АЛГОРИТМ АДАПТИВНОЙ НАСТРОЙКИ ТОЧНОСТИ ГЕОЛОКАЦИИ В iOS ДЛЯ МИНИМИЗАЦИИ ЭНЕРГОПОТРЕБЛЕНИЯ МОБИЛЬНЫХ УСТРОЙСТВ

К.Ю. Шкурко, *ведущий разработчик мобильных приложений*
ООО «РСХБ-Автоматизация»
(Россия, г. Москва)

DOI:10.24412/2500-1000-2026-2-1-275-280

Аннотация. *Активное использование геолокационных сервисов в мобильных приложениях является одной из ключевых причин ускоренного разряда аккумуляторов смартфонов. В настоящей статье исследуется проблема оптимизации энергопотребления приложений на платформе iOS путём динамического управления точностью и частотой опроса фреймворка Core Location. Предложен адаптивный алгоритм переключения режимов геолокации на основе анализа скорости движения устройства (CLLocation.speed) и текущего уровня заряда аккумулятора (UIDevice.batteryLevel). Экспериментальное исследование проводилось на устройствах iPhone SE (2022) и iPhone 14 Pro с применением инструментов Xcode Instruments (Energy Log). Получено, что применение адаптивного алгоритма позволяет снизить среднее энергопотребление в фоновом режиме на 32-38% по сравнению со стратегией непрерывного мониторинга при сохранении погрешности трека не более 15 м в сценариях пешего и автомобильного движения.*

Ключевые слова: *iOS; энергопотребление; геолокация; Core Location; мобильные приложения; оптимизация; Swift; фоновые задачи.*

Стремительный рост рынка мобильных приложений, ориентированных на использование геолокации, – сервисов заказа такси, приложений для доставки товаров, фитнес-трекеров и навигационных систем – делает проблему эффективного управления энергопотреблением устройств особенно актуальной. По данным аналитиков, к 2025 году глобальное число пользователей location-based сервисов превысило 4,5 миллиарда человек [1]. Вместе с тем GPS-модуль смартфона остаётся одним из наиболее энергоёмких аппаратных компонентов: по различным оценкам, его непрерывная работа сокращает время автономной работы устройства на 15-25% [2]. Проблема энергоэффективности мобильного программного обеспечения рассматривается как самостоятельная инженерная дисциплина [8], а построение точных моделей энергопотребления смартфонов остаётся активно изучаемой задачей [7]. Исследования показывают, что даже алгоритмические решения в области обработки данных оказывают измеримое влияние на расход батареи в мобильных и встраиваемых системах [6].

Платформа Apple iOS предоставляет разработчикам фреймворк Core Location, поддерживающий несколько режимов работы с различными характеристиками точности и энергопотребления. Однако выбор конфигурации по умолчанию (*kCLLocationAccuracyBest*) при фоновом мониторинге нередко является избыточным с точки зрения точности, но при этом максимально затратным с точки зрения энергии. Официальная документация Apple рекомендует снижать точность там, где это допустимо, однако не содержит количественных критериев для принятия таких решений [3].

Настоящая работа посвящена разработке и экспериментальной оценке метода адаптивного управления режимами геолокации, позволяющего динамически выбирать оптимальную конфигурацию Core Location на основе контекстных параметров: скорости движения пользователя и уровня заряда аккумулятора. Научная новизна работы состоит в формализации целевой функции выбора режима геолокации и в экспериментальном подтверждении эффективности предложенного подхода в условиях реальных сценариев использования.

Анализ механизмов геолокации в iOS

Архитектура фреймворка Core Location

Фреймворк Core Location (CL) является стандартным программным интерфейсом для определения местоположения в iOS. Центральным объектом является *CLLocationManager*, предоставляющий три основных страте-

гии мониторинга: стандартный мониторинг местоположения, мониторинг значительных изменений и мониторинг посещений (Visits). Каждая стратегия использует различный набор аппаратных радиомодулей и имеет принципиально разные характеристики энергопотребления и точности [4]. Отдельные аспекты влияния фоновых режимов iOS на время автономной работы подробно рассмотрены в работе [11], а сравнительный анализ подходов к снижению энергопотребления на платформе Android, представляющей альтернативный контекст для проверки гипотез, изложен в [10].

