

Определение местоположения по сигналам акселерометра*

Зайнулина Э. Т., Киселёва Е. А., Фатеев Д. А., Божedomов Н.,
Толканев А. А., Ночевкин В., Протасов В., Рябов А.

Аннотация: данная статья посвящена методам отслеживания местоположения человека по сигналам акселерометра, гироскопа, магнитометра. Основной задачей исследования является увеличение точности позиционирования в условиях, когда глобальная навигационная система не может быть использована. В качестве базовой модели был выбран PDR (pedestrian dead reckoning). Для уменьшения зашумленности данных был использован фильтр Калмана. Новизна исследования заключается в постановке задачи в терминах Projection to Latent Spaces.

Ключевые слова: *Pedestrian dead reckoning, (Indoor) inertial positioning, Simultaneous Localization and Mapping, PLS.*

1 Введение

В настоящее время системы по определению местоположения человека стали неотъемлемой частью повседневной жизни. Информация о точном местоположении человека используется для обеспечения безопасности, для “мобильного здоровья”, для эффективной организации рабочих процессов, для мониторинга толпы и др. Огромную роль в определении местоположения человека играет GNSS (глобальная навигационная система). Однако в помещении навигационные спутниковые сигналы не всегда доступны, из-за чего качество данных, предоставляемых GNSS, сильно уменьшается. Тем не менее большую часть времени человек проводит в помещениях, в связи с чем должны быть разработаны надежные, точные методы, позволяющие определять местоположение человека в помещении.

Современные смартфоны обладают большим числом сенсоров и высокой вычислительной способностью. Так как в настоящее время почти каждый человек им обладает, то методы определения местоположения человека с использованием смартфонов получили наибольшее внимание со стороны исследователей. Среди этих методов - методы, основанные на беспроводных сигналах (WiFi, Bluetooth, UWB) [12] [7], датчиках обзора (лазерный сканер, монокулярная и бинокулярная камера) [1], инерционных датчиках (акселерометр, гироскоп, магнитометр) [9] [4] [13] [6]. Многие из предложенных методов локализации человека представляют собой комбинацию выше перечисленных для увеличения точности позиционирования [3] [2] [8]. Методы, основанные на беспроводных сигналах и датчиках обзора, помимо наличия смартфона требуют также введения дополнительного оборудования либо наличия дополнительных знаний, например карты помещения или базы данных силы сигнала (RSSI) WiFi точки в зависимости от координаты (WiFi fingerprint). Однако не всегда возможно предоставить карту помещения, например, в силу конфиденциальности; вспомогательное оборудование, в свою очередь, требует технического обслуживания и больших затрат. Что касается WiFi позиционирования, то при наличии существующей базы данных WiFi fingerprint при некотором изменении среды, позиционирование будет неточным, поэтому база данных нуждается в постоянном обновлении [11].

Чтобы избежать данных проблем, предлагается метод, основанный на инерционных датчиках. В качестве базового алгоритма рассматривается pedestrian dead reckoning

* Научный руководитель: Стризов В.В. Консультант: Мотренко А.

(PDR) [5]. По сравнению с методами, основанными на беспроводных сигналах и датчиках обзора, PDR рассчитывает относительно точное местоположение человека быстрее и потребляя меньше вычислительной мощности. Для фильтрации шума в данных используется фильтр Калмана [14]. Особенность данной работы состоит в том, чтобы восстанавливать траекторию не от точки к точке, а всю целиком. Для работы с полученным многомерным пространством предлагается использовать метод PLS [10].

