

Санкт-Петербургский государственный университет

Кафедра системного программирования

Ташкинов Михаил Ильич

Оценка пульса с помощью камеры
мобильного телефона

Курсовая работа

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

Санкт-Петербург
2016

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Алгоритмы, основанные на подсчёте пиков	7
2.2. Алгоритмы, основанные на подсчёте пиков с дополни- тельной логикой	8
2.3. Алгоритмы, основанные на нахождении доминирующей частоты сигнала	9
2.4. Алгоритмы, основанные на анализе спектра в плавающем окне	10
3. Разработка	12
3.1. Приложение для сбора тестовых данных	12
3.1.1. Выбор инструментов	12
3.1.2. Создание приложения	12
3.2. Алгоритм	13
3.2.1. Получение среднего значения красного канала . .	14
3.2.2. Интерполяция сигнала	14
3.2.3. Разбиение на пересекающиеся отрезки	15
3.2.4. Вычисление спектра сигнала	15
3.2.5. Отбраковка шума	15
3.2.6. Вычисление пульса	17
3.3. Внедрение алгоритма в приложение	18
3.4. Тестирование	18
3.5. Внедрение алгоритма в платформу MedM	18
3.5.1. Выбор инструментов	18
3.5.2. Внедрение	19
4. Результаты	21
Список литературы	22

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 является неудобным.

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 постоянно находится при себе.

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 подсчёта 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.



Рис. 1: Процесс измерения

В наши дни становятся популярными платформы для автоматического сбора данных с медицинских устройств и сохранения их в облачных хранилищах. Они обеспечивают доставку медицинских данных от устройства, которым измеряется пациент, в облачное хранилище и предоставляют доступ к данной информации. Например, врач может удалённо контролировать состояние пациента, а также получать доступ к полной истории измерений. Таким образом, он может определить являются ли текущие показатели нормальными для данного пациента. И это всё - почти полностью автоматически. Одной из самых популярных платформ для сбора данных с медицинских устройств является платформа MedM компании Swissmed Mobile¹.

¹medm.com

1. Постановка задачи

Целью данной работы является разработка алгоритма подсчёта пульса и внедрение его в платформу MedM.

Для выполнения этой цели нужно выполнить следующие задачи:

- Разработать мобильное приложение для сбора тестовых данных
- Разработать алгоритм для оценки пульса по собранным тестовым данным
- Реализовать алгоритм для оценки пульса в мобильном приложении
- Протестировать алгоритм непосредственно на мобильных телефонах
- Встроить алгоритм в платформу MedM

2. Обзор

Все существующие алгоритмы для контактного подсчёта пульса с помощью камеры мобильного телефона состоят из двух этапов. Первый - преобразование кадра с камеры (рис. 2) в какую-либо числовую характеристику. Второй - вычисление пульса на основе зависимости данной числовой характеристики от времени.

На первом этапе, все алгоритмы работают примерно одинаково. На кадре выбирается область, на основании которой считается числовая характеристика кадра. Затем выбирается цветовая палитра и канал в ней. Чаще всего используется цветовая палитра RGB и её красный канал. После этого считается среднее значение выбранного канала на заданной области. Таким образом, это среднее значение принимается за числовую характеристику данного кадра.

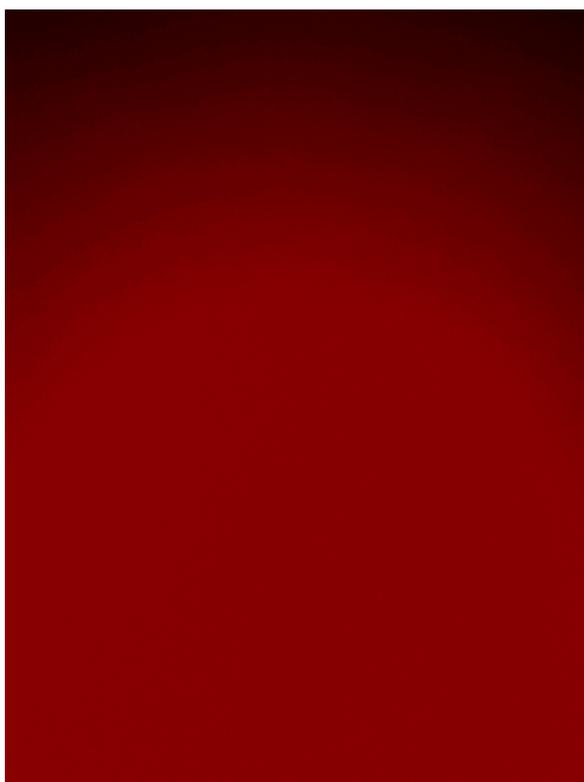


Рис. 2: Пример кадра с камеры с увеличенной контрастностью

В итоге, алгоритм получает сигнал (рис. 3), на котором явно различимы моменты притока крови к пальцу. Задача второго этапа - по этим данным получить пульс. На этом этапе проявляются основные разли-

чия алгоритмов. Можно выделить четыре основных класса алгоритмов:

- Алгоритмы, основанные на подсчёте пиков[1][6]
- Алгоритмы, основанные на подсчёте пиков с дополнительной логикой[3][4][7]
- Алгоритмы, основанные на нахождении доминирующей частоты сигнала
- Алгоритмы, основанные на анализе спектра в плавающем окне[2][5]

