

УДК 004.77; 004.514; 004.932

СОЗДАНИЕ КЛИЕНТ-СЕРВЕРНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ INTERFAN ДЛЯ ЭФФЕКТИВНОГО АНАЛИЗА ИНТЕРФЕРЕНЦИОННЫХ ИЗОБРАЖЕНИЙ ПЛАЗМЕННЫХ ОБЪЕКТОВ

Т. Ф. Хирьянов¹, А. И. Хирьянова², Е. В. Паркевич², А. С. Селюков²⁻⁴

В статье представлено специализированное ПО для обработки результатов лазерной интерферометрии, позволяющее получать достоверные данные о параметрах мелкомасштабных сложноструктурированных плазменных образований, формирующихся при развитии наносекундного разряда в газообразной среде. Описана архитектура программы, её функционал, опции обработки, а также полученные с её использованием результаты физических исследований.

Ключевые слова: интерферометрия, цифровая обработка изображений, архитектура программного обеспечения.

Введение. В связи с бурным исследованием наносекундных высоковольтных разрядов [1] и развитием связанных с ними приложений возникает спрос на использование высокоразрешающих оптических методов диагностики газоразрядной плазмы. Одной из таких диагностик, показавшей свою высокую эффективность в получении надежных количественных данных о развивающейся сильноионизованной плазме, является интерферометрия с использованием лазерного излучения. Детальное описание данной диагностики в приложении к исследованию газовых разрядов можно найти в работах [2–8]. В то же время была отмечена существенная сложность обработки результатов интерферометрии ввиду необходимости оперирования большим количеством информа-

¹ МФТИ (Национальный исследовательский университет), 141701 Россия, Московская область, г. Долгопрудный, Институтский пер., 9; e-mail: khiryanov.tf@mipt.ru.

² ФИАН, 119991 Россия, Москва, Ленинский пр-т, 53.

³ Московский государственный технический университет им. Н. Э. Баумана, 105005 Россия, Москва, 2-я Бауманская ул., д. 5, стр. 1.

⁴ Московский политехнический университет, 107023 Россия, Москва, ул. Б. Семеновская, 38.

ции при анализе объектов с малым поперечным масштабом. В некоторых случаях требовалось определить масштаб дифракционной расходимости излучения, прошедшего исследуемый объект, на некотором расстоянии после него, или же учесть аппаратную функцию оптической системы. Комбинирование подходов для учёта ключевых факторов, способных заметным образом влиять на качество интерференционных изображений, а также методов обработки соответствующих изображений позволяет получить необходимый набор инструментов для обработки интерферограмм плазменных формирований, типичных для газовых разрядов. Создание такого набора актуально для многих исследователей и научных коллективов.

В связи с этим нами было разработано специализированное программное обеспечение (далее называемое Interferogram Analyser или “Interfan”) для исследования конкретного класса фазовых объектов с расширенными опциями инструментов для интерактивной обработки результатов интерферометрии. В данной работе мы вкратце описываем данную программу и полученные с её помощью результаты.

Техническое задание на разработку ПО. Обработка интерференционных изображений включает в себя, как правило, фильтрацию шумов и выравнивание интенсивности на масштабе всего интерференционного кадра, проведение трасс интерференционных линий, формирование двумерной карты сдвига фазы и её обработку в рамках решения обратной задачи. Фильтрация имеет целью облегчить поиск точек максимумов и минимумов интенсивности за счет снижения влияния шумов и искажений. Алгоритмический поиск точек-кандидатов на экстремумы производится для очищенного кадра, затем эти точки соединяются в непрерывные линии – трассы интерференционных полос. Выбор того или иного метода фильтрации, равно как и алгоритма трассировки полос, напрямую зависит от особенностей самого исследуемого объекта и тонкостей регистрации изображения. Поэтому здесь существенную роль играет возможность контроля получаемого результата, т. е. пользователь должен иметь возможность ознакомиться с результатами фильтрации и трассировки и, основываясь на этих результатах, решить, насколько эффективно использование того или иного алгоритма обработки кадра и, при необходимости, выбрать другой алгоритм или вручную откорректировать трассы в особых зонах интерференционных линий. То же самое касается и визуального контроля двумерной карты фазы, извлекаемой из интерферограммы, наглядное представление которой позволяет лучше определить локальный ход оси симметрии объекта, сформировать опорные срезы для решения обратной задачи (в случае плазменного объекта – восстановление профиля электронной плотности).