2 Постановка задачи

Данные, полученные с помощью инерционных датчиков, представляют собой многомерные временные ряды $s(t) \in \mathbb{R}^N$. Каждому временному ряду сопоставляется вектор признаков. Эти вектора образуют матрицу признаков $X \in \mathbb{R}^{N \times T}$. По матрице X необходимо предсказать траекторию пешехода $Y \in \mathbb{R}^{2 \times T}$, где строками матрицы Y являются временные ряды $y(t)$, описывающие изменения глобальных координат (широта и долгота соответственно) пешехода во времени. Искомая модель:

$$f : X \rightarrow Y$$

В силу зависимости получаемых данных от расположения смартфона данная задача разбивается на 2 подзадачи. Первая подзадача состоит в том, чтобы определить класс P расположения датчиков: нога, рука, сумка, тело ($\{0, 1, 2, 3\}$). Вторая подзадача состоит уже в предсказании траектории на основе полученного класса:

$$f \rightarrow f_1 f_2$$

$$f_1 : X \rightarrow P = \{0, 1, 2, 3\}$$

$$f_2 : X, P \rightarrow Y$$

Для решения данных задач используется метод опорных векторов, который минимизирует следующий функционал $S(w|f, X, Y)$:

$$\min_{w, w_0} S(w, w_0) = \|w\|^2 + C \sum_i \xi_i$$

$$\text{subject to } y_i(w^T x_i + w_0) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \forall i$$

Для оценки качества модели используется критерий суммы квадратов отклонений предсказанных координат от истинных, а также корреляция между предсказанной и истинной траекториями пешехода.

Формально постановку задачи можно записать следующим образом:

$$w^* = \arg \min_w S(w|f, X, Y).$$

3 Базовый алгоритм

Для получения матрицы признаков X предварительно из данных удаляются высокочастотные шумы с помощью сглаживания Гаусса. Полученные линейные и угловые скорости уже преобразуются в вектор признаков.

В качестве базового алгоритма используется каскадная регрессия, состоящая из SVM-классификатора и 8 SVR-регрессоров [14].

По матрице признаков X SVM-классификатор определяет, какому расположению датчиков (в руке, на ноге, на поясе, в сумке) соответствует данное описание, т.е. решает задачу многоклассовой классификации с 4 классами.

После этого для каждого класса на тренировочной выборке обучается свой SVR-регрессор, который выдает скорости движения пешехода для каждого временного блока. Полученные скорости содержат ошибку, связанную с неточностями инерционных датчиков, поэтому далее для этих скоростей находится смещение (по предположению низкочастотное) из следующей задачи минимизации:

$$\min_{\{x_I^1, x^5 1_I, \dots\}} V_{bias} = \min_{\{x_I^1, x^5 1_I, \dots\}} \sum_{f \in F_2} \|v_C^f - v_R^f\| + \lambda \sum_{f \in F_1} \|x_I^f\|^2,$$

$$v_C^f = R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'} (a_I^{f'} + x_I^{f'}),$$

где f - единица блока выборки, F - блок выборки, v_C^f - скорректированное значение скорости, v_R^f - предсказанное значение скорости, I - система координат устройства, W - глобальная система координат, S - IMU-стабилизированная система координат, R_{AB} - матрица перехода из системы координат B в систему координат A .

Также для каждого класса создается SVR-регрессор, предсказывающий угловые скорости пешехода в каждом временном блоке.

На контрольной выборке для SVM-классификатора и каждого SVR-регрессора подбираются оптимальные значения гиперпараметров.

По полученным значениям скоростей восстанавливается траектория пешехода.

Формально алгоритм описывается следующим образом:

После получения из данных матрицы признаков X

Вход: $X, Y_{class}, Y, X_{test}$

- 1: *initialize classifier_options*
- 2: *classifier = SVMClassifier(classifier_options);*
- 3: *classifier.fit(X, Y_class)*
- 4: **для** *cls in classes*:
- 5: *initialize regressor_cls_options*
- 6: *regressor_cls = SVRRRegressor(regressor_cls_options)*
- 7: *regressor_cls.fit(X[X[ind] ∈ cls], Y[Y[ind] ∈ cls])*
- 8: *Y_test-class = classifier.predict(X_test)*
- 9: **для** *cls in classes*:
- 10: *Velocity_cls = regressor_cls.predict(X_test[Velocity_class[ind] == cls])*
- 11: $x_I^1, x^5 1_I, \dots = \arg \min_{\{x_I^1, x^5 1_I, \dots\}} V_{bias_cls}$
- 12: *Velocity_cls = $R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'} (a_I^{f'} + x_I^{f'})$*
- 13: *Trajectory_cls recovery depending on Velocity_cls*
- 14: **return** *Full_trajectory*

