

Санкт-Петербургский государственный университет

Программная инженерия

Ташкинов Михаил Ильич

Система сбора и анализа пульса человека на основе данных с мобильных устройств

Выпускная квалификационная работа

Научный руководитель:
к. ф.-м. н., доцент Романовский К. Ю.

Рецензент:
Зам. ген. директора ООО "ПитерСофтвареХаус" Хитров Д. В.

Санкт-Петербург
2017

SAINT PETERSBURG STATE UNIVERSITY

Software Engineering

Mikhail Tashkinov

System for collection and analysis of human pulse based on data from mobile devices

Graduation Thesis

Scientific supervisor:
assistant professor Konstantin Romanovsky

Reviewer:
Deputy director general of "PiterSoftwareHouse" Ltd. Denis Khitrov

Saint Petersburg
2017

Оглавление

Введение	5
1. Постановка задачи	7
2. Обзор	8
2.1. Алгоритмы оценки пульса по камере	8
2.1.1. Алгоритмы основанные на подсчёте R-пиков . . .	8
2.1.2. Алгоритмы основанные на нахождении доминирующей частоты	8
2.1.3. Алгоритмы основанные на комбинации подходов .	10
2.2. Решения для получения данных с медицинских bluetooth устройств	10
2.3. Системы для расширенного анализа пульса	11
3. Архитектура	12
3.1. Функциональность системы	12
3.1.1. Оценка пульса с помощью камеры телефона . . .	12
3.1.2. Получение данных от bluetooth пульсометров . .	12
3.1.3. Хранение данных о сердцебиении	12
3.1.4. Дополнительный анализ сердцебиения	13
3.2. Компоненты системы	13
3.2.1. Получение данных о сердцебиении	13
3.2.2. Хранение данных	13
3.2.3. Анализирующие алгоритмы	14
3.2.4. Интеграция анализирующих алгоритмов	14
3.3. Используемые технологии	15
3.3.1. Приложение для ОС Android	15
3.3.2. Приложение для ОС iOS	15
3.3.3. Веб сервис	15
4. Оценка пульса с помощью камеры телефона	16
4.1. Частотная фильтрация	16

4.2.	Амплитудное выравнивание	17
4.3.	Уточнение пульса	18
4.4.	Оценка динамики пульса	18
4.5.	Детектор наличия пальца на камере	20
5.	Получение данных с bluetooth пульсометров	21
5.1.	Общее описание библиотеки	21
5.2.	API библиотеки	22
5.3.	Формат сериализации данных с устройств	23
6.	Система для интеграции алгоритмов анализа пульса	25
7.	Тестирование	27
7.1.	Алгоритм оценки пульса с помощью камеры телефона .	27
7.2.	Библиотека для получения данных с медицинских bluetooth устройств	28
7.2.1.	Юнит-тесты с тестовыми устройствами	28
7.2.2.	Приложение для автоматической проверки работы с конкретным устройством	29
7.2.3.	Приложение с частичной функциональностью для проверки работы с устройствами	29
7.3.	Интеграция алгоритмов анализа пульса	29
8.	Заключение	31
	Список литературы	32

Введение

Пульс является одним из самых важных медицинских показателей человека. В связи с этим, существует много ситуаций, когда людям нужно оценить свой пульс. Например, его важно знать спортсменам при тренировках. Существуют различные способы измерения пульса.

Пульс можно измерять вручную. Для этого нужно засечь отрезок времени с помощью секундомера и замерить количество пульсаций артерии за этот отрезок. Но такой способ измерения не очень удобен.

Также, пульс можно измерять с помощью специальных устройств: пульсоксиметров, пульсометров, умных часов и спортивных трекеров. Но данный способ требует покупки отдельного устройства, стоимость которого достаточно велика. К тому же, его может не оказаться под рукой в нужный момент.

В свою очередь, сегодня почти у каждого человека есть смартфон, который он постоянно носит с собой.

Таким образом, если смартфон научится считать пульс человека, то это сделает его самым доступным и удобным устройством для вычисления пульса. И действительно, мощность современных телефонов и наличие у них камеры со вспышкой позволяют создавать приложения для оценки пульса.

Они основаны на том, что при просвечивании пальца с помощью вспышки можно засечь приток крови к нему. Соответственно, подсчитав количество таких притоков, приложение может оценить пульс человека.

С другой стороны, для многих людей, в особенности спортсменов, помимо значения пульса в данный момент и его динамики, могут быть важными более продвинутые данные, которые можно получить из информации о сердцебиении.

Для получения таких данных нужно длительное измерение с получением подробной информации о сердцебиении. Соответственно, для этого нужен профессиональный пульсометр, который может каким-то образом автоматически передавать данные на телефон. Наиболее рас-

пространенным способом передачи данных с пульсометров на телефон в наше время является технология Bluetooth.

Алгоритмы для вычисления расширенных параметров, помимо подробной информации о сердцебиении, требуют затратных вычислений, которые не представляется возможным выполнять на мобильном телефоне. В связи с этим, для вычисления данных параметров требуется отправлять данные о пульсе на сервер, а затем получать с него результат.

Обычно, анализирующие алгоритмы разрабатываются специалистами по цифровой обработке сигнала и представляют собой программу на языке MATLAB. Очевидно, что для промышленного использования требуется создать оболочку для данной программы, которая будет получать входные данные по заданному протоколу, проверять их на корректность и передавать обратно результат. Лучше всего для этой цели подходит превращение данной программы в веб сервис, который предоставляет универсальное API для вычисления анализируемого параметра. В связи с чем интеграция алгоритма значительно упрощается.

Таким образом, для профессионального и доступного анализа сердцебиения требуется система с возможностью оценки пульса с помощью камеры телефона, возможностью получать данные с Bluetooth пульсометров, возможностью вычислять сложные параметры на сервере и удобной интеграцией алгоритмов, вычисляющих данные параметры.

1. Постановка задачи

Целью данной работы является разработка системы для оценки пульса, его динамики и предоставляющую возможности дополнительного анализа на основе информации с мобильных устройств.

Для выполнения этой цели нужно выполнить следующие задачи.

- Разработать архитектуру системы.
- Реализовать оценку пульса с помощью камеры телефона.
- Реализовать получение данных с bluetooth пульсометров.
- Реализовать систему для интеграции алгоритмов анализа пульса.
- Провести тестирование.

2. Обзор

2.1. Алгоритмы оценки пульса по камере

Алгоритмы оценки пульса по камере можно разделить на три категории:

- Алгоритмы основанные на подсчёте R-пиков.
- Алгоритмы основанные на нахождении доминирующей частоты.
- Алгоритмы основанные на комбинации этих подходов.