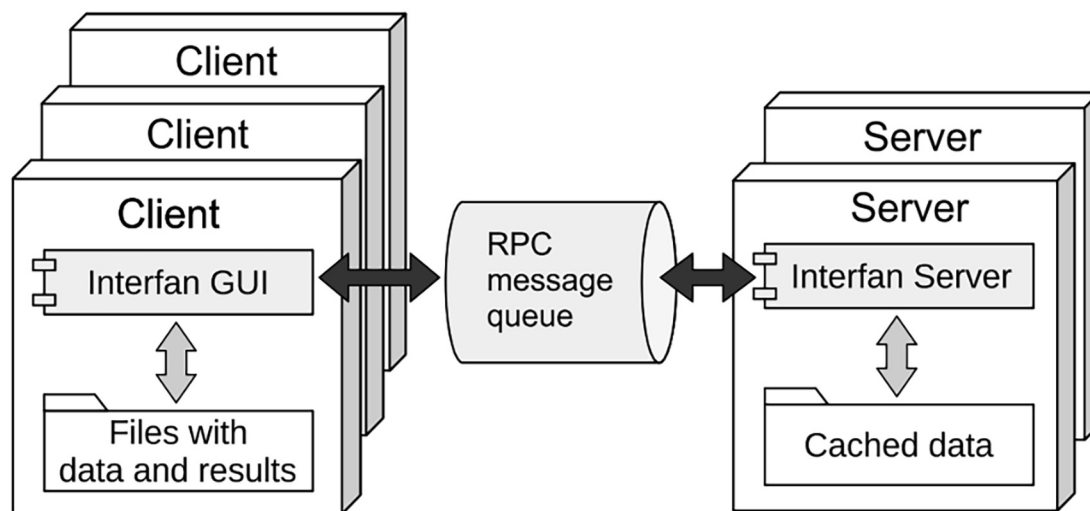


Рис. 1: Клиент-серверная архитектура Interfan.

Архитектура программы Interfan. Поскольку алгоритмы обработки интерференционных изображений вычислительно сложны, важным требованием к программе является возможность выполнять вычисления на удалённом компьютере высокой мощности или вычислительном кластере. Для реализации клиент-серверной архитектуры мы прибегнули к удалённому вызову процедур или, иначе говоря, Remote Procedure Calls (RPC) [12] (см. схему на рис. 1). Клиентская компонента программы имеет графический интерфейс (далее Interfan GUI), запускаемый на компьютере конечного пользователя. Серверная компонента программы (далее Interfan Server) устанавливается на физическом сервере и непрерывно находится в запущенном состоянии в ожидании вызова. Для возможности одновременной работы нескольких клиентов и нескольких физических серверов необходим менеджер очередей (он же “брокер сообщений”) [13], осуществляющий распределение работы и балансировку нагрузки.

Пользователь через графический интерфейс Interfan GUI имеет возможность изучить интерференционное изображение, выбрать соответствующие алгоритмы обработки и поставить задачу по очередному этапу обработки в соответствующую очередь. Брокер сообщений выбирает наименее загруженный из серверов, готовых обслужить вызов RPC. Затем Interfan Server на выбранном брокером физическом сервере берёт задание из очереди, запускает реализацию алгоритма и возвращает результат соответствующему клиенту. Для создания программного обеспечения Interfan был использован язык программирования Python, поскольку он адаптирован как для создания GUI,

так и для научных расчётов. Для преодоления узких мест в программе используется язык С как наиболее вычислительно эффективный и удобный для автоматического распараллеливания вычислений на многоядерных системах с общей памятью.