4 Базовый эксперимент

Цель эксперимента: найти параметры моделей такие, что предсказанная траектория пешехода будет наиболее близкой к истинной.

Для эксперимента используется часть данных, собранных в статье, описывающей алгоритм RIDI [14]. Эти данные были получены с помощью инерционных датчиков, расположенных в смартфонах в нескольких случаях: когда смартфон располагался на поясе, в руке, на ноге или в сумке. Выборки содержат различные траектории длиной в 100 минут и частотой сигнала 200 Гц. В качестве объекта рассматривается положение в определенный момент времени i . Признаками объекта являются угловые скорости и линейные ускорения в стабилизированной системе координат датчиков в моменты времени $i - window_size, \dots, i$, где $window_size$ - размер окна (равен 200). Целевыми переменными являются метки классов, характеризующие то, в каком положении находился смартфон при получении определенных данных, а также скорости в данный момент времени i , которые вычисляются через координаты пешехода и прошедшее время. По полученным данным после уточнения скоростей с помощью оптимизации функции V_{bias} строится предсказанная траектория пешехода.

В ходе эксперимента исследовалась зависимость качества моделей на контрольной выборке в зависимости от параметров SVM-регрессоров. Во всех моделях в качестве ядер были выбраны радиальные базисные функции, подбирались такие параметры как коэффициент штрафа C и ядерный коэффициент γ . Качество измерялось с помощью кросс-валидации. Из результатов эксперимента следует, что для каждого расположения смартфона и каждого канала данных должны быть выбраны свои параметры модели. Это подтверждает разумность классификации типа расположения смартфона перед непосредственным предсказанием траектории.

Графики зависимости качества предсказания модели от параметров:

1. Выборка 1 состоит из 30742 объектов (8728 объектов класса рука, 6106 объектов класса нога, 7758 объектов класса тело, 8150 объектов класса сумка).
2. Выборка 2 состоит из 42731 объектов (13204 объектов класса рука, 8083 объектов класса нога, 11105 объектов класса тело, 10339 объектов класса сумка).
3. Выборка 3 состоит из 35892 объектов (9458 объектов класса рука, 7304 объектов класса нога, 13306 объектов класса тело, 5824 объектов класса сумка).