Рассмотрим их подробнее.

2.1.1. Алгоритмы основанные на подсчёте R-пиков

подавляющее большинство существующих алгоритмов этого типа опирается на подсчёт локальных экстремумов, соответствующих притоку крови к пальцу. Пульс вычисляется как $\frac{60 \times n}{t}$, где n - количество найденных пиков, а t - длина временного окна в секундах. Преимуществами алгоритмов данного типа является сравнительно небольшая вычислительная мощность и возможность оценки пульса в реальном времени. Основным недостатком таких алгоритмов являются возможные проблемы при детекции пиков. Ведь пропуск даже одного пика при небольшом временном окне приводит к значительному изменению конечного результата. К данному типу относится подавляющее большинство существующих алгоритмов оценки пульса по камере [1, 2, 4–6, 8–10] поскольку они приоритезируют возможность быстрой оценки пульса над корректной его оценкой при сильно зашумлённом измерении.

2.1.2. Алгоритмы основанные на нахождении доминирующей частоты

Алгоритмы данного типа [3, 7, 11] опираются на разложение сигнала на частоты и нахождении среди них доминирующей. Наиболее часто в подобных алгоритмах для разложения этой цели используется Быстрое

Преобразование Фурье¹, которое позволяет разложить дискретный сигнал на частоты и сопоставить каждой частоте её амплитуду. Преимуществом такого подхода является большая устойчивость оценки пульса к зашумлённости сигнала.

Одним из главных недостатком такого рода алгоритмов является зависимость дискретизации разложения от длины и частоты сигнала. Таким образом, для получения точного ответа таким алгоритмам требуется длительное измерение. В связи с этим, такие алгоритмы плохо проявляют себя при резком изменении пульса так как могут выдать только одну из частот, но не какое-либо промежуточное значение. А новое значение пульса будет определено алгоритмом только после того как оно станет доминирующим т.е. будет встречаться на большей части сигнала. Таким образом, изменение значение пульса такой алгоритм определит позже, чем алгоритмы первого типа. По этой же причине применение таких алгоритмов в режиме реального времени затруднено.

Пример спектра для проблемного измерения можно наблюдать на рисунке 1. На спектре видно два экстремума на частотах 70 уд/мин. и 85 уд/мин. Это связано с тем, что во время данного измерения пульс резко увеличился с 70 до 85 уд/мин.

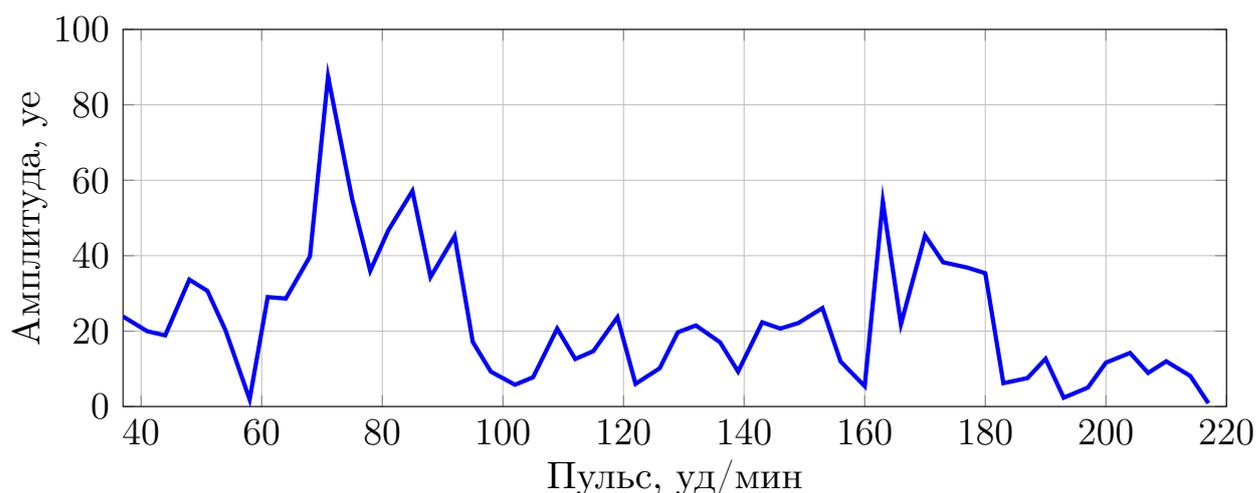


Рис. 1: Участок спектра при резком росте пульса

¹en.wikipedia.org/wiki/Fast_Fourier_transform

2.1.3. Алгоритмы основанные на комбинации подходов

Данный тип алгоритмов является комбинацией рассмотренных ранее алгоритмов в 2.1.1 и 2.1.2 и позволяет исправить указанные недостатки каждого алгоритма. Обычно это достигается за счёт выполнения двух алгоритмов разных типов последовательно или параллельно. Указанный подход позволяет с одной стороны оценивать пульс для достаточно зашумлённых измерений, а с другой стороны проводить измерения в реальном времени.

Описанный в данной работе алгоритм оценки пульса относится именно к последнему типу алгоритмов.

2.2. Решения для получения данных с медицинских bluetooth устройств

В наше время многие медицинские устройства, такие как: весы, тонометры, глюкометры, пульсометры, пульсоксиметры умеют передавать данные об измерениях с помощью технологии bluetooth². Несмотря на то, что уже существуют стандартные протоколы³ для передачи медицинских измерений с помощью bluetooth, многие производители предпочитают использовать собственные протоколы для передачи данных. В первую очередь это связано с тем, что стандартизированные протоколы появились сравнительно недавно. Ещё одной причиной является желание производителя запретить получение данных с их устройств другими приложениями.

В итоге многие производители медицинских bluetooth устройств разрабатывает своё приложение для получения с них данных^{4,5}. Также, существует несколько приложений, которые поддерживают небольшой набор устройств⁶. Обычно такие приложения поддерживают устройства исключительно со стандартным протоколом.

²www.bluetooth.com

³www.bluetooth.com/specifications/adopted-specifications

⁴play.google.com/store/apps/details?id=com.foracare.ifora

⁵play.google.com/store/apps/details?id=com.taidoc.tdlink.bp

⁶play.google.com/store/apps/details?id=com.mhealth.VitalSignsMonitor

Одним из лидеров в данной области является компания MedM⁷, которая поддерживает большое количество устройств от различных производителей с различными протоколами передачи данных.