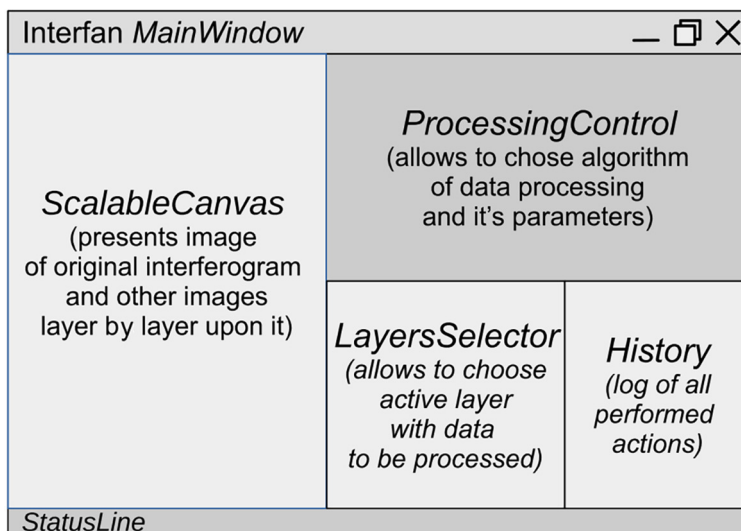


Рис. 2: Схема графического интерфейса *Interfan GUI*.

Пользовательский интерфейс Interfan GUI. Пользовательский интерфейс программы изображен на рис. 2. Левая половина окна *Interfan GUI* заполнена масштабируемым холстом *ScalableCanvas* для визуального контроля обработки интерференционного кадра. Вторая половина разделена на виджет контроля вычислений (*ProcessingControl*), виджет выбора слоя (*LayersSelector*), а также виджет истории (*History*).

Для информативности результата обработки выполняется послойное графическое отображение результатов вычислений поверх исходного интерференционного изображения с возможностью включения режима полупрозрачности. Базовые точки, трассы интерференционных линий, двумерная карта сдвига фазы, а также итоговая восстановленная карта электронной плотности, рассчитываемая по выбранным срезам (см. [10]), накладываются по очереди. Для управления слоями служит виджет выбора слоя, где можно включить или выключить видимость каждого слоя. Выбор активного слоя в списке слоёв переключает виджет контроля процессов вычислений в соответствующий режим: отображается форма выбора алгоритма обработки текущего класса данных и кнопка запуска вычислений. Виджет истории записывает в лог все действия, которые сказываются на изменении данных.

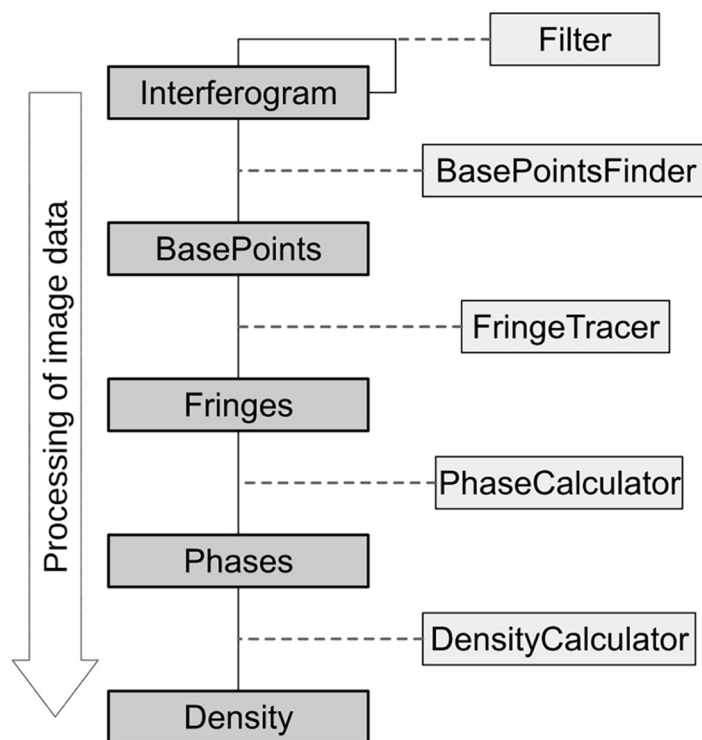


Рис. 3: Диаграмма классов данных и классов процессов.