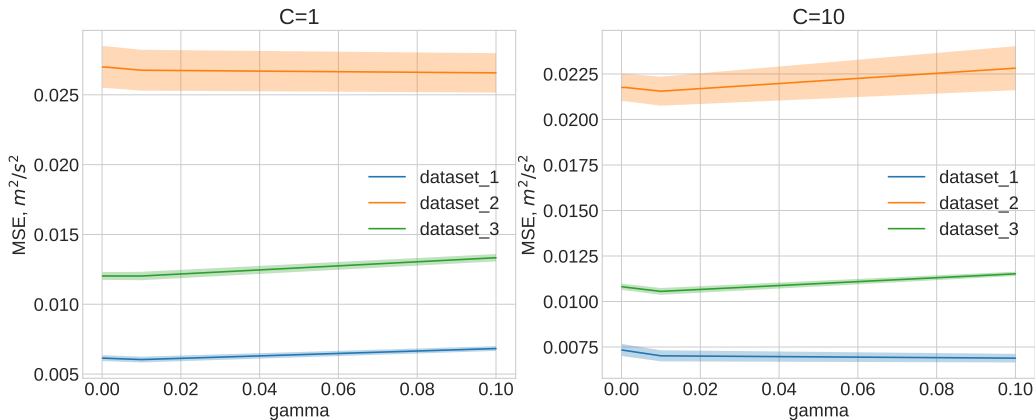


Рис. 1 Рука, канал 0

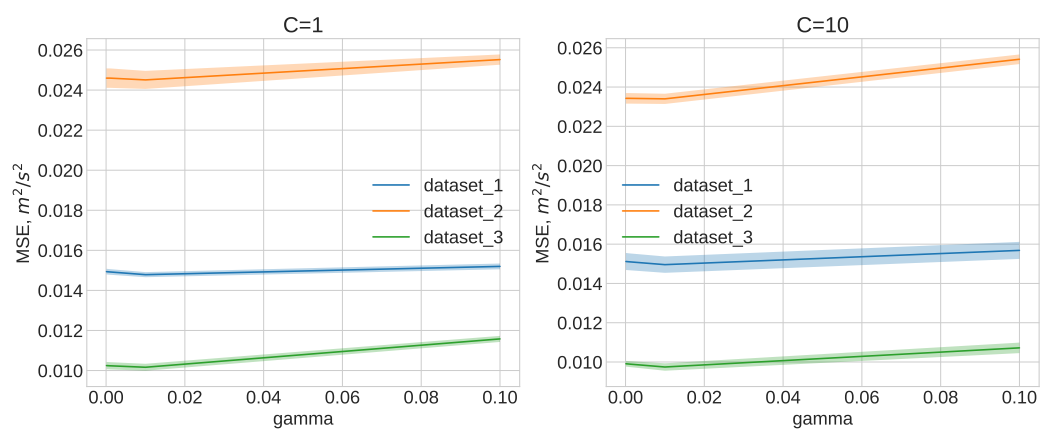


Рис. 2 Рука, канал 1

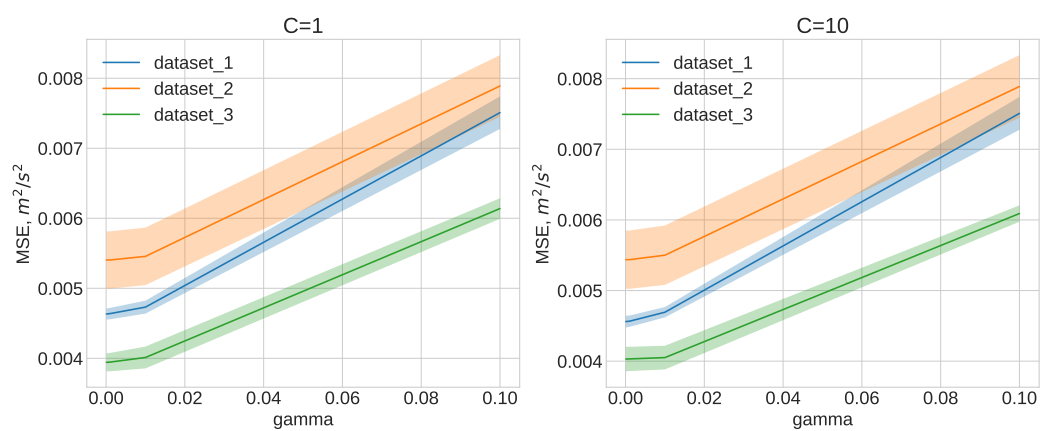