Также, сейчас на рынке появляются различные интеграторы приложений такие как HealthKit⁸ компании Apple, sHealth⁹ компании Samsung и GoogleFit¹⁰ от Google. К сожалению, все они позволяют работать исключительно с bluetooth устройствами, которые передают данные с помощью стандартного протокола.

2.3. Системы для расширенного анализа пульса

Существует не так много систем, которые позволяют продвинуто анализировать данные сердцебиения. К сожалению, все они либо предусматривают работу с крайне ограниченным количеством пульсометров, либо не предусматривают работу с устройствами в принципе, а проводят анализ по экспортированному измерению.

Для примера можно взять систему Welltory¹¹, которая анализирует уровень стресса на основе постоянных измерений сердцебиения и на основе этих данных рекомендует изменения в распорядке дня. Она позволяет собирать данные о пульсе с помощью камеры и четырёх устройств анализирующих вариабельность сердечного ритма. Таким образом, для полноценного использования такой системы с большой вероятностью нужно приобрести новые устройства даже если у вас уже есть свои. Что безусловно делает использование подобных систем крайне неудобным.

⁷medm.com

⁸developer.apple.com/healthkit/

⁹health.apps.samsung.com/

¹⁰www.google.com/fit/

¹¹welltory.com

3. Архитектура

3.1. Функциональность системы

В целом система обладает следующими функциями.

- Оценивает пульс при помощи камеры телефона.
- Получает данные от bluetooth пульсометров.
- Хранит данные о сердцебиении.
- Дополнительно анализирует сердцебиения.

Первые 3 функции реализованы в мобильном приложении. Последняя функция - на выделенном сервере. Разберём эти функции подробнее.

3.1.1. Оценка пульса с помощью камеры телефона

Пользователь прислоняет палец к камере телефона, закрывая одновременно камеру и вспышку. В течении примерно 15 секунд приложение собирает данные с камеры телефона и затем оценивает значение пульса пользователя. Примерно через каждые следующие 3 секунды приложение обновляет значение пульса. Приложение определяет приложен ли палец к камере телефона. И как только он убран останавливает измерение.

3.1.2. Получение данных от bluetooth пульсометров

Пользователь подключает к телефону с помощью приложения свой пульсометр с поддержкой технологии bluetooth. После этого, данные о сердцебиении пользователя с пульсометра получаются приложением автоматически.

3.1.3. Хранение данных о сердцебиении

Данные о сердцебиении пользователя сохраняются в приложении на телефоне. Пользователь может просматривать их, а также отправлять на дополнительный анализ.

3.1.4. Дополнительный анализ сердцебиения

Пользователь может отправить данные о сердцебиении на детальный анализ на сервер. Алгоритм детального анализа может требовать данных при специальном измерении. Например при последовательном увеличении нагрузки каждую минуту. Соответственно, для некоторых сохранённых данных о сердцебиении провести детальный анализ не представляется возможным. Результат детального анализа хранится вместе с исходной информацией о сердцебиении.

3.2. Компоненты системы

Система состоит из нескольких компонент.

- Компонента получения данных о сердцебиении.
- Компонента хранения данных о сердцебиении.
- Анализирующие алгоритмы.
- Компонента интеграции анализирующих алгоритмов.

Рассмотрим их подробнее.

3.2.1. Получение данных о сердцебиении

Данная компонента отвечает за получение данных о сердцебиении с bluetooth устройств, а также оценку пульса и его динамики с помощью камеры телефона. Получение данных с bluetooth пульсометров реализовано в качестве подключаемой библиотеки для ОС Android и iOS. Алгоритм оценки пульса по камере телефона реализована как модуль на C++, который на основе данных с камеры оценивает пульс. Получение данных с камеры телефона осуществляется платформозависимо.

3.2.2. Хранение данных

Данная компонента отвечает за сохранение данных о сердцебиении, а также результатов дополнительного анализа.

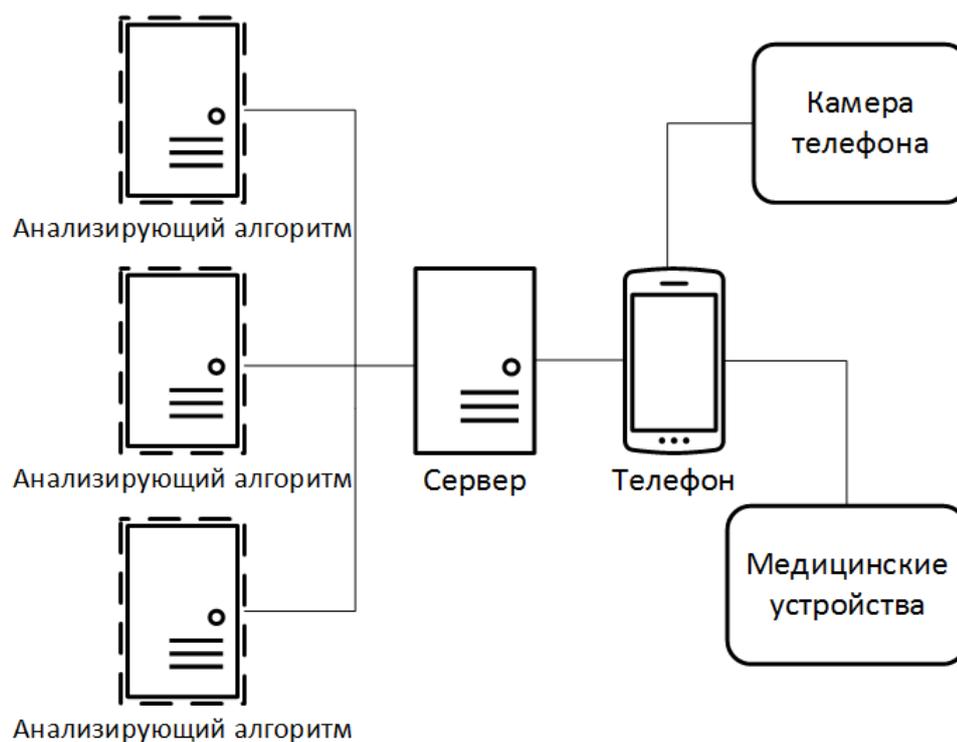


Рис. 2: Общая архитектура системы

3.2.3. Анализирующие алгоритмы

Анализирующий алгоритм отвечает за расширенный анализ пульса и представляет собой программу на языке MATLAB. Она просто принимает входные данные и по ним синхронно выдаёт ответ. Чаще всего, валидации входных данных в ней не предусмотрено. В связи с высокой ресурсоёмкостью алгоритмов они вынесены из мобильного приложения на сервер.