Архитектура клиентского приложения. Компонента Interfan GUI строится в соответствии с объектно-ориентированным подходом [14]. На рис. 3 представлена диаграмма классов, при этом классы данных на диаграмме находятся в левом столбце, а классы процессов – в правом. Классы данных имеют направленную ассоциацию друг с другом в соответствии с процессом обработки данных (на рисунке он показан стрелкой): Interferogram → BasePoints → Fringes → Phases → Density. Класс Interferogram также ассоциирован сам с собой, поскольку наложение фильтра приводит к появлению нового объекта того же типа. Ответственность классов данных – хранение данных соответствующего вида, загрузка, отображение на холсте, сохранение в файл. Для одного типа данных одновременно допустимо существование в памяти нескольких объектов.

Классы процессов соответствуют этапам обработки. В них хранятся параметры обработки, отображаемые виджетом контроля вычислений. Классы процессов не обрабатывают данные самостоятельно, но посылают RPC запросы на вычисления в очередь сообщений и асинхронно обслуживают результаты, полученные из очереди. Успешное окончание процесса означает появление нового экземпляра класса данных и соответствующего ему нового слоя в списке слоёв виджета LayersSelector. Полученные данные

визуализируются на виджете ScalableCanvas поверх нижележащих слоёв. Удаление слоя из списка слоёв означает удаление соответствующего экземпляра класса данных из оперативной памяти.

Архитектура серверной компоненты. Компонента Interfan Server устроена по плоской архитектуре. Каждый алгоритм представляет собой функцию, написанную на Python, которая ожидает появления задач в соответствующей очереди сообщений. Получив сообщение, эта функция десериализует данные и запускает алгоритм на языке C. Полученные результаты возвращаются функции, написанной на языке Python, которая сериализует их и отправляет клиенту. В качестве коннектора клиентской и серверной компонент в проекте Interfan используется брокер сообщений RabbitMQ [13], который подходит к проекту как с точки зрения свободной лицензии, так и по своим характеристикам при работе в высоконагруженных системах.

Форматы файлов изображений. Для поддержки различных форматов изображений в проекте Interfan используется библиотека Pillow, что позволяет работать с форматами TIFF, JPEG, PGM, PPM и другими. Форматом по умолчанию выбран SPIDER, где на каждый пиксель отводится 32-битное число с плавающей точкой, что позволяет с высокой точностью сохранять фазовую карту и итоговую картину распределения электронной плотности плазменного объекта.

Использование Interfan для научных целей. Описанная программа была задействована для обработки результатов интерферометрии сильноионизованной плазмы при исследовании развития газового разряда. Информативность программы, возможность визуального контроля процесса обработки, а также высокая вычислительная мощность позволила значительно упростить процедуру обработки интерферограмм. Заложенные опции программы Interfan позволили учесть специфику регистрируемых плазменных объектов (пространственная неоднородность, наличие микроструктуры, малый поперечный масштаб плазменных формирований и т. д.), а также применить специализированные методы обработки интерферограмм (разработанные в работах [10, 11]), включая различные методы фильтрации шума, трассировку и последующую обработку двумерных карт сдвига фазы. Кроме того, программа предоставила возможность визуального контроля процесса обработки и ручной корректировки трасс интерференционных линий в зонах понижения контраста интерференционной картины и областях высоких градиентов плотностей. Такие эффекты характерны для процессов плазмообразования в приэлектродных областях и позволяют исследовать процессы фундаментальной физики возникновения сильноионизованных плазменных каналов в разряде. Высокая