Рис. 3 Нога, канал 0

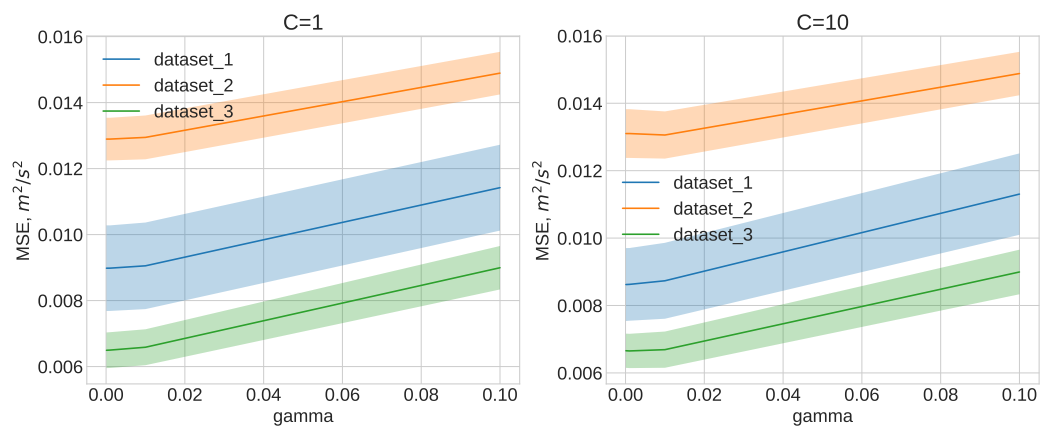


Рис. 4 Нога, канал 1

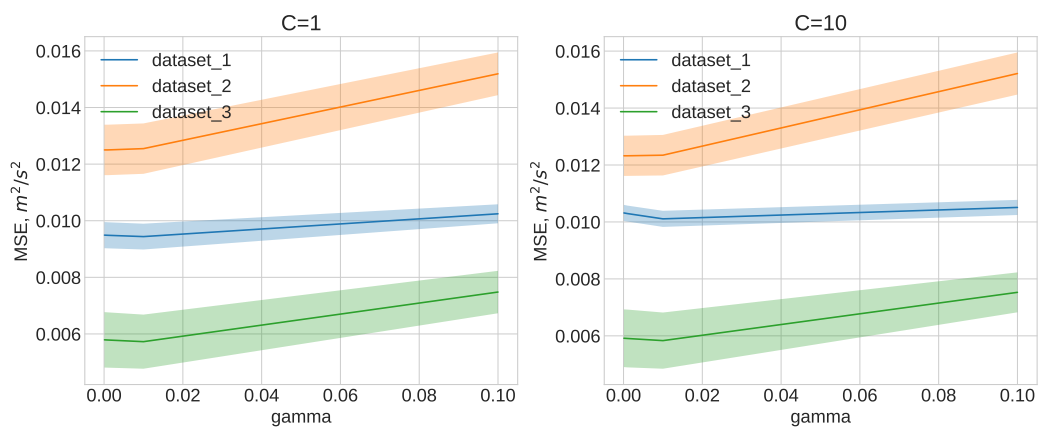


Рис. 5 Сумка, канал 0

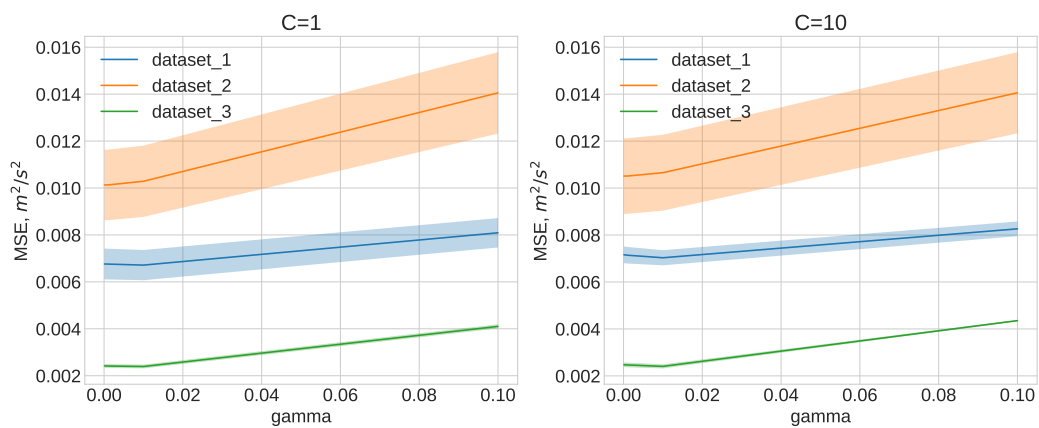