3.2.4. Интеграция анализирующих алгоритмов

Данная компонента отвечает за интеграцию анализирующих алгоритмов, рассмотренных в предыдущей части. Это включает в себя: предоставление API для взаимодействия с алгоритмом для других компонент системы, а также, потенциально, для других систем. Также она проверяет входные данные на корректность и выдаёт соответствующие ошибки.

3.3. Используемые технологии

3.3.1. Приложение для ОС Android

Несмотря на то, что стандартным языком для написания приложений для ОС Android является язык Java, для реализации приложения на ОС Android был выбран язык Kotlin компании JetBrains¹². Данный язык предоставляет полную совместимость с Java, что делает возможным его использование для написания Android приложений. В свою очередь, Kotlin имеет более короткий синтаксис и встроенную null безопасность, что делает разработку на нём Android приложений гораздо удобнее и эффективнее.

3.3.2. Приложение для ОС iOS

Для реализации приложения на ОС iOS был выбран язык Objective C поскольку он предоставляет совместимость с C++. А также, в связи с тем, что у меня уже был опыт разработки на Objective C.

3.3.3. Веб сервис

Для реализации веб сервиса была выбрана платформа Node.js из-за того, что она является самой популярной платформой для реализации задач подобного рода. К тому же, она имеет широкий набор библиотек, которые сильно упрощают процесс разработки.

¹²kotlinlang.org

4. Оценка пульса с помощью камеры телефона

Для получения пульса и его динамики с помощью камеры телефона был взят алгоритм, который был разработан мной в качестве курсовой работы[12], но с некоторыми доработками. Далее представлены наиболее важные из них.

4.1. Частотная фильтрация

В изначальном сигнале, полученном с камеры, довольно большое количество шума. Одним из способов частичного избавления от него является частотная фильтрация сигнала полосовым фильтром в диапазоне частот, соответствующем допустимому пульсу человека. В данном алгоритме за допустимый диапазон были приняты значения от 45 уд/мин до 220 уд/мин т.е. от 0,(6) Гц до 3,(6) Гц.

Пример работы фильтрации частоты представлен на рисунках 3 и 4.

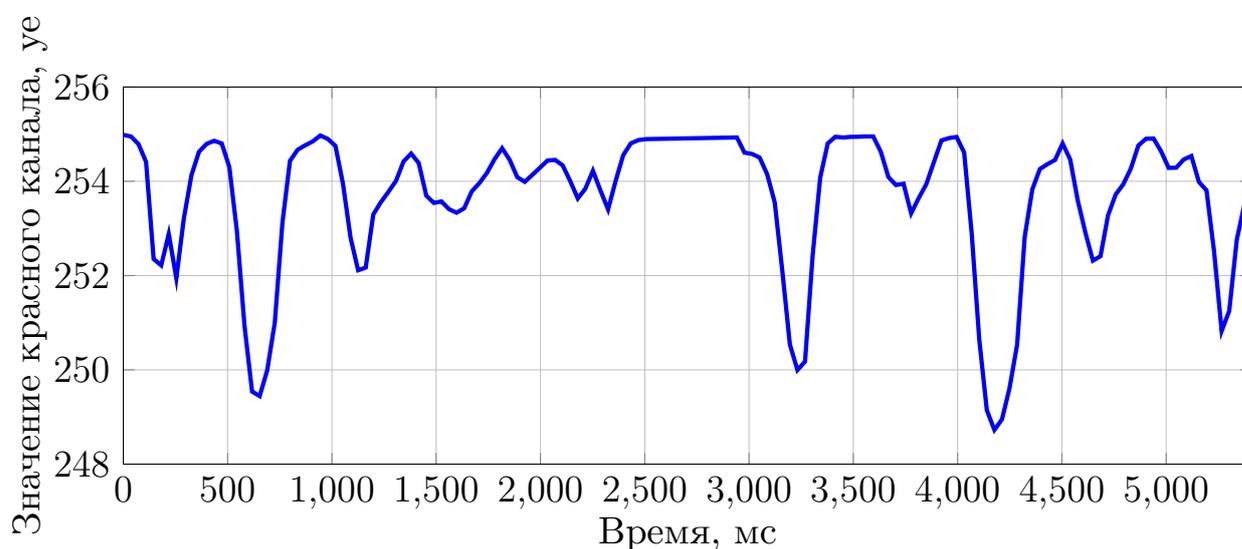


Рис. 3: Пример исходного сигнала



Рис. 4: Пример отфильтрованного по частоте сигнала

4.2. Амплитудное выравнивание

Прошлая версия алгоритма была чувствительна к шуму к высокой амплитудой. В данной версии проводится выравнивание амплитуды на разных участках сигнала.

Выравнивание амплитуд проводится по следующему алгоритму:

1. Сигнал делится на отрезки такие, что конец и начало каждого отрезка находятся в одинаковых частях R-пика. В данном алгоритме в качестве концов отрезка выбираются переходы сигнала через 0.
2. Для каждого отрезка вычисляется энергия по формуле $E_i = \sum_{j=k_i}^{m_i} x_j^2$, где E_i - энергия i -го отрезка, k_i и m_i - начало и конец i -го отрезка, а x_j - значение сигнала в точке j .
3. Находится отрезок с максимальной энергией $E_{max} = \max_{0 \leq i \leq n} E_i$.
4. Итоговое значение точки сигнала вычисляется как $y_j = x_j \times \frac{E_{max}}{E_i}$, где $k_i \leq j \leq m_i$.

Таким образом, энергии всех отрезков становятся одинаковыми. Пример работы выравнивания амплитуды можно наблюдать на рисунках 4 и 5.