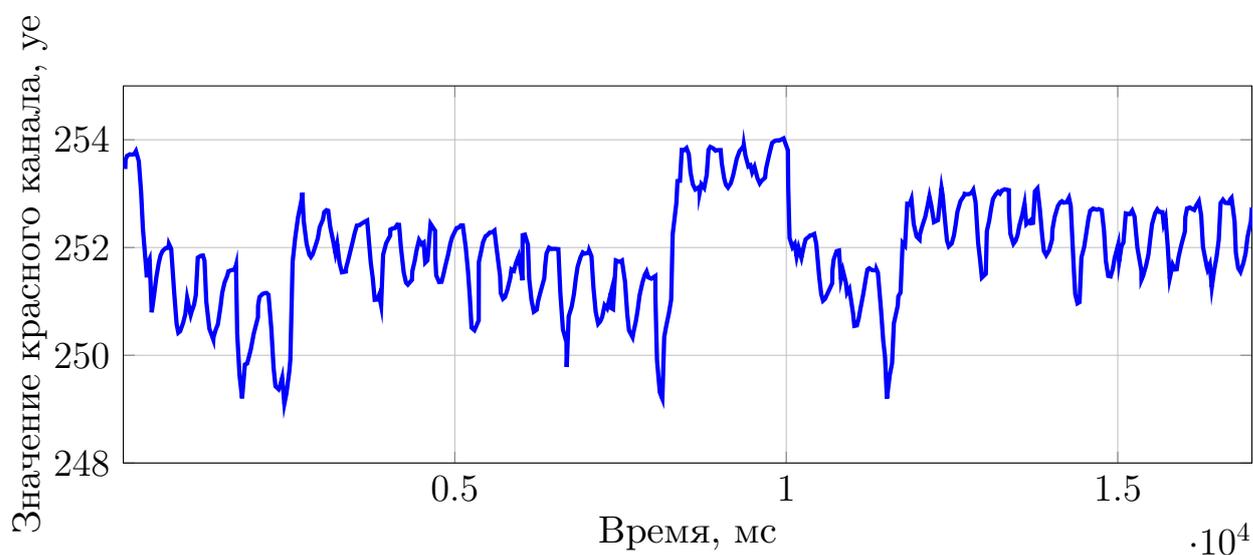


Рис. 3: Пример сигнала длиной 17 секунд после первого этапа

2.1. Алгоритмы, основанные на подсчёте пиков

Данный класс алгоритмов опирается на простой подсчёт пиков, соответствующих каждому притоку крови. Таким образом, поделив количество найденных пиков на время измерения, данный класс алгоритмов получает пульс.

Проблемой данных алгоритмов является неустойчивость к ошибкам в детекции пиков. Ибо пропуск одного или нескольких пиков приводит к значительному отклонению конечного результата. Например, пропуск

двух пиков при пульсе 60 уд/мин при измерении длиной 15 секунд приводит к тому, что алгоритм выдаст результат 52 уд/мин, что значительно отличается от реального пульса.

2.2. Алгоритмы, основанные на подсчёте пиков с дополнительной логикой

Этот класс алгоритмов также опирается на подсчёт пиков, но при этом, эти алгоритмы пытаются исправить недостатки первого класса вводя дополнительную логику поверх подсчёта пиков. Например, при подсчёте пиков дополнительно вычисляется среднее расстояние между ними, и в случае отсутствия пика через данное среднее расстояние, к числу найденных пиков просто добавляется единица. Таким образом, частично решается проблема пропуска пиков.

Несмотря на то, что данные алгоритмы уменьшают спектр проблем, возникающих в первом типе алгоритмов, они никак не влияют на проблемы, появляющиеся непосредственно при детекции пиков. Например, на приведённом сигнале (рис. 4) почти каждый пик, соответствующий притоку крови в палец состоит из двух пиков, которые иногда почти не отличаются между собой, что во многих алгоритмах приведёт к подсчёту лишних пиков.