Рис. 6 Сумка, канал 1

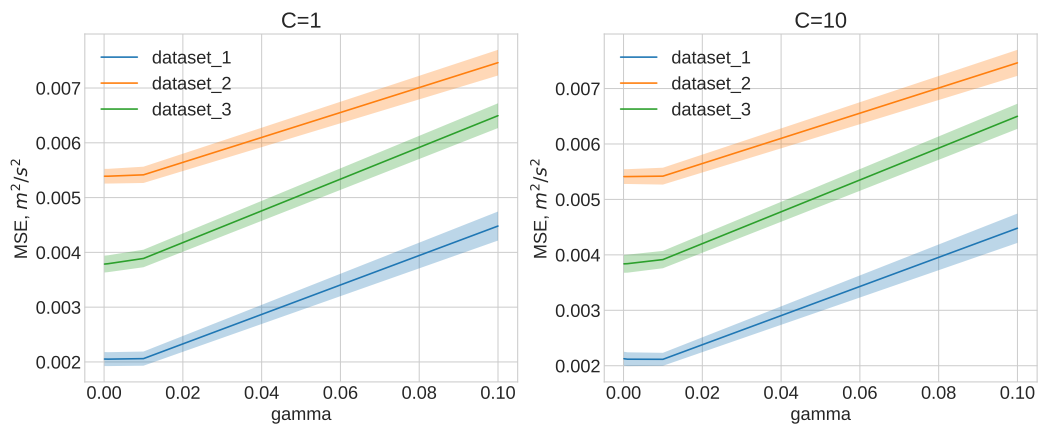


Рис. 7 Тело, канал 0

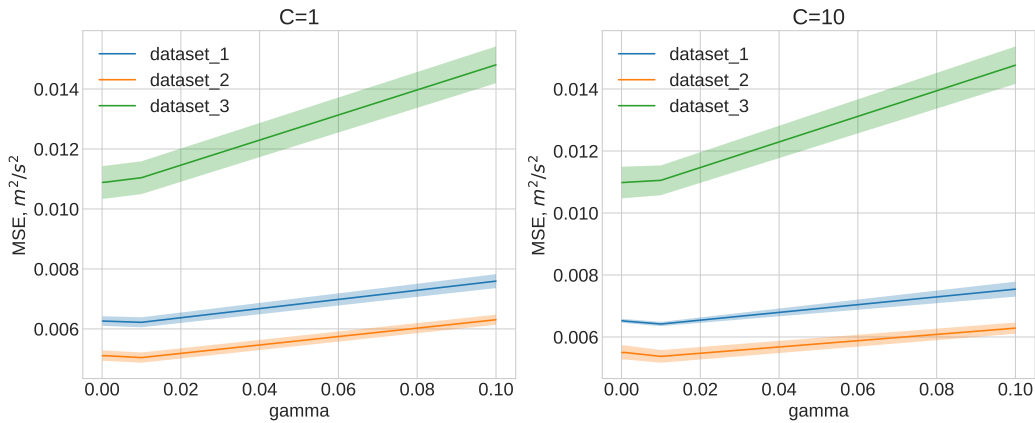
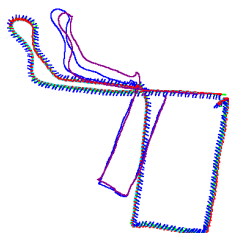