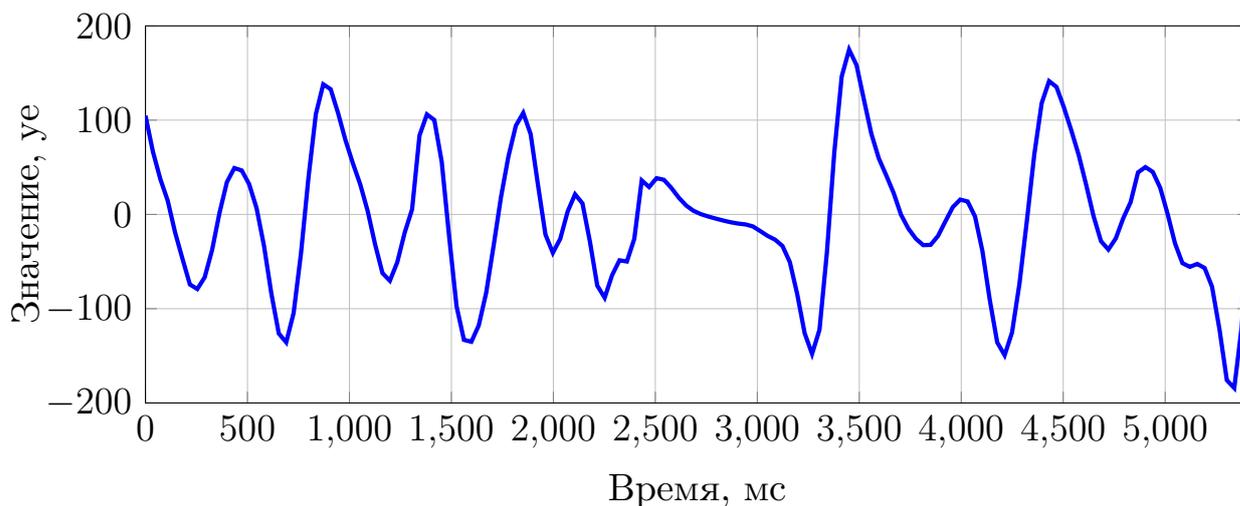


Рис. 5: Пример сигнала с выровненной амплитудой

4.3. Уточнение пульса

Предыдущая версия алгоритма считала значением пульса доминирующую частоту на промежутке. Но это не всегда корректно. Например, если пульс на данном промежутке резко изменился. В таком случае прошлая версия алгоритма выдала либо пульс до изменения, либо после, но не реальное значение пульса.

Для решения этой проблемы после получения доминирующей частоты исходный сигнал заново фильтруется, но с окном частот ± 20 уд/мин. от результата предыдущего алгоритма. Таким образом получается сигнал гораздо больше похожий на синусоидальный. После этого для получения точного значения пульса достаточно посчитать количество пиков в данном окне. Для этого, например, достаточно правильно проинтерпретировать информацию о переходах сигнала через 0.

Пример усиленной фильтрации сигнала представлен на рисунках 3 и 6.

4.4. Оценка динамики пульса

Часто не достаточно знать одно отдельное значение пульса. Для получения полной картины требуется знать динамику его изменения со временем. Это, например, важно для спортсменов чтобы узнать на-

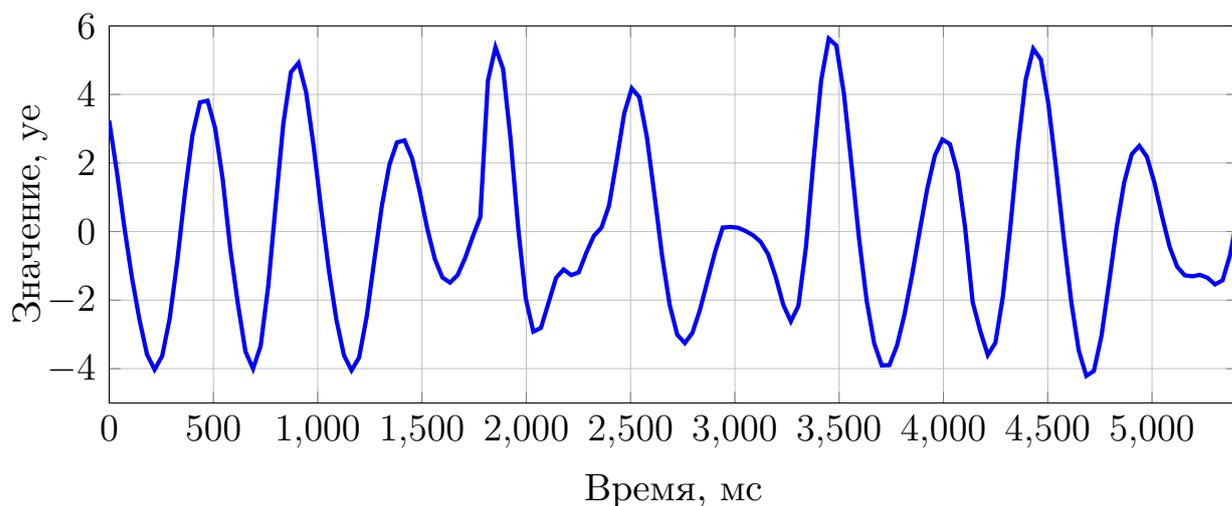


Рис. 6: Пример сигнала с усиленной фильтрацией

сколько быстро их сердечный ритм приходит в норму после окончания физической нагрузки или насколько быстро он возрастает при её начале.

Прошлый алгоритм пришлось дополнительно приспособить для оценки динамики. Основной целью изменений было максимальное переиспользование данных при вычислении следующего значения пульса. Таким образом для вычисления следующего значения в алгоритме переиспользуется:

- Интерполированные данные. Новые данные интерполируются с использованием того-же шага интерполяции в реальном времени. Таким образом заново интерполировать все данные не требуется.
- Информация о разложении по амплитудам. Для этого существующее разделение измерения на пересекающиеся отрезки не пересматривается, а лишь дополняется новыми. Недостатком такого подхода является то, что следующее значение пульса возможно получить не раньше получения нового отрезка. Т.е. до получения ≈ 2 новых секунд измерения.
- Информация об отрезках и их энергиях при выравнивании амплитуд, а также данные с уже выровненной амплитудой. Таким

образом, при вычислении следующего значения часто не требуется пересчитывать предыдущие данные.

4.5. Детектор наличия пальца на камере

Для удобства пользователя был придуман дополнительный алгоритм для определения наличия пальца. По каждому кадру, полученному с камеры, помимо среднего значения красного канала вычисляется среднее значение красного и зелёного. Наличием пальца на камере для данного кадра считается среднее значение красного канала > 128 , а синего и зелёного < 128 . Данные показатели оцениваются для кадров из последних 5 секунд измерения, и если количество кадров для которых данный критерий выполняется превышает 50%, то считается, что пользователь убрал палец с камеры и измерение прекращается.

5. Получение данных с bluetooth пульсометров

Медицинские bluetooth устройства производятся большим количеством компаний и зачастую имеют различные протоколы передачи данных. Таким образом реализация поддержки получения данных с обширного количества пульсометров затруднительна.