вычислительная мощность серверной компоненты программы использовалась как для предварительной многонаправленной очистки кадра от шумов на основе попиксельного построения гистограмм зарегистрированной интенсивности в кадре, так и для трассировки ресурсоемкими методами, учитывающими локальное направление и ширину интерференционной полосы. Возможность использовать ресурсоемкие методики очистки, обрабатывающие каждый элемент кадра исходя из его индивидуальных характеристик зашумления и информацию о локальных характеристиках полос, позволила провести всесторонний анализ результатов обработки интерферограмм, что было недоступным для многих специализированных средств обработки данных, и в итоге получить надежные данные о параметрах плазмы в первые наносекунды от момента пробоя разрядного промежутка [5–7], информацию о микроструктуре отдельных развивающихся искр [8], и многое другое.

Научные результаты обработки и анализа интерферограмм были получены при поддержке гранта Российского научного фонда № 19-79-30086. Программный код был разработан при поддержке гранта Президента Российской Федерации МК-703.2020.2.

Л И Т Е Р А Т У Р А

- [1] I. Adamovich, S. D. Baalrud, A. Bogaerts, et al., *J. Phys. D: Appl. Phys.* **50**(32), 323001 (2017). DOI: 10.1088/1361-6463/aa76f5.
- [2] E. V. Parkevich, *Instrum. Exp. Tech.* **60**(3), 383 (2017). DOI: 10.1134/S0020441217030137.
- [3] E. V. Parkevich, S. I. Tkachenko, A. V. Agafonov, et al., *J. Exp. Theor. Phys.* **124**(4), 531 (2017). DOI: 10.1134/S1063776117030074.
- [4] E. V. Parkevich, A. I. Khirianova, A. V. Agafonov, et al., *J. Exp. Theor. Phys.* **126**(3), 422 (2018). DOI: 10.1134/S1063776118030160.
- [5] E. V. Parkevich, G. V. Ivanenkov, M. A. Medvedev, et al., *Plasma Sources Sci. Technol.* **27**(11), 11LT01 (2018). DOI: 10.1088/1361-6595/aaebdb.
- [6] E. V. Parkevich, M. A. Medvedev, A. I. Khirianova, et al., *J. Russ. Las. Res.* **40**(1), 56 (2019). DOI: 10.1007/s10946-019-09769-5.
- [7] E. V. Parkevich, M. A. Medvedev, G. V. Ivanenkov, et al., *Plasma Sources Sci. Technol.* **28**(9), 095003 (2019). DOI: 10.1088/1361-6595/ab3768.
- [8] E. V. Parkevich, M. A. Medvedev, A. I. Khirianova, et al., *Plasma Sources Sci. Technol.* **28**(12), 125007 (2019). DOI: 10.1088/1361-6595/ab518e.
- [9] E. V. Parkevich, M. A. Medvedev, A. S. Selyukov, et al., *Opt. Laser Eng.* **116**, 82 (2019). DOI: 10.1016/j.optlaseng.2018.12.014.

- [10] A. I. Khirianova, E. V. Parkevich, M. A. Medvedev, J. Russ. Laser Res. **40**(1), 48 (2019). DOI: 10.1007/s10946-019-09768-6.
- [11] A. I. Khirianova, E. V. Parkevich, M. A. Medvedev, J. Russ. Laser Res. **41**(2), 141 (2020). DOI: 10.1007/s10946-020-09859-9.
- [12] A. L. Ananda, B. H. Tay, E. K. Koh, SIGOPS Oper. Syst. Rev. **26**(2), 92 (1992). DOI: 10.1145/142111.142121.
- [13] B. Christudas, in *Practical Microservices Architectural Patterns* (Apress, Berkeley, CA, 2019), pp. 827-841. DOI: 10.1007/978-1-4842-4501-9_21.
- [14] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley, USA, 1994). ISBN: 0-201-63361-2.

Поступила в редакцию 10 сентября 2020 г.

После доработки 15 октября 2020 г.

Принята к публикации 16 октября 2020 г.