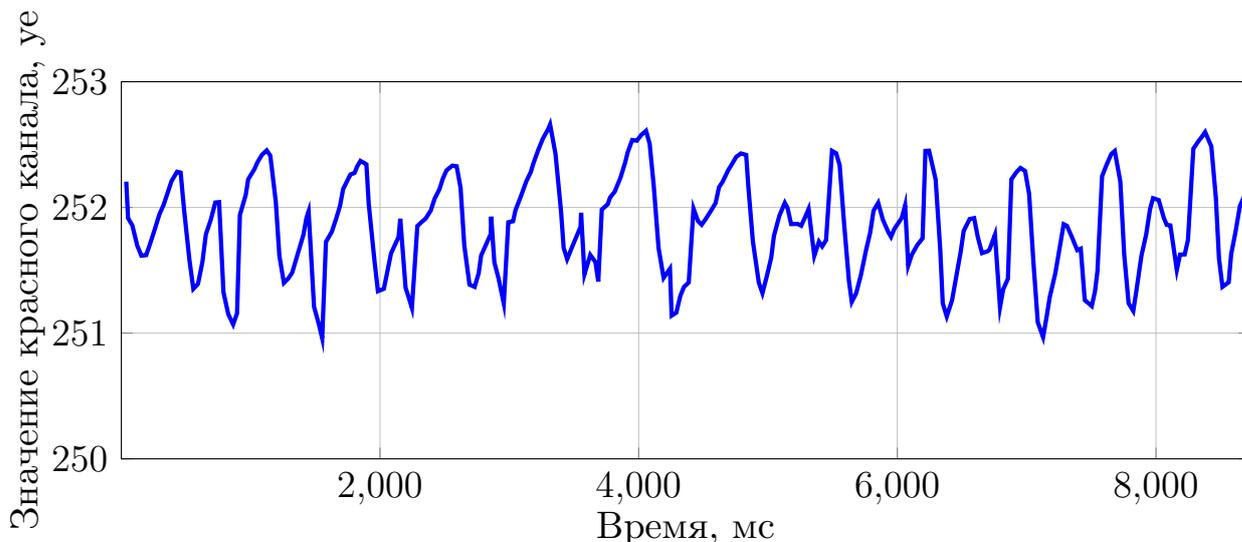


Рис. 4: Пример проблемного сигнала

2.3. Алгоритмы, основанные на нахождении доминирующей частоты сигнала

Данный класс алгоритмов представляет собой абсолютно другой подход к оценке пульса. Вместо нахождения отдельных пиков эти алгоритмы рассматривают сигнал целиком и находят в нём доминирующую частоту. Доминирующая частота дискретного сигнала обычно находится с помощью дискретного преобразования Фурье (ДПФ), которое позволяет получить зависимость амплитуды спектра сигнала от частоты. В общем случае, количество операций для получения спектра с помощью ДПФ равно $O(N^2)$, где N - кол-во отсчетов в исходном сигнале. Однако используя алгоритм быстрого вычисления дискретного преобразования Фурье (БПФ), можно получать спектр дискретного сигнала за $O(N \times \log(N))$ операций, что позволяет применять данный алгоритм в режиме реального времени. После расчета спектра сигнала отбраковываются частоты, не соответствующие возможным значениям пульса человека. Из оставшегося диапазона частот выбирается частота с максимальной амплитудой.

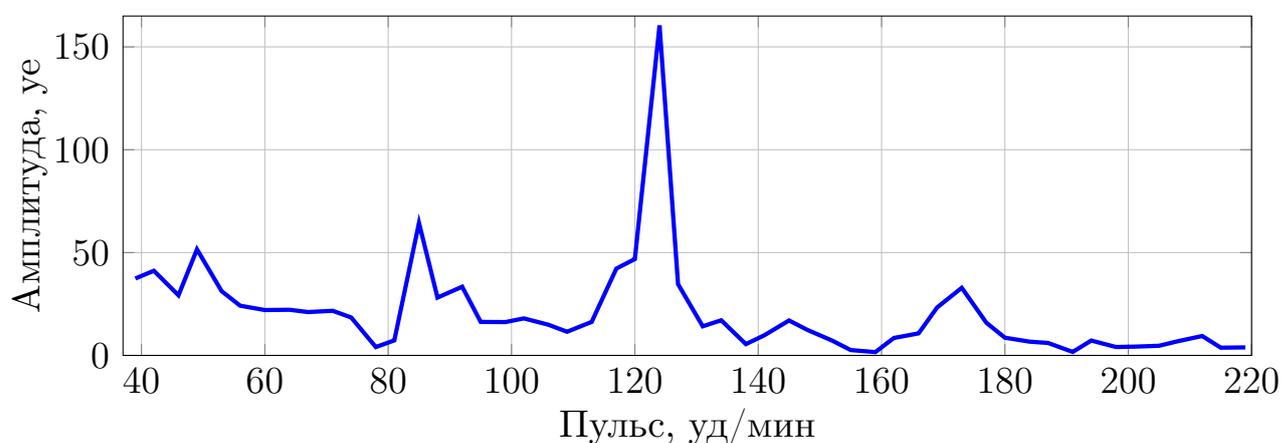


Рис. 5: Пример участка спектра

Соответственно, получив такое разложение для всего измерения достаточно найти частоту с самой большой амплитудой. Конечно же, перед этим нужно выбрать только те частоты, которым может соответствовать реальное значение пульса человека.

Такие алгоритмы проявляют себя гораздо лучше, чем предыдущие

два класса, и позволяют получать достаточно устойчивые результаты при длительной записи сигнала.

Одним из главных ограничений при использовании БПФ является потребность в том, чтобы размер входных данных был степенью двойки и то, что дискретизация амплитудного спектра зависит от частоты и длины входных данных. Шаг дискретизации при измерении пульса вычисляется как $60 \times \frac{\text{частота}}{\text{размер данных}}$. Таким образом, для получения маленькой дискретизации, таким алгоритмам нужен большой размер входных данных. Например, для сигнала с частотой 30 и размером данных 1024 шаг дискретизации будет равен $60 \times \frac{30}{1024} \approx 1.76$. При этом, чтобы получить такую дискретизацию входной сигнал должен иметь длину $\frac{1024}{30} \approx 34$ секунды.

2.4. Алгоритмы, основанные на анализе спектра в плавающем окне

Алгоритмы описанные в пункте 2.3 имеют один существенный недостаток, при небольшой выборке по времени они очень чувствительны к шуму, который имеет высокую амплитуду и расположен в небольшой части сигнала.