Рис. 8 Тело, канал 1

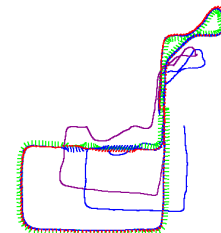
Для всех классов и выборок оптимальные значения параметра γ близки к 0.001, 0.01, поэтому при дальнейшем обучении моделей на большом количестве данных при заранее не заданных параметрах SVM-регрессоров, при поиске по сетке для параметра γ будут использоваться только эти значения. Тогда для построенных моделей оптимальными параметрами будут следующие:

	Рука	Нога	Сумка	Тело
C	10	1	1	1
γ	0.01	0.001	0.01	0.001

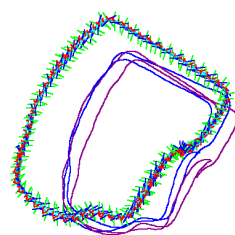
По полученным значениям ошибок на кросс-валидации были выбраны оптимальные модели. С помощью этих моделей были построены траектории для каждого класса расположения смартфона (в качестве тестовой выборки была использована выборка Zhicheng). При этом траектории были построены для случаев, когда дополнительная корректировка весов с помощью оптимизации V_{bias} не производилась (сиреневая линия) и когда производилась (синяя линия). Истинная траектория обозначена красным цветом.



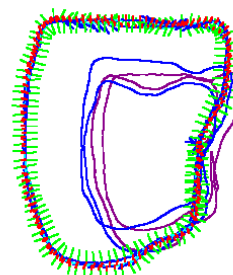
Класс-сумка



Класс-рука



Класс-нога



Класс-тело

Рис. 10 Траектории

5 Выводы

Изначальное определение класса расположения смартфона помогло установить более подходящие параметры моделей, что увеличило точность предсказания. Также, как видно из рисунков полученных траекторий, дополнительное уточнение скоростей дает лучшее приближение истинных траекторий. В ходе данной работы были повторены результаты статьи RIDI [14]. Также в этой статье при формировании выборки для уменьшения высокочастотного шума на изначальных данных предлагается применять фильтр Гаусса. В дальнейшем планируется применить полученную модель для дополнительно собранных данных, а также улучшить методы обработки данных для уменьшения шума (применение фильтра Калмана) и посмотреть другие способы оптимизации модели.

6 Приложения

Таблица 1 Зависимости $MSE (m^2/s^2)$ от параметров моделей для выборки 1

Регрессор	C=1				C=10			
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$
Сумка, 0	0.00949	0.00948	0.00944	0.01025	0.01032	0.01029	0.01011	0.01051
Сумка, 1	0.00676	0.00676	0.00671	0.00809	0.00716	0.00714	0.00703	0.00826
Тело, 0	0.00205	0.00205	0.00206	0.00448	0.00213	0.00212	0.00212	0.00448
Тело, 1	0.00626	0.00626	0.00622	0.00759	0.00652	0.00651	0.00642	0.00754
Рука, 0	0.00614	0.00613	0.00604	0.00683	0.00734	0.00731	0.00702	0.00689
Рука, 1	0.01494	0.01492	0.01479	0.0152	0.01512	0.0151	0.01496	0.01568
Нога, 0	0.00463	0.00464	0.00473	0.00751	0.00456	0.00457	0.00469	0.00751
Нога, 1	0.00898	0.00898	0.00905	0.01142	0.00862	0.00863	0.00873	0.01131

Таблица 2 Зависимости $MSE (m^2/s^2)$ от параметров моделей для выборки 2

Регрессор	C=1				C=10			
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$
Сумка, 0	0.0125	0.0125	0.01255	0.01519	0.01232	0.01232	0.01234	0.01521
Сумка, 1	0.01013	0.01013	0.01029	0.01406	0.01051	0.01051	0.01065	0.01406
Тело, 0	0.00205	0.00205	0.00206	0.00448	0.00213	0.00212	0.00212	0.00448
Тело, 1	0.00511	0.00511	0.00504	0.00631	0.0055	0.0055	0.00537	0.00629
Рука, 0	0.02699	0.02699	0.02676	0.02657	0.02176	0.02176	0.02155	0.02282
Рука, 1	0.0246	0.0246	0.02451	0.02552	0.02342	0.02342	0.0234	0.02541
Нога, 0	0.0054	0.0054	0.00546	0.00789	0.00544	0.00544	0.0055	0.00789
Нога, 1	0.01289	0.01289	0.01295	0.01489	0.0131	0.0131	0.01306	0.01488

