

# Определение местоположения по сигналам акселерометра

Зайнулина Э. Т., Киселёва Е. А., Фатеев Д. А., Божедомов Н., Протасов В.

Abstract—Системы внутреннего и наружного позиционирования играют важную роль в современном мире. Для определения места положения используются Глобальные системы позиционирования (GPS); однако, не всегда есть возможность воспользоваться ими.

В данной статье предложен метод увеличения точности отслеживания человека по сигналам акселерометра, гироскопа и магнитометра.

Чтобы увеличить точность и уменьшить время вычислений, мы используем модель PDR (pedestrian dead reckoning). Так как данные неизбежно собираются с некоторым шумом, применяем Гауссовское сглаживание. Задача исследования ставится в терминах Projection to Latent Spaces, т.к. восстановление траектории происходит целиком, а не от точки к точке.

Index Terms—PDR, PLS, Position system, User location.

## I. ВВЕДЕНИЕ

Определение местонахождения человека чрезвычайно важная задача. Мы пользуемся системами GNSS, чтобы не потеряться в городе или добраться до соседнего, обеспечить безопасность в толпе, координировать действия рабочих. Но GNSS определяет местоположение в здании неточно.

Методы основанные на сигналах WiFi, Bluetooth, UWB [7][12] требуют дополнительной информации: карты помещения или набора ключевых точек. Это не всегда удобно или даже реализуемо.

С другой стороны сейчас у каждого человека в кармане целый арсенал средств для ориентирования в пространстве. Все спрятано в наших смартфонах: акселерометр, гироскоп, магнитометр. В нашей работе мы предлагаем восстанавливать полную траекторию по сигналам этих датчиков, используя модель PDR, избавляясь от зашумленности с помощью Гауссовского сглаживания.

## II. ПОСТАНОВКА ЗАДАЧИ

### A. Модель

$$f : X \rightarrow Y$$

$X \in R^{N \times T}$  - матрица признаков,

$Y \in R^{2 \times T}$  - траектория пешехода

$y(t)$  - строки матрицы  $Y$ , временные ряды, описывающие изменения глобальных координат пешехода во времени.

Научный руководитель: Стрижов В. В. Консультант: Мотенко А.

### B. Подзадачи

- 1) Определение класса местоположения датчика: рука, нога, сумка, тело.
- 2) Предсказании траектории на основе полученного класса

$$f \rightarrow f_1 f_2$$

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

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

### C. Методы

Используем метод опорных векторов для классификации и регрессии.

Минимизируем  $S(w|f, X, Y)$ :

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

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

$$\xi_i \geq 0 \quad \forall i$$

$S(w, w_0)$  - штраф за суммарную ошибку.

### D. Оценка качества модели

Используем критерий суммы квадратов отклонений предсказанных координат от истинных, корреляция между предсказанной и истинной траекториями пешехода.

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

## III. БАЗОВАЯ МОДЕЛЬ

- Чтобы устранить зашумленность данных применяем Гауссовское сглаживание для 6 каналов гиросtabilизатора и 2 скоростных каналов. Полученные скорости преобразуем в вектор признаков.
- Строим матрицу признаков  $X$ .
- С помощью матрицы  $X$  и SVM-классификатора определяем, где находится датчик. (Рука, нога, сумка, тело).
- Для каждого класса обучаем 2 SVMR-регрессора. Первый определяет скорости движения человека в разные промежутки времени и угловые скорости для канала  $X$ . Но эти результаты имеют ошибку из-за неточности датчиков гироскопа, акселерометра. Решаем задачу минимизации этой ошибки.

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

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

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  $A$ .

Второй предсказывает то же для 2 канала.

- По полученным скоростям восстанавливаем траекторию пешехода.
- Алгоритм:

Require:  $X, Y_{class}, Y, X_{test}$

```

1: initialize classifier_options
2: classifier = SVMClassifier(classifier_options);
3: classifier.fit(X, Y_class)
4: for cls in classes: do
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: end for
9: Y_test-class = classifier.predict(X_test)
10: for cls in classes: do
11:   Velocity_cls = regressor_cls.predict(X_test[Velocity_class[ind]
   cls])
12:    $x_I^1, x_I^5, \dots = \{x_I^1, x_I^5, \dots\} V_{bias\_cls}$ 
13:   Velocity_cls =  $R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'}(a_I^{f'} + x_I^{f'})$ 
14:   Trajectory_cls recovery depending on Velocity_cls
15: end for
16: return Full_trajectory
```

#### IV. БАЗОВЫЙ ЭСПЕРИМЕНТ

Для эксперимента используется часть данных, собранных в статье, описывающей алгоритм RIDI [7]. 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  $i - window\_size$ ,  $Window\_size$  - 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 строится предсказанная траектория пешехода.