На приведённом сигнале(рис. 6) можно заметить, что первые 6 секунд измерения занимает шум, который, к тому же, имеет высокую амплитуду. Это приводит к тому, что итоговый спектр может получить ложный пик на частоте данного шума, который также будет иметь наибольшую амплитуду, что приведёт к неправильному результату.

Поэтому предварительно исходный сигнал разбивается на пересекающиеся отрезки.

Дальнейшие действия зависят от алгоритма. Чаще всего, разделение на пересекающиеся отрезки используется чтобы отбросить части сигнала, которые выбиваются из общего спектра. Например, если доминирующая частота на каком-то отрезке отличается от остальных, то данный отрезок просто отбрасывается.

Можно заметить, что при разбиении на отрезки происходит умень-

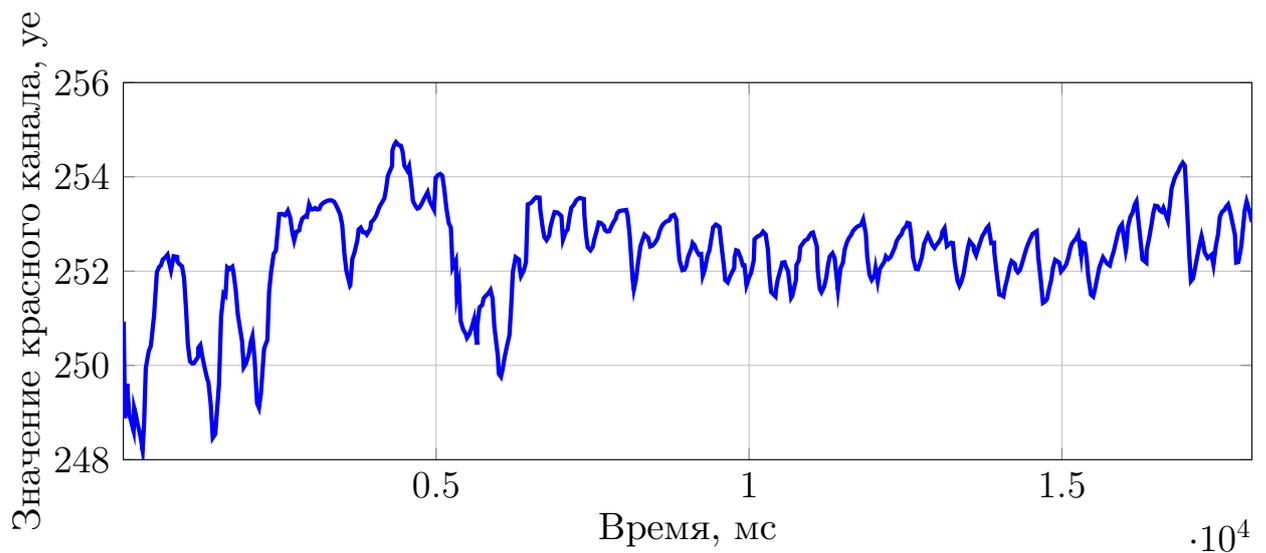


Рис. 6: Пример сигнала с шумом с высокой амплитудой

шение размера входных данных для БПФ. Следовательно, размер шага дескритезации будет увеличиваться. Таким образом, ответ, полученный исключительно за счёт вычисления БПФ для отрезков, будет иметь крайне маленькую точность по сравнению с ответом, полученным после вычисления БПФ для всего измерения.

3. Разработка

3.1. Приложение для сбора тестовых данных

Для разработки алгоритма и его первоначального тестирования необходимо было собрать тестовые данные. Для этого требовалось сделать мобильное приложение которое бы работало на одной из популярных мобильных платформ и умело записывать данные с камеры и каким-либо образом отправлять их нам.

3.1.1. Выбор инструментов

Для реализации приложения была выбрана платформа Android². Выбор Android обоснован широкой популярностью данной платформы среди пользователей мобильных телефонов в России³, а также тем, что я уже имел опыт программирования и работы с камерой на платформе Android.

Несмотря на то, что стандартным языком для написания Android приложений является язык Java, для реализации данного приложения был выбран язык Kotlin компании JetBrains⁴. 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.

3.1.2. Создание приложения

При разработке приложения необходимо было решить проблему, связанную с тем, что в платформе Android по умолчанию обработка кадров с камеры происходит в основном потоке приложения. Это приводит к тому, что при обработке данных камера начинает присылать

²android.com

³statista.com/statistics/262174/market-share-held-by-mobile-operating-systems-in-russia

⁴kotlinlang.org

кадры с камеры гораздо реже.

Например, если камера обычно присылает 25-30 кадров в секунду, то когда мы начинаем как-то обрабатывать каждый кадр в основном потоке приложения (например, считать среднее значение красного канала на кадре), то количество кадров в секунду падает до 10-15.

Для решения такой проблемы в приложении создается отдельный поток, в котором происходит приемка кадров с камеры и их обработка.

Также, стоит отметить, что стандартным форматом для кадров с камеры в платформе Android является YUV420SP, а не RGB. Поэтому, каждый кадр нужно дополнительно переводить в цветовую палитру RGB.

Для отправки тестовых данных с камеры было решено использовать обычную электронную почту. В приложении создаётся запрос на отправку почты с прикрепленными данными на заданный в приложении email.

3.2. Алгоритм

Общая структура алгоритма представлена на рисунке 7.