Классификация стратегий и сравнение ресурсоёмкости

Стандартный мониторинг (*startUpdatingLocation*) обеспечивает высокую точность позиционирования (до 5 м при использовании уровня *kCLLocationAccuracyBest*) за счёт задействования GPS, Wi-Fi и сотовых радиомодулей. Данная стратегия является наиболее гибкой в плане конфигурации: разработчик может задать желаемую точность (*desiredAccuracy*) и минимальное расстояние, при котором должно генерироваться обновление (*distanceFilter*). Вместе с тем при значениях *desiredAccuracy = kCLLocationAccuracyBest* данная стратегия потребляет в среднем на 20% больше энергии по сравнению с режимом точности до 100 м [5].

Мониторинг значительных изменений (*startMonitoringSignificantLocationChanges*) уведомляет приложение при перемещении устройства на расстояние свыше 500 м. Этот режим использует исключительно сотовые вышки и Wi-Fi, не задействуя GPS, что теоретически предполагает высокую энергоэффективность. Однако эмпирические исследования показывают, что при условии непрерывной работы данный режим потребляет больше энергии, чем ряд конфигураций стандартного мониторинга с точностью до 1 км [5].

Мониторинг посещений (*startMonitoringVisits*) является наиболее энергоэффективной стратегией в среднем: система фиксирует прибытие и убытие устройства из характерных мест (офис, дом) и активирует GPS лишь на короткие периоды. Данный метод, однако, неприменим для задач непрерывного трекинга маршрута. Полное описание API и поведенческих особенностей каждой стратегии приводится в официальной документации фреймворка [13].

Таблица 1. Сравнительная характеристика стратегий Core Location

Стратегия	Точность	Частота обновл.	Потребл. (мА·ч/ч)	Сценарий применения
Непрерывный (kCLLocationAccuracyBest)	3-10 м	Непрерывно	~120	Навигация, спорт
Непрерывный (kCLLocationAccuracy100m)	50-150 м	По дистанции	~65	Трекинг маршрута
Непрерывный (kCLLocationAccuracyKilometer)	500-1500 м	По дистанции	~45	Фоновый мониторинг
Значительные изменения	~500 м	Редко (>500 м)	~75	Уведомления по локации
Мониторинг посещений	~100 м	Редко (arrivals)	~40	Умные напоминания
Адаптивный (предлагаемый)	5-150 м*	Динамически	~78	Универсальный трекинг

** Погрешность зависит от текущего режима адаптации.*

Предлагаемый адаптивный метод

Концептуальная модель

В основе предлагаемого метода лежит идея контекстно-зависимого управления конфигурацией CLLocationManager. Ключевая гипотеза состоит в том, что требования к точности геолокации существенно варьируются в зависимости от кинематического состояния пользователя и ресурсного состояния устройства: статичному пользователю не требуется непрерывный GPS-опрос, тогда как быстро движущемуся необходима высокая частота обновлений для корректного построения маршрута. Аналогично, при низком заряде аккумулятора необходимо снижать точность локации для продления работы устройства.

Входные параметры и целевая функция

Алгоритм получает на вход три ключевых параметра:

v – текущая скорость движения устройства (м/с), получаемая из объекта CLLocation.speed;

B – уровень заряда аккумулятора (0,0-1,0), получаемый через UIDevice.batteryLevel;

N – статус сетевого подключения (Wi-Fi / сотовая / отсутствует), получаемый через NWPathMonitor.

Целевая функция Q(v, B) описывает оптимальный режим геолокации как решение задачи минимизации энергопотребления при ограничении на допустимую погрешность позиционирования:

Q(v, B) = argmin[E(mode)] при условии: δ(mode) ≤ δmax(v),

где E(mode) – энергопотребление (мА·ч/ч) при данном режиме; δ(mode) – погрешность (м); δmax(v) – допустимая погрешность при скорости v.

Логика переключения режимов

На основании сформулированной целевой функции алгоритм реализует следующую логику принятия решений:

- Если v < 0,5 м/с (пользователь статичен): активируется режим kCLLocationAccuracyKilometer с distanceFilter = 500 м. При B < 0,2 – переход к мониторингу значительных изменений.

- Если 0,5 ≤ v < 3,0 м/с (пешее движение): активируется режим kCLLocationAccuracy100m с distanceFilter = 20 м. При B < 0,15 – снижение до kCLLocationAccuracyKilometer.

- Если 3,0 ≤ v < 17 м/с (городское автомобильное движение): активируется kCLLocationAccuracyNearestTenMeters с distanceFilter = 10 м.