С другой стороны, в платформе MedM компании MedM реализована поддержка большого количества протоколов получения данных с медицинских устройств. Но работа с ними не отделена от самой платформы. Таким образом, для использования готовой реализации протоколов получения данных с bluetooth медицинских устройств было решено выделить работу с ними из платформы MedM в отдельную библиотеку.

5.1. Общее описание библиотеки

Библиотека представляет собой стандартную подключаемую библиотеку для платформ iOS и Android. Т.е. файл с форматом `.framework` для iOS и `.aar` для Android.

Она предоставляет доступ к основным операциям работы с bluetooth устройствами:

- Поиск.
- Добавление.
- Получение данных.

В связи с тем, что библиотека позволяет получать данные о различных видов измерений было решено сериализовывать данные в формат XML. Это позволяет предоставлять одинаковый API для всех видов измерений, а также позволяет пользователю самому преобразовывать данные в удобное для него представление.

Основная логика работы реализована в платформонезависимом коде как часть платформы MedM. Платформозависимо реализована лишь

тонкая обёртка, которая перенаправляет запросы в платформу. Это позволяет легко поддерживать несколько платформ, а также заметно упрощает поддержку новых.

5.2. API библиотеки

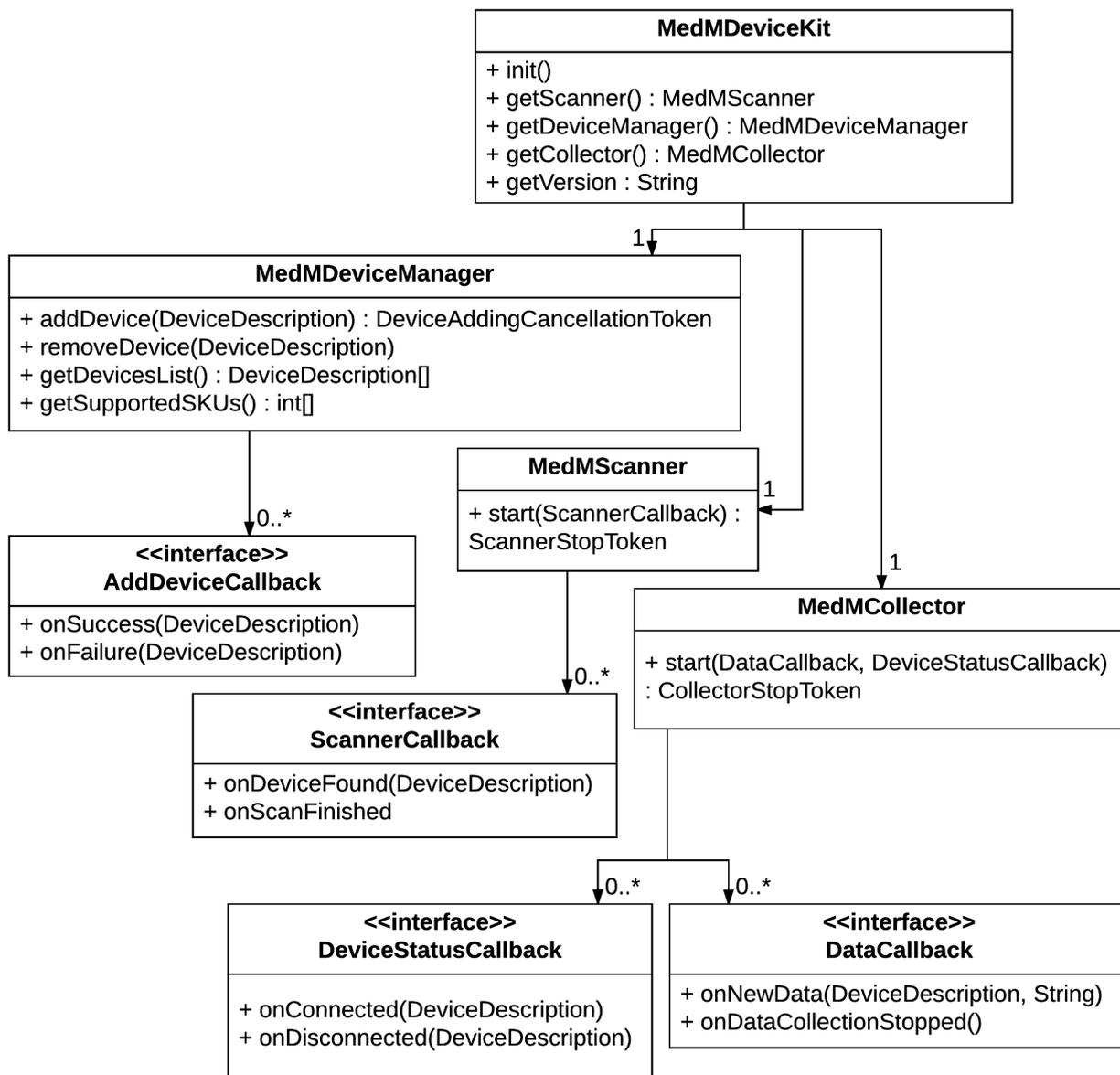


Рис. 7: Упрощённая структура API библиотеки

Упрощённая структура API библиотеки представлена на рисунке 7. Единственной доступной точкой входа для пользователя является класс MedMDeviceKit. Он хранит ссылки на MedMDeviceManager,

MedMScanner, MedMCollector, которые всегда существуют в единственном экземпляре и выдаются пользователю по запросу. Каждый из них выполняет свою функцию: управление списком добавленных устройств, поиск устройств и получение данных соответственно.

Отдельно стоит отметить, что все асинхронные операции (добавление устройства, сбор данных, поиск устройств) можно отменить. Это реализовано с помощью специальных токенов. При старте каждой такой операции пользователю выдаётся токен с помощью которого он может остановить операцию, соответствующую данному токену. Хотя на данный момент библиотека подразумевает выполнение только одной операции, такой механизм позволяет в будущем поддерживать несколько одновременных операций. К тому же, это не позволяет остановить операцию до её начала, а также после её окончания.

5.3. Формат сериализации данных с устройств

Как уже говорилось выше, данные с устройств сериализуются в XML. Данный формат был выбран поскольку он позволяет универсально представлять сильно различающиеся данные, поскольку в платформе MedM уже существовала реализация построения XML документов, а также для максимальной схожести на существующее серверное API платформы¹³. Сериализованное измерение всегда содержит время начала измерения по UTC¹⁴ и местному времени, а также разницу между ними. Также, оно может содержать подробные сведения об устройстве, если на момент получения данных платформа MedM их знает.