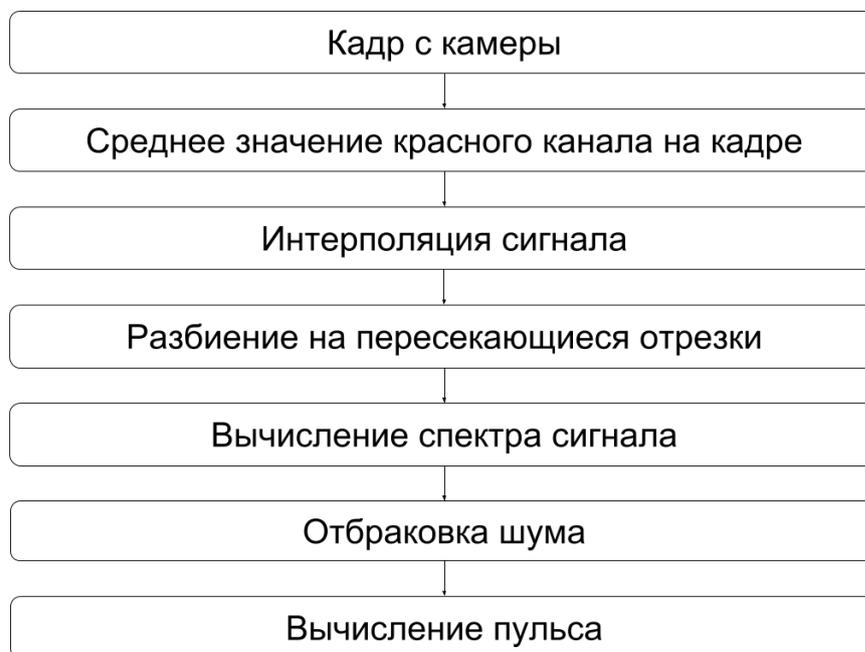


Рис. 7: Структура алгоритма

3.2.1. Получение среднего значения красного канала

Для каждого кадра в качестве его числовой характеристики используется среднее значение красного канала. Среднее значение считается для всего кадра. Оно может изменяться в пределах от 0 до 255.

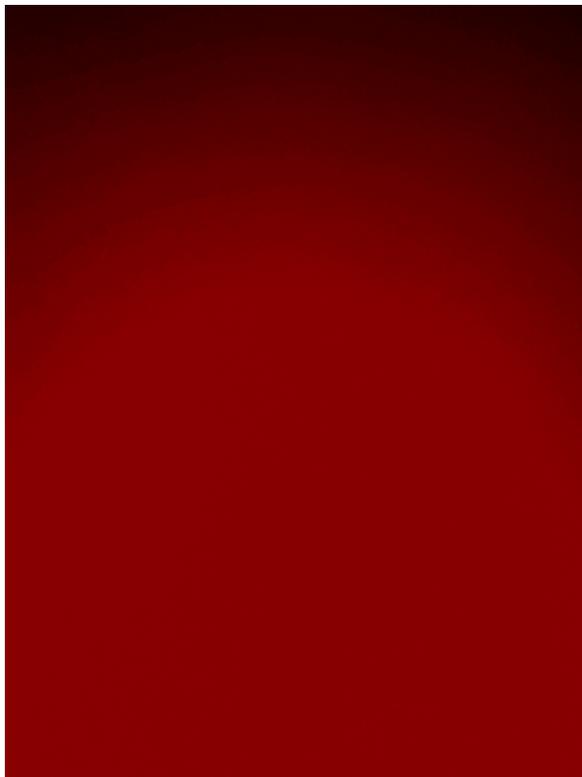


Рис. 8: Кадр с камеры с увеличенной контрастностью

3.2.2. Интерполяция сигнала

Для получения спектра сигнала с помощью БПФ исходный сигнал должен иметь одинаковый равномерный шаг дискретизации по времени. Однако не все мобильные платформы и Android в том числе гарантируют приход кадров с камеры через равные промежутки времени, поэтому перед применением алгоритма БПФ необходима интерполяция исходного сигнала.

Т.к. при реализации алгоритма одним из основных критериев является скорость вычислений, то в качестве интерполяции была выбрана интерполяция полиномом первой степени (линейная интерполяция). Узлы сетки интерполяции выбирались с равномерным шагом равным

средней частоте оцифровки кадров камеры. Обычно такой шаг составлял 30 - 34 мс, что вполне соответствовало необходимой точности измерений.

3.2.3. Разбиение на пересекающиеся отрезки

После интерполяции сигнал разбивается на пересекающиеся отрезки. Длина отрезка составляет четверть от исходной длины измерения. Смещение составляет половину длины отрезка. Таким образом, измерение делится на семь пересекающихся отрезков.

Таким образом, для измерения длиной около 17 секунд, длина отрезка будет составлять ≈ 4.25 секунды. Стоит отметить, что такая длина является достаточной, чтобы предварительно оценить пульс т.к. даже при низком пульсе равном 40 уд/мин в такой отрезок попадёт информация о не менее, чем трёх притоках крови к пальцу, т.е. три периода сигнала.

Как уже отмечалось ранее, разбиение на отрезки приводит к увеличению шага дискретизации амплитудного спектра. В данном случае, для измерения длиной 512 с частотой 30 шаг дискретизации спектра будет равен $60 \times \frac{30}{128} \approx 14$.

