорошо виден.

Алгоритмы фильтрации успешно справились с задачей фильтрации зашумленного изображения с небольшим количеством выбросов. Согласно графикам, мы можем видеть, что оптимальная ширина окна для выполнения задачи составляет 14 пикселей.

Текущие ограничения и будущие изменения

На данный момент алгоритмы контроллера рассчитаны только на разрешение кадра 4096*2. Этот факт накладывает ограничения на используемые камеры. В будущем необходимо доработать алгоритм, чтобы можно было использовать его в сочетании с любой камерой.

Реализация алгоритмов с помощью инструмента Vitis HLS позволяет довольно легко модифицировать работу алгоритма, что позволяет использовать в будущем алгоритмы медианного фильтра или любые Гауссовы фильтры.

Заключение

В результате работы были получены значения затраченных ресурсов ПЛИС на реализацию фильтров, а также реализованные фильтры были протестированы на сгенерированных данных, чтобы получить информацию о качестве фильтрации.

Количество затраченных ресурсов (10 тыс. LUT) является приемлемым значением для такой реализации алгоритмов так как такая реализация позволяет быстро вносить изменения в алгоритм работы фильтров.

Метрика PSNR фильтра, скользящего среднего (~20 дБ) находится в приемлемом диапазоне качества фильтрации. Так как пик на картинке с камеры ярко выраженный, такое качество фильтрации является приемлемым.

Благодарности

Авторы выражают благодарность руководству группы компаний "Лазеры и Апертура ТМ" за помощь в материально-техническом обеспечении экспериментальных исследований и моделировании рассматриваемого процесса.

Список литературы

1. Salkov Yu.A., Tretiyakova O.N., Tuzhilin D.N. Development and application of algorithms of video stream filtering and processing on programmable logic integrated circuits FPGA // Scientific Visualization. 2024. T. 16. № 4. С. 102-108.
2. Разработка программного обеспечения контроллера системы соосного слежения за профилем образца при лазерной обработке в режиме реального времени Сальков Ю.А., Сапрыкин Д.Л., Третьякова О.Н., Тужилин Д.Н. Приборы. 2024. № 5 (287). С. 8-14.
3. Navaneethan, S., Varshini, M. A., Anil, A., & Thanusha, C. (2022). A Hardware Efficient Real-time Video Processing on ПЛИС with OV 7670 Camera Interface and VGA. Proceedings of the Sixth International Conference on Electronics, Communication and Aerospace Technology (ICECA 2022). IEEE.
4. Said, Y., Saidani, T., Smach, F., Atri, M., & Snoussi, H. (2012). Embedded Real-Time Video Processing System on ПЛИС. In A. Elmoataz et al. (Eds.), Lecture Notes in Computer Science (LNCS), vol. 7340, pp. 85–92, Springer-Verlag

5. Ivashko, A., Zuev, A., Karaman, D., & Moškon, M. (2024). FPGA-Based Implementation of a Gaussian Smoothing Filter with Powers-of-Two Coefficients. *Advanced Information Systems*, Vol. 8, No. 2, pp. 39–46. DOI: 10.20998/2522-9052.2024.2.05
6. Molotkov A.A., Tretiyakova O.N. On possible approaches to visualizing the process of selective laser melting // *Scientific Visualization*, 2019, volume 11, number 4, pages 1 - 12, DOI: 10.26583/sv.11.4.01
7. Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения // *Физматкнига*, 2010. р. 672 с.
8. Кондратенко В.С., Третьякова О.Н., Шевченко Г.Ю. Разработка средств управления лазерным технологическим оборудованием с различными кинематическими схемами // *Журнал «Вестник Московского авиационного института»*, 2015, т.22, №2, С.121-131.
9. Молотков А.А. Сапрыкин Д.Л., Третьякова О.Н., Тужилин Д.Н. Программное обеспечение регистрации потоковых видеоданных FlexRecorder Свидетельство о государственной регистрации программы для ЭВМ №2021669002. Заявка №.2021668322. Дата поступления 16 ноября 2021г. Дата государственной регистрации в Реестре программ для ЭВМ 23 ноября 2021.
10. Третьякова О.Н., Молотков А.А. Семашко В.С. Программа FastDetector машинного зрения для создания детали в процессе селективного лазерного сплавления. Свидетельство о государственной регистрации программы для ЭВМ №2021669857. Заявка №.2021669184. Дата поступления 26 ноября 2021г Дата государственной регистрации в Реестре программ для ЭВМ 03 декабря 2021.
11. Третьякова О.Н., Молотков А.А. Желябин И.А. Программа HarrisDetector машинного зрения для создания детали в процессе селективного лазерного сплавления Свидетельство о государственной регистрации программы для ЭВМ №2021680165. Заявка №.2021669247. Дата поступления xx декабря 2021г. Дата государственной регистрации в Реестре программ для ЭВМ 03 декабря 2021.
12. Jia, Chuanmin, et al. "Fpx-nic: An fpga-accelerated 4k ultra-high-definition neural video coding system." *IEEE Transactions on Circuits and Systems for Video Technology* 32.9 (2022): 6385-6399.
13. Cong, Jason, et al. "FPGA HLS today: successes, challenges, and opportunities." *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 15.4 (2022): 1-42.
14. Bailey, Donald G. *Design for embedded image processing on FPGAs*. John Wiley & Sons, 2023.
15. Cui, Beiyao, et al. "Design of Image Acquisition Circuit for Pixel Coupled Infrared Polarization Camera Based on FPGA." 2023 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC). IEEE, 2023.
16. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. T. 1. — IEEE. 2001. — С. I—511.