Таблица 3 Зависимости MSE (m^2/s^2) от параметров моделей для выборки 3

Регрессор	C=1				C=10			
	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$	$\gamma = 0.0001$	$\gamma = 0.001$	$\gamma = 0.01$	$\gamma = 0.1$
Сумка, 0	0.00579	0.00579	0.00573	0.00748	0.00592	0.00591	0.00583	0.00753
Сумка, 1	0.00242	0.00241	0.00239	0.0041	0.00247	0.00247	0.00241	0.00435
Тело, 0	0.00379	0.00379	0.00389	0.0065	0.00384	0.00384	0.00391	0.0065
Тело, 1	0.00511	0.00511	0.00504	0.00631	0.0055	0.0055	0.00537	0.00629
Рука, 0	0.02699	0.02699	0.02676	0.02657	0.02176	0.02176	0.02155	0.02282
Рука, 1	0.01025	0.01024	0.01016	0.01158	0.00991	0.00989	0.00975	0.01072
Нога, 0	0.00394	0.00395	0.00401	0.00614	0.00403	0.00403	0.00405	0.00609
Нога, 1	0.00649	0.0065	0.00659	0.009	0.00666	0.00665	0.00669	0.00899

Литература

- [1] Erich Bruns and Oliver Bimber. Adaptive training of video sets for image recognition on mobile phones. *Personal and Ubiquitous Computing*, 13(2):165–178, 2009.
- [2] L. Chen, E. H. Wu, M. Jin, and G. Chen. Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation. *IEEE Sensors Journal*, 14(11):4034–4042, Nov 2014.
- [3] Frédéric Evennou and François Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *EURASIP J. Adv. Sig. Proc.*, 2006, 2006.
- [4] Michael Hardegger, Daniel Roggen, and Gerhard Tröster. 3d actionslam: wearable person tracking in multi-floor environments. *Personal and Ubiquitous Computing*, 19(1):123–141, 2015.
- [5] R. Hostettler and S. Särkkä. Imu and magnetometer modeling for smartphone-based pdr. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.
- [6] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, May 2015.
- [7] Soo-Cheol Kim, Young-Sik Jeong, and Sang Oh Park. Rfid-based indoor location tracking to ensure the safety of the elderly in smart home environments. *Personal and Ubiquitous Computing*, 17(8):1699–1707, 2013.
- [8] M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, April 2015.
- [9] Kwanghyo Park, Hyojeong Shin, and Hojung Cha. Smartphone-based pedestrian tracking in indoor corridor environments. *Personal and Ubiquitous Computing*, 17(2):359–370, 2013.
- [10] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In Craig Saunders, Marko Grobelnik, Steve Gunn, and John Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection*, pages 34–51, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [11] Joaquín Torres-Sospedra and Adriano J. C. Moreira. Analysis of sources of large positioning errors in deterministic fingerprinting. *Sensors*, 17(12):2736, 2017.
- [12] Rodrigo Vera, Sergio F. Ochoa, and Roberto G. Aldunate. Edips: an easy to deploy indoor positioning system to support loosely coupled mobile work. *Personal and Ubiquitous Computing*, 15(4):365–376, 2011.
- [13] Boyuan Wang, Xuelin Liu, Baoguo Yu, Ruicai Jia, and Xingli Gan. Pedestrian dead reckoning based on motion mode recognition using a smartphone. *Sensors*, 18(6):1811, 2018.
- [14] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. *CoRR*, abs/1712.09004, 2017.