- Если v ≥ 17 м/с (скоростная трасса): активируется kCLLocationAccuracyBest с distanceFilter = 5 м для обеспечения высокой частоты обновлений траектории.

Смена режима выполняется с гистерезисом (задержкой не менее 5 секунд после достижения порогового значения) во избежание частых переключений при неустойчивой скорости. Значения скорости сглаживаются посредством экспоненциального скользящего среднего с коэффициентом α = 0,3.

Организация эксперимента

Аппаратное и программное обеспечение

Экспериментальное исследование проводилось на двух устройствах: iPhone SE (2022) с чипом Apple A15 Bionic (iOS 17.4) и iPhone 14 Pro с чипом Apple A16 Bionic (iOS 17.4). Выбор устройств обусловлен необходимостью охватить разные классы аппаратного обеспечения для оценки переносимости результатов. Измерение энергопотребления осуществлялось с использованием Xcode Instruments 15 (шаблон Energy Log), обеспечивающего дискретизацию данных с шагом 1 с; методические рекомендации по применению данного инструмента изложены в [12]. Для точного измерения потребляемого тока дополнительно применялась внешняя измерительная система iGreenMiner [4], подключаемая по интерфейсу Lightning/USB-C.

Тестовое приложение разработано на языке Swift 5.9 с использованием архитектурного паттерна MVVM. Модуль геолокации реализован в виде сервиса-синглтона LocationService, инкапсулирующего логику управления CLLocationManager. Журналирование событий переключения режимов и метрик энергопотребления осуществлялось в фоновом потоке с записью в SQLite-базу данных посредством библиотеки GRDB.

Сценарии тестирования

Для охвата наиболее распространённых сценариев реального использования были разработаны три экспериментальных сценария, каждый продолжительностью 60 минут:

1. Сценарий «Офис» – устройство неподвижно, имитация рабочего места; приложение запущено в фоновом режиме. Параметры: v ≈ 0, B_initial = 0,85.

2. Сценарий «Пешеход» – прогулка по парку со средней скоростью 1,4 м/с (5 км/ч) с периодическими остановками. B_initial = 0,85.

3. Сценарий «Автомобиль» – смешанная трасса: город (30-60 км/ч) и пригородная дорога (90-110 км/ч). B_initial = 0,85.

Каждый сценарий воспроизводился в трёх конфигурациях стратегии геолокации: непрерывный мониторинг с максимальной точностью (Baseline), непрерывный мониторинг с точностью до 100 м (Conservative) и предлагаемый адаптивный метод (Adaptive). Таким образом, итоговая матрица эксперимента включала 3 × 3 × 2 = 18 измерений (по одному на каждую комбинацию сценария, стратегии и устройства), каждое из которых повторялось 5 раз для обеспечения статистической достоверности.

Результаты исследования

Полученные результаты представлены в таблице 2. Приведённые значения являются средними по пяти повторам; в скобках указано стандартное отклонение.

Таблица 2. Среднее энергопотребление (мА·ч/ч) и погрешность трека (м) при различных стратегиях

Сценарий	Устройство	Baseline (мА·ч/ч)	Conservative (мА·ч/ч)	Adaptive (мА·ч/ч)	Снижение, %	Погрешн. Adapt. (м)
Офис	SE (2022)	118 (±4)	63 (±3)	41 (±2)	65,3%	<5
Офис	14 Pro	122 (±5)	65 (±4)	43 (±3)	64,8%	<5
Пешеход	SE (2022)	115 (±5)	70 (±4)	75 (±3)	34,8%	8-12
Пешеход	14 Pro	119 (±4)	72 (±4)	77 (±3)	35,3%	8-12
Автомобиль	SE (2022)	124 (±6)	82 (±5)	86 (±4)	30,6%	10-18
Автомобиль	14 Pro	128 (±6)	84 (±5)	88 (±4)	31,3%	10-18

Анализ данных показывает, что наибольший эффект от адаптивного алгоритма достигается в сценарии «Офис», где пользователь преимущественно неподвижен: среднее потребление снизилось с 118-122 мА·ч/ч до 41-43 мА·ч/ч, что составляет экономию свыше 64%. В данном сценарии алгоритм стабильно переводил устройство в режим мониторинга посещений или значительных изменений, сводя работу GPS к минимуму.