Optimization of Filtering Algorithms and Video Signal Peak Search on Programmable Logic Integrated Circuits (FPGA) for Laser Cutting Tasks

A.A. Molotkov^{1,A}, Yu.A. Salkov^{2,B}, O.N. Tretiyakova^{3,A}, D.N. Tuzhilin^{4,B}

^A Moscow Aviation Institute (National Research University) MAI, Moscow, 125993, Russia

^B LLC “PROMIS LAB” of the group of companies “Lasers and Equipment TM”,
Moscow, Russia

¹ ORCID: 0000-0002-9335-5219, karacerr@gmail.com

² ORCID: 0009-0002-4928-3171, salkov@laser-app.ru

³ ORCID: 0000-0003-0256-4558, tretiyakova_olga@mail.ru

⁴ ORCID: 0000-0002-8570-1732, tuzhilin@laserapr.ru

Abstract

In the process of laser cutting, it is crucial to accurately monitor the distance between the surface of the workpiece and the focusing lens of the laser beam. To achieve this, an optical gap-tracking system (OGTS) was developed. The operation of this optical setup requires high-frequency detection of the peak generated by the laser beam on camera frames.

For high-speed video processing, the Zynq-7000 system-on-chip (SoC) from Xilinx was employed. To determine the peak position within a frame, filtering algorithms and a center-of-mass peak detection method were developed. These algorithms were adapted for FPGA implementation.

This article extends previous research on the design and deployment of computer vision algorithms on FPGA. The paper presents implementation results as well as the performance of the algorithms under various parameter settings.

Keywords: Zynq, computer vision, video processing, FPGA, image filtering, moving average, HLS.

References

1. Salkov Yu.A., Tretiyakova O.N., Tuzhilin D.N. Development and application of algorithms of video stream filtering and processing on programmable logic integrated circuits FPGA // Scientific Visualization. 2024. Vol. 16, No. 4, pp. 102–108.
2. Сальков Ю.А., Сапрыкин Д.Л., Третьякова О.Н., Тужилин Д.Н. Разработка программного обеспечения контроллера системы соосного слежения за профилем образца при лазерной обработке в режиме реального времени [Software development of a controller for a coaxial sample profile tracking system during real-time laser processing] // Приборы [Instruments]. 2024. No. 5 (287), pp. 8–14.
3. Navaneethan, S., Varshini, M. A., Anil, A., & Thanusha, C. (2022). A Hardware Efficient Real-time Video Processing on ПЛИС [FPGA] with OV 7670 Camera Interface and VGA. Proceedings of the Sixth International Conference on Electronics, Communication and Aerospace Technology (ICECA 2022). IEEE.
4. Said, Y., Saidani, T., Smach, F., Atri, M., & Snoussi, H. (2012). Embedded Real-Time Video Processing System on ПЛИС [FPGA]. In A. Elmoataz et al. (Eds.), Lecture Notes in Computer Science (LNCS), vol. 7340, pp. 85–92. Springer-Verlag.
5. Ivashko, A., Zuev, A., Karaman, D., & Moškon, M. (2024). FPGA-Based Implementation of a Gaussian Smoothing Filter with Powers-of-Two Coefficients. Advanced Information Systems, Vol. 8, No. 2, pp. 39–46. DOI: 10.20998/2522-9052.2024.2.05

6. Molotkov A.A., Tretiakova O.N. On possible approaches to visualizing the process of selective laser melting // Scientific Visualization, 2019, Vol. 11, No. 4, pp. 1–12. DOI: 10.26583/sv.11.4.01
7. Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения [Image processing and analysis in machine vision tasks] // Физматкнига [Fizmatkniga], 2010. 672 p.
8. Кондратенко В.С., Третьякова О.Н., Шевченко Г.Ю. Разработка средств управления лазерным технологическим оборудованием с различными кинематическими схемами [Development of control tools for laser technological equipment with various kinematic schemes] // Вестник Московского авиационного института [Bulletin of Moscow Aviation Institute], 2015, Vol. 22, No. 2, pp. 121–131.
9. Молотков А.А., Сапрыкин Д.Л., Третьякова О.Н., Тужилин Д.Н. Программное обеспечение регистрации потоковых видеоданных FlexRecorder [FlexRecorder software for video stream registration]. State Registration Certificate of Computer Program No. 2021669002. Application No. 2021668322. Submission Date: 16 November 2021. Registration Date: 23 November 2021.
10. Третьякова О.Н., Молотков А.А., Семашко В.С. Программа FastDetector машинного зрения для создания детали в процессе селективного лазерного сплавления [FastDetector machine vision software for part creation in selective laser melting]. State Registration Certificate of Computer Program No. 2021669857. Application No. 2021669184. Submission Date: 26 November 2021. Registration Date: 3 December 2021.
11. Третьякова О.Н., Молотков А.А., Желябин И.А. Программа HarrisDetector машинного зрения для создания детали в процессе селективного лазерного сплавления [HarrisDetector machine vision software for part creation in selective laser melting]. State Registration Certificate of Computer Program No. 2021680165. Application No. 2021669247. Submission Date: December 2021. Registration Date: 3 December 2021.
12. Jia, Chuanmin, et al. "FPX-NIC: An FPGA-accelerated 4K ultra-high-definition neural video coding system." IEEE Transactions on Circuits and Systems for Video Technology, 32(9), 6385–6399, 2022.
13. Cong, Jason, et al. "FPGA HLS today: successes, challenges, and opportunities." ACM Transactions on Reconfigurable Technology and Systems (TRETS), 15(4), 1–42, 2022.
14. Bailey, Donald G. Design for Embedded Image Processing on FPGAs. John Wiley & Sons, 2023.
15. Cui, Beiyao, et al. "Design of Image Acquisition Circuit for Pixel Coupled Infrared Polarization Camera Based on FPGA." 2023 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC). IEEE, 2023.
16. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001. pp. I–511.