Все типы измерений можно разделить на два вида: точечные и поточные. Точечное измерение содержит один фиксированный набор значений, а поточное - несколько последовательных. Например, поточное измерение пульса может содержать информацию о последовательных измерениях пульса через каждые 5 секунд в течении 5 минут. Передавать такую информацию в формате XML довольно затратно. Напри-

¹³health.medm.com/docs/api/v3/index.html

¹⁴en.wikipedia.org/wiki/Coordinated_Universal_Time

мер, из-за того что XML теги будут занимать гораздо больше места, чем сами хранимые значения. В связи с этим, было решено отдавать поточные измерения в том же виде, в котором они хранятся в платформе MedM и передаются через серверное API. А именно в формате JSON.

Пример сериализованных данных для поточного измерения пульса приведён ниже.

```
<measurements-heartrate-stream>
  <measured-at>2017-05-01T20:59:59Z</measured-at>
  <measured-at-local>2017-05-01T23:59:59+03:00</measured-at-local>
  <measured-at-utc-offset>180</measured-at-utc-offset>
  <chunks>
    <chunk>
      <position>0</position>
      <start>0</start>
      <end>0</end>
      <heartrate>
        {'irregular':{'pulse':[79,0], 'pulse_quality':[255,0]}}
      </heartrate>
    </chunk>
  </chunks>
  <sensor>
    <name>Mambo HR</name>
    <model>405B6</model>
    <address>C8:46:B2:A6:E5:4F</address>
    <hardware>A002</hardware>
    <device-db-id>383</device-db-id>
    <firmware>A091</firmware>
    <device-db-title>Mambo</device-db-title>
    <serial>4FE5A6B246C8</serial>
    <manufacturer>Transtek</manufacturer>
  </sensor>
</measurements-heartrate-stream>
```

6. Система для интеграции алгоритмов анализа пульса

Алгоритмы для продвинутого анализа пульса чаще всего разрабатываются специалистами по цифровой обработке сигнала на языке MATLAB и представляют собой скорее прототип, чем программу пригодную для промышленного использования. В них зачастую полностью отсутствует валидация входных данных, а также сообщения об ошибках. В связи с этим, для таких алгоритмов приходится создавать обёртку, которая предоставляет API для отдачи данных на анализ и получения результата.

К тому же, такие алгоритмы часто требуют большой вычислительной мощности, что препятствует их интеграции в мобильное приложение. В связи с этим, наиболее простым вариантом для их интеграции в систему является реализация их как веб сервисов.

Одним из важных преимуществ такого подхода является возможность затем интегрировать алгоритмы с другими системами.

Общая схема интеграции алгоритма представлена на рисунке 8.

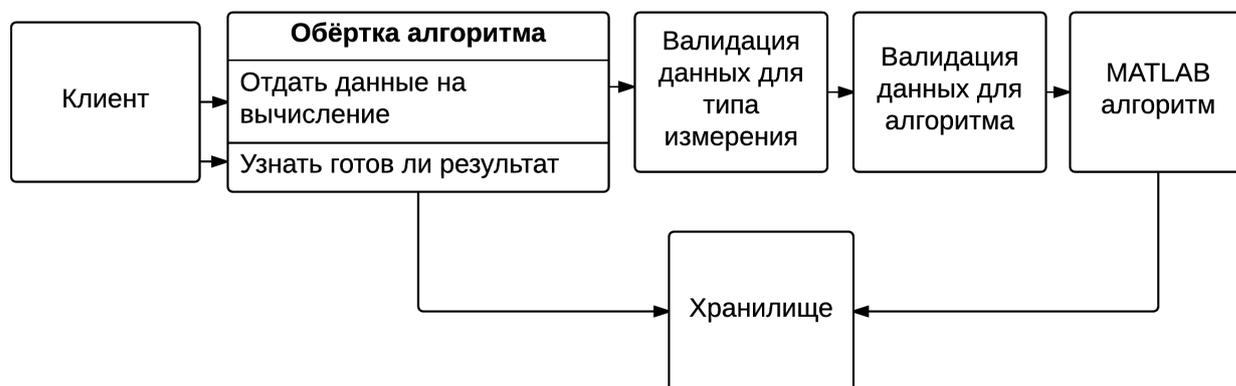


Рис. 8: Структура интеграции алгоритма

Клиент посылает POST запрос с прикрепленными данными об измерении и, возможно, некоторыми дополнительными параметрами для алгоритма. При получении данные проверяются сначала на соответствие заявленному типу измерения, а затем на соответствие заявленно-

му алгоритму. Отдельные шаги валидации сделаны для более простого интегрирования новых алгоритмов в систему. Например, для поддержки нового алгоритма для уже поддержанного типа измерения достаточно заменить MATLAB алгоритм и модуль проверки данных для алгоритма.

После проверки данных сервис придумывает уникальный идентификатор операции и возвращает либо адрес, придуманный на основе данного идентификатора, по которому затем можно будет получить результат анализа, либо сообщение об ошибке валидации с описанием проблемы. Этот механизм создан в связи с тем, что анализ потенциально может занимать длительное время. Соответственно отвечать синхронно не представляется возможным.

После успешной валидации входных данных они сохраняются на сервере, помещаются в очередь и ожидают передачи в анализирующий алгоритм. Анализирующий алгоритм сохраняет результаты анализа с соответствующим идентификатором запроса в хранилище.

По запросу на выданный адрес клиент может получить один из четырёх результатов:

- Сообщение о том, что данные всё ещё обрабатываются.
- Результат анализа.
- Сообщение об ошибке во время анализа данных.
- Сообщение о том, что адрес не существует.

Наличие результата проверяется по наличию записи с соответствующим идентификатором в хранилище. После того как пользователь обратился за ним и получил результат, он и сохранённые входные данные сразу же удаляются. Это сделано для экономии ресурсов на сервере.

В качестве примера анализирующего алгоритма был реализован алгоритм анализа анаэробного порога.

7. Тестирование

Для каждой компоненты системы было проведено тестирование. Алгоритм оценки пульса по камере был сравнён с его предыдущей реализацией, для библиотеки для получения данных с медицинских устройств было разработано три тестовых приложения, а тестирование системы интеграции алгоритмов анализа пульса было разбито на три модуля в соответствии с разбиением в реализации. Рассмотрим это подробнее:

7.1. Алгоритм оценки пульса с помощью камеры телефона

Результат работы алгоритма сравнивался с измерениями пульсоксиметра Nonin 3150¹⁵. Измерения проводились на людях европеоидной расы. Это важно, поскольку результаты работы такого рода алгоритмов могут зависеть от цвета кожи. Измерения проводились при неподвижном положении руки т.к. движение руки, особенно на улице, может привести к изменению освещённости с одинаковой частотой (например, движение руки при беге), что выльется в неправильную оценку пульса.