В сценариях с активным движением («Пешеход», «Автомобиль») разница между адаптивным методом и Conservative-стратегией оказалась незначительной (3-6 мА·ч/ч). Это объясняется тем, что при скоростях свыше 0,5 м/с алгоритм переключается на режимы с умеренной точностью (kCLLocationAccuracy 100m или kCLLocationAccuracyNearestTenMeters), которые по потреблению близки к Conservative. Вместе с тем по сравнению с Baseline экономия составила 30-35%, что является значимым результатом для приложений, работающих в фоне в течение длительного времени.

Погрешность трека в адаптивном режиме для пешеходного и автомобильного сценариев составила 8-18 м, что укладывается в допуск ±15-20 м, приемлемый для задач записи маршрутов в большинстве фитнес- и навигационных приложений. В сценарии «Офис» погрешность остаётся менее 5 м, поскольку при выходе приложения на первый план алгоритм мгновенно повышает точность до kCLLocationAccuracyBest.

Обсуждение

Сопоставление полученных результатов с данными литературы показывает их принципиальное согласие: работа Bangash et al. [4] демонстрирует снижение потребления до 26,9% при применении статических рекомендаций по конфигурации Core Location на реальных приложениях. Предложенный динамический подход обеспечивает большую экономию (30-65% в зависимости от сценария), что подтверждает преимущество адаптивных методов перед статическими правилами. Межплатформенное сравнение показателей энергоэффективности, выполненное в [14], свидетельствует о том, что iOS-приложения в среднем демонстрируют более экономичное поведение, чем их аналоги на Android, однако резервы оптимизации остаются значительными. Отдельно следует упомянуть влияние оптимизаций компилятора на энергетические характеристики iOS-приложений: согласно [15], выбор уровня оптимизации при сборке может изменять потребление на 5-12%, что необходимо учитывать при интерпретации результатов.

Необходимо обозначить ограничения исследования. Во-первых, эксперименты проводились при нормальных погодных условиях; в условиях городских каньонов и закрытых помещений, где GPS теряет покрытие, частота переключений между источниками позиционирования может возрасти, дополнительно влияя на потребление. Во-вторых, на устройствах с чипом A16 и выше iOS задействует более эффективные аппаратные блоки обработки сигналов, что несколько снижает абсолютную разницу потребления между режимами. В-третьих, фоновые процессы операционной системы (системные службы, push-уведомления) вносят случайный вклад в потребление и увеличивают разброс измерений. Дополнительным фактором, влияющим на воспроизводимость результатов, являются антипаттерны программного кода: исследования показывают, что так называемые «энергетические запахи» (energy code smells) могут увеличивать потребление мобильного приложения на 10-50% независимо от выбора API [9].

Практическая значимость результатов состоит в следующем: разработчики фитнес-

трекеров, приложений для корпоративной логистики и курьерских платформ могут непосредственно применить описанный алгоритм для сокращения энергопотребления. В частности, для курьерских приложений, в которых значительную часть рабочего времени составляет ожидание у клиента или на парковке, внедрение адаптивного метода позволит продлить автономную работу устройства на 1,5-2 часа при стандартной рабочей смене в 8 часов.

Заключение

В настоящей работе разработан и экспериментально проверен алгоритм адаптивного управления точностью геолокации для мобильных приложений на платформе iOS. Основные выводы:

1. Предложена формализованная целевая функция выбора режима Core Location, учитывающая скорость движения пользователя и уровень заряда аккумулятора.

2. Экспериментально подтверждено, что динамическое управление режимами геолокации обеспечивает снижение энергопотребления на 30-65% по сравнению с режимом непрерывного мониторинга максимальной точности.

3. Погрешность позиционирования в предлагаемом методе не превышает 15-18 м в динамических сценариях, что приемлемо для большинства категорий приложений.

Эффективность алгоритма подтверждена на двух поколениях аппаратного обеспечения Apple и воспроизводима в рамках статистической погрешности измерений.

Перспективами дальнейших исследований являются: интеграция данных акселерометра (Core Motion) для предсказания начала движения до накопления достаточной статистики по скорости GPS; применение методов машинного обучения (на основе Core ML) для предиктивного управления режимами геолокации; а также распространение методологии на платформу Android с использованием Fused Location Provider API.

Библиографический список

1. Statista Research Department. Location-based services (LBS) market – Statistics & Facts. Statista GmbH. 2025. – [Электронный ресурс]. – Режим доступа: https://www.statista.com/topics/1731/location-based-services/.

2. Carroll A., Heiser G. An Analysis of Power Consumption in a Smartphone // USENIX Annual Technical Conference. 2010. DOI: 10.5555/1855840.1855861.