А. 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).

см. Fig. 1- 8

В. Оптимальные параметры эксперимента

см. Table I

С. Результаты

см. Fig.9-10

#### V. ВЫВОДЫ

Повторив эксперимент статьи [7] мы убедились, что классификация нахождения датчика (рука, нога, сумка, тело) повышает точность предсказания траектории. Убедились, что для устранения шума лучше использовать фильтр Гаусса.

#### ACKNOWLEDGMENT

##### References

- [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] Frederic Evennou and Francois 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 Troster. 3d actionslam: wearable person tracking in multi-floor environments. Personal and Ubiquitous Computing, 19(1):123–141, 2015.
- [5] R. Hostettler and S. Sarkka. 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] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. CoRR, abs/1712.09004, 2017.

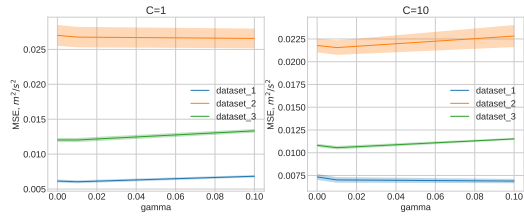


Fig. 1. Рука, канал 0

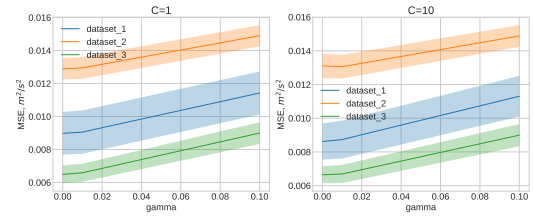


Fig. 4. Нога, канал 1

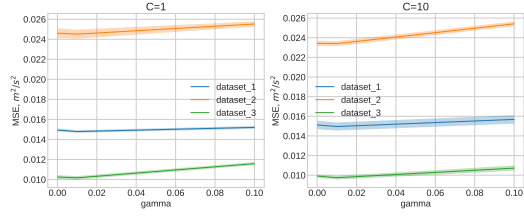


Fig. 2. Рука, канал 1

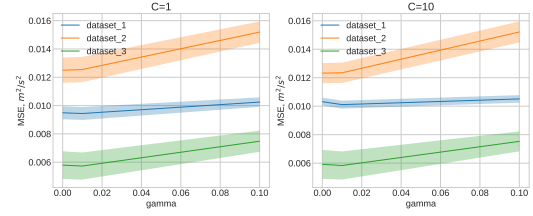


Fig. 5. Сумка, канал 0

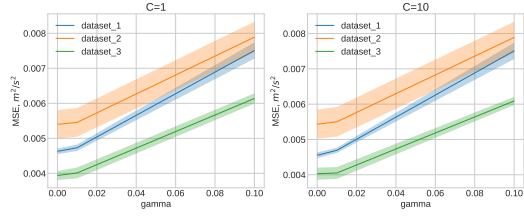


Fig. 3. Нога, канал 0

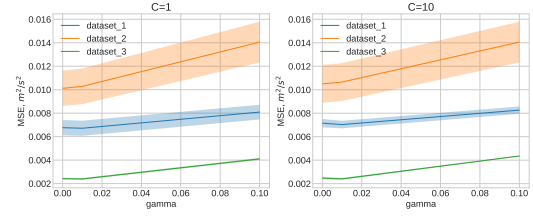


Fig. 6. Сумка, канал 1

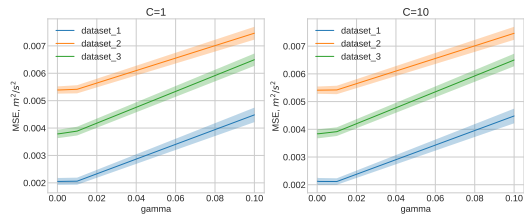


Fig. 7. Тело, канал 0



Класс-сумка                      Класс-рука

Fig. 9. Траектории

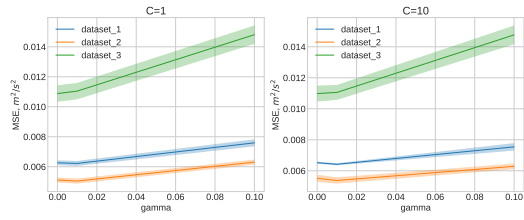
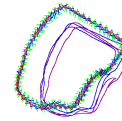


Fig. 8. Тело, канал 1

TABLE I  
Table 1

	Рука	Нога	Сумка	Тело
C	10	1	1	1
$\gamma$	0.01	0.001	0.01	0.001



Класс-нога

Fig. 10. Траектории