Такая точность результата является неприемлимой для конечного результата. Но поскольку, разбиение на отрезки используется исключительно для примерной оценки пульса, то такой точности нам будет достаточно.

3.2.4. Вычисление спектра сигнала

На каждом полученном отрезке с помощью БПФ вычисляется дискретный амплитудный спектр сигнала. Частоты рассчитанного спектра сигнала затем преобразуются в пульс в уд/мин.

3.2.5. Отбраковка шума

После предыдущего этапа мы получаем семь спектров. На данном этапе требуется определить какие из отрезков выбиваются из общей

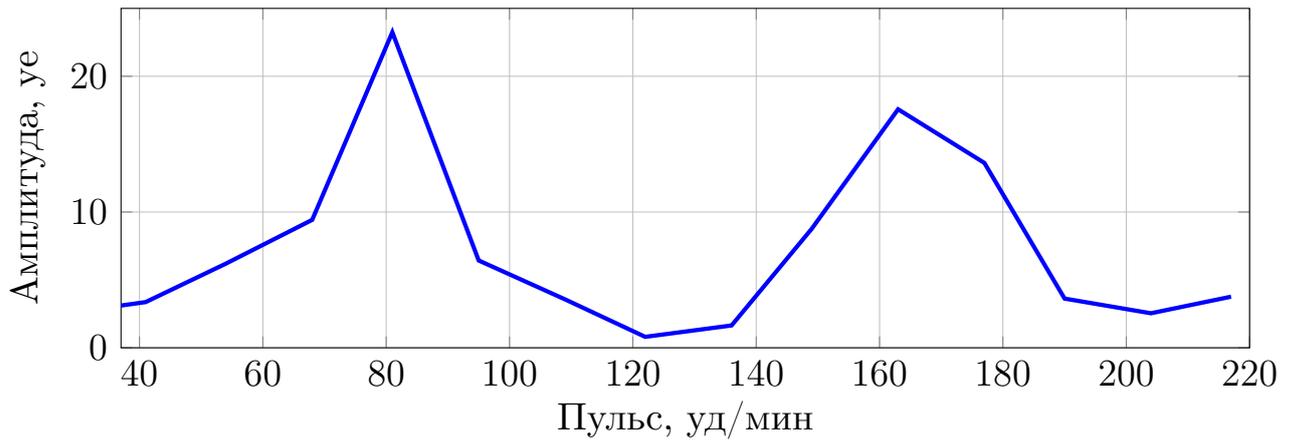


Рис. 9: Участок спектра для одного из отрезков

картины т.е. являются шумом. А также, в случае, если большинство отрезков является мусором, то отбраковать всё измерение.

В каждом спектре находятся два наибольших пика. Если у большинства отрезков совпадает хотя бы один из двух пиков - он принимается за предварительный результат и передаётся на следующий этап. А те отрезки, у которых нет данного пика на спектре - отбраковываются. Если у четырёх из семи отрезков нет общего пика - измерение считается плохим и результат пользователю не выдаётся.

На рисунке 9 показан спектр для хорошего отрезка. Спектр отрезка из того же измерения, который признан шумом изображён на рисунке 10.

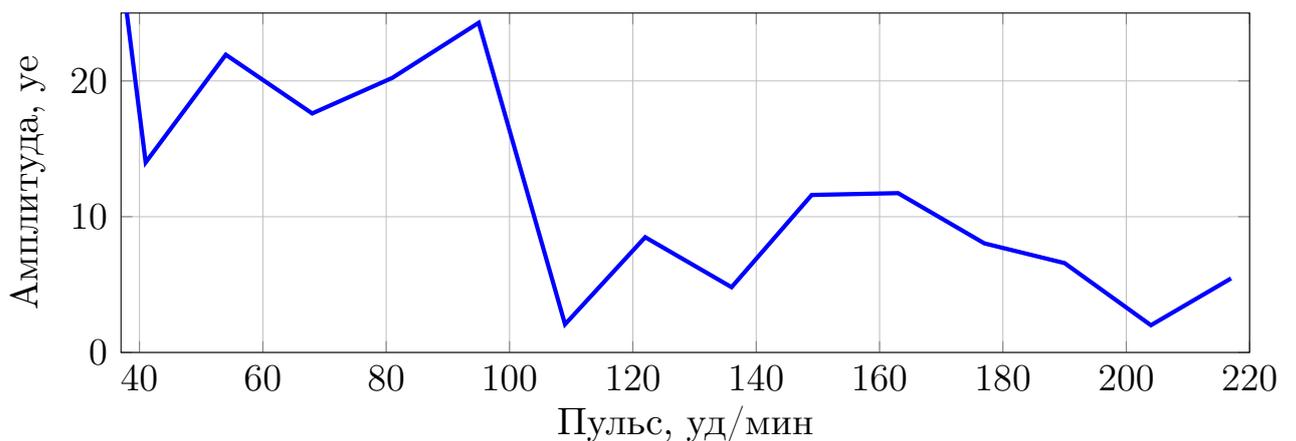


Рис. 10: Участок спектра для отрезка признанного шумом

3.2.6. Вычисление пульса

На вход финальной стадии алгоритма мы получаем общие пики и отрезки, которые не являются шумом. Задача финального этапа - выдать пульс. Сначала с помощью БПФ мы строим спектр для всего измерения. Затем, на спектре всего измерения детектируются пики, которые соответствуют выявленным общим пикам отрезков. Т.е. пики на спектре всего измерения расстояние между вершинами которых и соответствующих вершин из прошлого этапа минимально. Среди найденных таким образом пиков выбирается наибольший по амплитуде.