В тестировании принимало участие около 20 человек различных возрастных групп.

В таблице 1 приведён сравнительный результат тестирования текущей версии алгоритма с предыдущей.

Стоит отметить, что погрешность 5уд/мин является допустимой для пульсометров так как они оценивают пульс опосредованно (на пальце, запястье), в отличии от мониторов сердечного ритма, которые непосредственно измеряют количество сокращений сердца.

¹⁵www.nonin.com/PulseOximetry/Wrist-Worn/WristOx23150

Пульс	% измерений с относительной погрешностью < 5 уд/мин	
	Прошлый алгоритм	Новый алгоритм
Обычный (<100 уд/мин)	86	90
Повышенный (<140 уд/мин)	85	87
Высокий (>140 уд/мин)	85	87

Таблица 1: Сравнительный результат алгоритма с прошлой версией

7.2. Библиотека для получения данных с медицинских bluetooth устройств

Библиотеку для работы с bluetooth устройствами довольно трудно протестировать поскольку с одной стороны работу с множеством других устройств трудно проверить автоматически, а с другой - функциональность библиотеки слишком обширна чтобы быстро разработать тестовое приложение для Android и iOS, которое будет использовать все возможности библиотеки. В связи с этим тестирование библиотеки разделено на три части. Рассмотрим их подробнее.

7.2.1. Юнит-тесты с тестовыми устройствами

В платформе MedM существуют тестовые устройства, которые эмулируют работу реальных bluetooth устройств. С их использованием возможно написать классические юнит-тесты для проверки всей функциональности библиотеки. Проблема заключается в том, что для реальных устройств многие операции выполняются гораздо сложнее, чем для тестовых. Например, операция добавления может занимать около 30 секунд. Соответственно, такие тесты защищают лишь от малой части ошибок. К тому же, юнит тесты проверяют логику работы находясь

внутри библиотеки, а не используя уже собранную версию.

7.2.2. Приложение для автоматической проверки работы с конкретным устройством

Данное приложение проверяет большинство функций библиотеки для конкретного устройства. Из-за того что оно работает с реальным устройством, оно требует участия человека в тестировании, который будет включать соответствующие режимы на приборе и проводить измерение. Минусом такого подхода является то, что не всё при работе с реальными устройствами можно проверить автоматически. Например корректность отдаваемых значений.

7.2.3. Приложение с частичной функциональностью для проверки работы с устройствами

Данное приложение предназначено для проверки базовой функциональности библиотеки при работе с различными устройствами. Оно поддерживает поиск, добавление, удаление устройств, сбор данных с них и отображение данных. Это позволяет наиболее полно проверить работу с устройствами, но не проверяет все возможности библиотеки.

7.3. Интеграция алгоритмов анализа пульса

Тестирование системы для интеграции алгоритмов расширенного анализа пульса представляет собой набор юнит-тестов и разбито на модули как и реализация:

- Тесты для проверки валидации данных для типа измерения. Представляют собой набор корректных и некорректных данных для конкретного измерения с ожидаемыми ответами, включая сообщения об ошибках.
- Тесты для проверки валидации данных для алгоритма. Представляют собой набор корректных и некорректных данных для конкретного алгоритма. Причём всегда корректных для данного типа

измерения. В них, например, может отсутствовать необходимый для алгоритма дополнительный параметр. Также содержат ожидаемый ответ вплоть до конкретного сообщения об ошибке.

- Тесты для проверки работы алгоритма. Представляют собой набор корректных данных с ожидаемыми результатами для конкретного алгоритма.

Благодаря такой модульной структуре добавить тесты для нового типа измерения или нового алгоритма достаточно просто, так как это не требует заметного изменения остальной системы тестирования.

8. Заключение

В рамках данной выпускной квалификационной работы были получены следующие результаты.

- Разработана архитектура системы.
- Улучшена оценка пульса с помощью камеры телефона.
- Получение данных с bluetooth устройств выделено в библиотеку.
- Реализована система для интеграции алгоритмов анализа пульса.
- Проведено тестирование.

Список литературы

- [1] David D. McManus Jinseok Lee Oscar Maitas-Nada Esa Rahul Pidikiti Alex Carlucci Josephine Harrington Eric Mick Ki H. Chon. A novel application for the detection of an irregular pulse using an iPhone 4S in patients with atrial fibrillation. — 2013.
- [2] Fan Xiangmin, Wang Jingtao. BayesHeart: A Probabilistic Approach for Robust, LowLatency Heart Rate Monitoring on Camera Phones. — 2015.
- [3] Improved heart rate detection using smart phone / Arpan Pal, Aishwarya Visvanathan, Anirban Dutta Choudhury, Aniruddha Sinha. — 2014.
- [4] Laure Denis, Paramonov Ilya. Improved Algorithm for Heart Rate Measurement Using Mobile Phone Camera. — 2013.
- [5] Lenskiy Artem A., Aitzhan Yerlan. Extracting Heart Rate Variability from a Smartphone Camera. — 2013.
- [6] A Novel Method to Detect Heart Beat Rate Using a Mobile Phone / Pelegris P., Banitsas K., Orbach T., Marias K. — 2010.
- [7] A Novel Method to Detect Heart Beat Rate Using a Mobile Phone / Arpan Pal, Aniruddha Sinha, Anirban Dutta Choudhury et al. — 2013.
- [8] Real time heart rate variability assessment from Android smartphone camera photoplethysmography: Postural and device influences / F. Guede-Fernández, V. Ferrer-Mileo, J. Ramos-Castro et al. — 2015.
- [9] Rong-Chao Peng Xiao-Lin Zhou Wan-Hua Lin Yuan-Ting Zhang. Extraction of Heart Rate Variability from Smartphone Photoplethysmograms. — 2015.
- [10] Vikram Chandrasekaran B.E. Measuring Vital Signs Using Smart Phones. — 2010.

- [11] Yuriy Kurylyak Francesco Lamonaca Domenico Grimaldi. Smartphone-Based Photoplethysmogram Measurement. — 2012.
- [12] М.И. Ташкинов. Оценка пульса с помощью камеры мобильного телефона. — 2016.