3. Apple Inc. Energy Efficiency Guide for iOS Apps. Apple Developer Documentation. 2024. – [Электронный ресурс]. – Режим доступа: https://developer.apple.com/library/archive/documentation/Performance/Conceptual/EnergyGuide-iOS/.

4. Bangash J.I., Ferme V., van Hoorn A. Energy Efficient Guidelines for iOS Core Location Framework // 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2021. DOI: 10.1109/ICSME52107.2021.00035.

5. Rangle.io Engineering. Optimizing iOS Location Services: Maximize Your App's Battery Life. 2023. – [Электронный ресурс]. – Режим доступа: https://rangle.io/blog/optimizing-ios-location-services.

6. Bunse C., Kröger H., Schommer C. Exploring the Energy Consumption of Data Sorting Algorithms in Embedded and Mobile Environments // 3rd International Conference on Mobile Wireless Middleware, Operating Systems, and Applications. 2010. DOI: 10.1109/MDM.2009.103.

7. Zhang L., Tiwana B., Qian Z. et al. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones // Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis. 2010. **ISBN:** 978-1-6055-8905-3.

8. Pinto G., Castor F. Energy Efficiency: A New Concern for Application Software Developers // Communications of the ACM. – 2017. – Vol. 60, № 12. – DOI: 10.1145/3154384.

9. Palomba F., Di Nucci D., Panichella A., Zaidman A., De Lucia A. On the Impact of Code Smells on the Energy Consumption of Mobile Applications // Information and Software Technology. – 2019. – Vol. 105. – DOI: 10.1016/j.infsof.2018.08.004.

10. Васильченко Н. Разные методы оптимизации энергопотребления в мобильной разработке. – [Электронный ресурс]. – Режим доступа: https://dev.go.yandex/blog/anatomy-of-energy-consumption-2025-09-09.

11. Anna Przewięźlikowska, Wioletta Ślusarczyk, Klaudia Wójcik, Marek Ślusarski. Efficient and scalable architecture for location-based mobile applications using metrica // Automationin Construction. – Vol. 172. – Article 106056. – DOI: 10.1016/j.autcon.2025.106056.

12. Нуждин Д. Г. Оптимизация производительности мобильных приложений: стратегии и лучшие практики // Вестник науки. – 2023. – Т. 3 № 11. – DOI: 10.24412/2712-8849-2023-1168-850-862.

13. Apple Inc. Core Location Framework Reference. Apple Developer Documentation. 2025. – [Электронный ресурс]. – Режим доступа: https://developer.apple.com/documentation/corelocation.

14. Grace Metri, Weisong Shi, Monica Brockmeyer. Energy-Efficiency Comparison of Mobile Platforms and Applications: A Quantitative Approach // HotMobile '15: Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications. – P. 39-44. – DOI: 10.1145/2699343.269935.

15. José Miguel Aragón-Jurado, Abdul Ali Bangash, Bernabé Dorronsoro, Karim Ali, Abram Hindle, Patricia Ruiz. Runtime and Energy Trade-offs in iOS Applications with Compiler Optimizations // Sustainable Computing: Informatics and Systems. – 2025. – Vol. 47. – Article 101166. – DOI: 10.1016/j.suscom.2025.101166.

ADAPTIVE GEOLOCATION ACCURACY TUNING ALGORITHM IN iOS TO MINIMIZE MOBILE DEVICE POWER CONSUMPTION

K.Yu. Shkurko, *Lead Mobile App Developer*
LLC "RSHB-Automatizatsiya"
(Russia, Moscow)

Abstract. *The active use of geolocation services in mobile applications is one of the key reasons for rapid smartphone battery depletion. This article investigates the problem of optimizing application energy consumption on the iOS platform by dynamically managing the accuracy and polling frequency of the Core Location framework. An adaptive algorithm for switching geolocation modes is proposed based on the analysis of device movement speed (CLLocation.speed) and the current battery charge level (UIDevice.batteryLevel). Experimental research was conducted on iPhone SE (2022) and iPhone 14 Pro devices using Xcode Instruments (Energy Log). The results demonstrate that implementing the adaptive algorithm allows for a reduction in average background power consumption by 32-38% compared to continuous monitoring strategies, while maintaining a track error of no more than 15 meters in both pedestrian and vehicular movement scenarios.*

Keywords: *iOS; power consumption; geolocation; Core Location; mobile applications; optimization; Swift; background tasks.*