Если полученный таким образом пик отклоняется от исходного пика больше, чем на половину шага дискретизации спектра для отрезков. (например, для спектров на рисунках 9 и 10 шаг дискретизации равен четырнадцати), то он дополнительно проверяется подсчётом спектра для той части измерения, которая содержит наименьшее количество шума.

Если результаты совпадают - он выдаётся как ответ. Иначе - измерение считается плохим.

На рисунке 11 изображены пики, найденные на спектре всего измерения, которые соответствуют двум пикам со спектра, изображённого на рисунке 9.

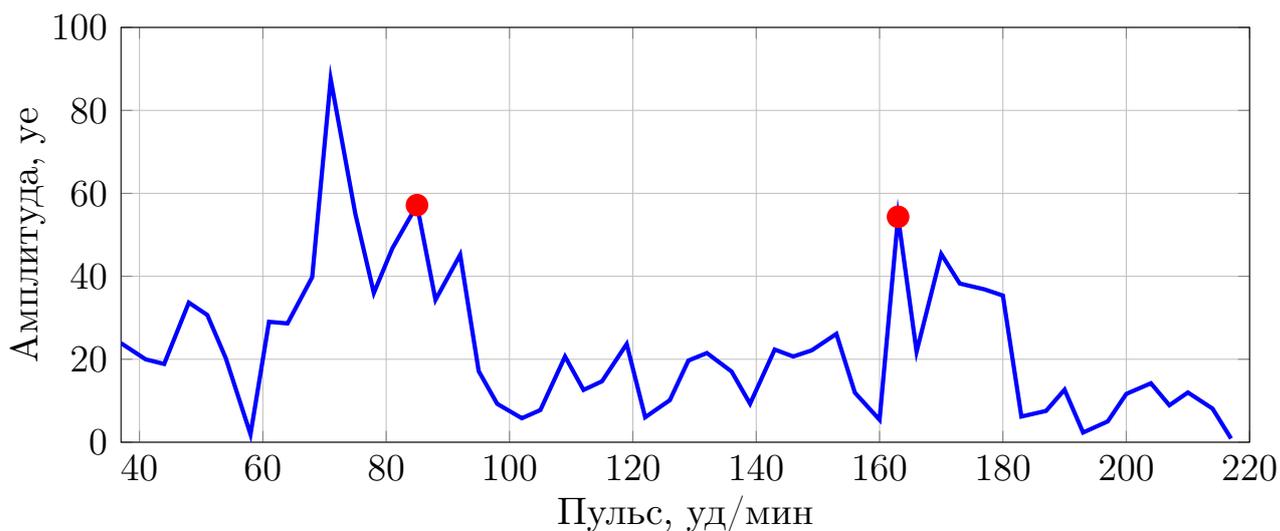


Рис. 11: Участок спектра с отмеченными найденными пиками

3.3. Внедрение алгоритма в приложение

Алгоритм был оформлен как отдельный класс и также, как и приложение, написан на языке Kotlin. После измерения, данные, помимо сохранения, также передаются на вход алгоритму. Результат работы алгоритма отображается на экране.

3.4. Тестирование

Результат работы алгоритма сравнивался с измерениями пульсоксиметра Nonin 3150⁵. 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 освещения с одинаковой частотой (например, движение руки при беге), что вызовет неправильный результат у алгоритма.

В тестировании принимало участие около 20 человек различных возрастных групп.

В качестве итоговой характеристики работы алгоритма выбран процент измерений для которых алгоритм дал отклонение пульса меньше, чем 5 уд/мин. Т.е. алгоритм достаточно точно оценил пульс. Из результатов тестирования можно заметить, что на точность алгоритма не влияет яркий солнечный свет. А также, что точность алгоритма не зависит от размера реального пульса.

3.5. Внедрение алгоритма в платформу MedM

3.5.1. Выбор инструментов

В данном случае, выбор инструментов обусловлен архитектурой платформы MedM. Алгоритм внедрён в платформонезависимую часть. Для этого он был переписан на C++. Это сделано для использования одного алгоритма для подсчёта пульса на различных мобильных платформах.

⁵www.nonin.com/PulseOximetry/Wrist-Worn/WristOx23150

Пульс	% измерений с относительной погрешностью < 5 уд/мин	
	В помещении	На улице
Обычный (<100 уд/мин)	86	85
Повышенный (<140 уд/мин)	85	84
Высокий (>140 уд/мин)	85	83

Таблица 1: Результаты тестирования алгоритма

Работа с камерой на платформе Android внедрена в платформозависимую часть. Но, поскольку платформозависимая часть в платформе MedM реализована на языке Java, работу с камерой пришлось переписать с Kotlin на Java.

3.5.2. Внедрение

Платформа MedM изначально была создана для работы с внешними медицинскими устройствами, поэтому моему алгоритму пришлось эмулировать такое устройство.

Работа с внешним устройством состоит из четырёх этапов:

- Обнаружение устройства
- Запоминание устройства
- Соединение с устройством
- Получение данных с устройства

Общая схема взаимодействия пользователя платформы MedM с моим алгоритмом, эмулирующим внешнее устройство, представлена на рисунке 12.

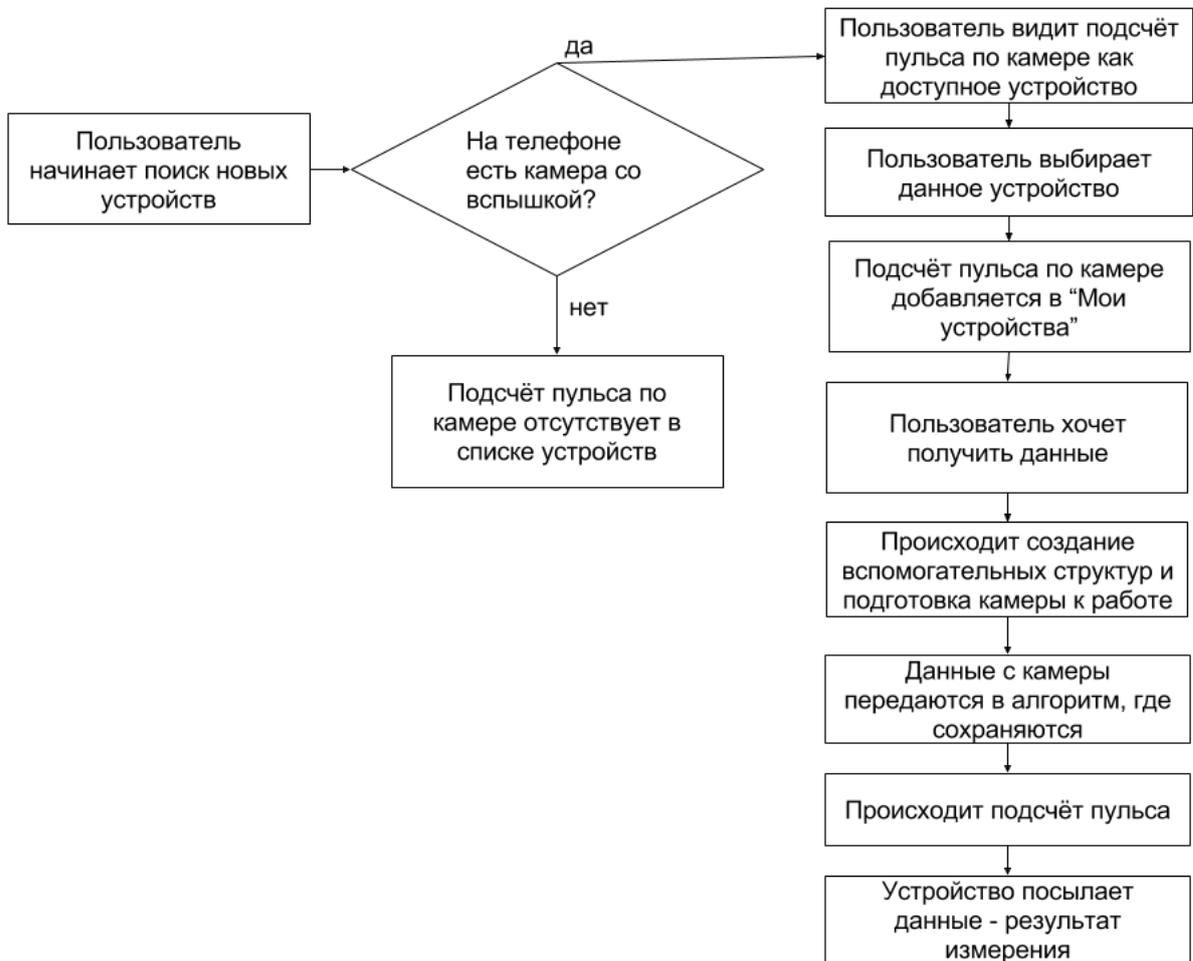


Рис. 12: Схема взаимодействия пользователя платформы MedM с алгоритмом

4. Результаты

В рамках данной курсовой работы были получены следующие результаты:

- Разработано мобильное приложение на платформе Android для сбора тестовых данных
- Разработан алгоритм для оценки пульса по тестовым данным
- В мобильном приложении реализован алгоритм для оценки пульса
- Алгоритм протестирован на мобильных телефонах
- Алгоритм встроен в платформу MedM

Список литературы

- [1] Fan Xiangmin, Wang Jingtao. BayesHeart: A Probabilistic Approach for Robust, LowLatency Heart Rate Monitoring on Camera Phones. — 2015.
- [2] Improved heart rate detection using smart phone / Arpan Pal, Aishwarya Visvanathan, Anirban Dutta Choudhury, Aniruddha Sinha. — 2014.
- [3] Laure Denis, Paramonov Ilya. Improved Algorithm for Heart Rate Measurement Using Mobile Phone Camera. — 2013.
- [4] A Novel Method to Detect Heart Beat Rate Using a Mobile Phone / Pelegris P., Banitsas K., Orbach T., Marias K. — 2010.
- [5] A Novel Method to Detect Heart Beat Rate Using a Mobile Phone / Arpan Pal, Aniruddha Sinha, Anirban Dutta Choudhury et al. — 2013.
- [6] 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.
- [7] Vikram Chandrasekaran B.E. Measuring Vital Sings Using Smart Phones. — 2010.