VAISL: Visual-Aware Identification of Semantic Locations in Lifelog

Ly-Duyen Tran[1(✉)], Dongyun Nie[1], Liting Zhou[1], Binh Nguyen[2,3], and Cathal Gurrin[1]

[1] Dublin City University, Dublin, Ireland
ly.tran2@mail.dcu.ie
[2] AISIA Research Lab, Ho Chi Minh City, Vietnam
[3] Vietnam National University, Ho Chi Minh University of Science, Ho Chi Minh City, Vietnam

Abstract. Organising and preprocessing are crucial steps in order to perform analysis on lifelogs. This paper presents a method for preprocessing, enriching, and segmenting lifelogs based on GPS trajectories and images captured from wearable cameras. The proposed method consists of four components: data cleaning, stop/trip point classification, post-processing, and event characterisation. The novelty of this paper lies in the incorporation of a visual module (using a pretrained CLIP model) to improve outlier detection, correct classification errors, and identify each event's movement mode or location name. This visual component is capable of addressing imprecise boundaries in GPS trajectories and the partition of clusters due to data drift. The results are encouraging, which further emphasises the importance of visual analytics for organising lifelog data.

Keywords: Lifelogging · GPS trajectories · Embedding models

1 Introduction

Lifelog refers to a comprehensive personal record of daily life activities captured by individuals called lifeloggers. Lifelog data can contain varying amounts of detail depending on the purpose of keeping such a record and could provide insight into an individual's behaviours [9]. Some examples of lifelog data can be images, videos, and biometrics collected from a variety of wearable devices and sensors. Due to its multimodal nature, lifelogging has many applications, namely, aiding memory, retrieving past moments, and reminiscing, to name a few.

Challenges in lifelogging research involve efficient capturing, storing, accessing, and analysing large volumes of multimodal data. This is because lifelog data tend to be passively captured in a continuous manner and the archive can grow in size very quickly. Thus, organising and preprocessing, including annotating/enriching data and segmenting lifelogs into *events*, are crucial to manage a lifelog [9]. The definition of *events* is not yet agreed upon; however, some factors are suggested to the basic contexts of a past event are *who*, *what*, *where*, and

D.-T. Dang-Nguyen et al. (Eds.): MMM 2023, LNCS 13834, pp. 659–670, 2023.
https://doi.org/10.1007/978-3-031-27818-1_54

when clues [23]. Therefore, boundaries in lifelogs can be created when there is a change of social interactions (e.g. with family, friends, strangers, etc.), activities (e.g., running, eating, working, etc.), locations (e.g. restaurants, parks, or personal locations such as home, work, etc.), or relative time (e.g. concepts such as yesterday, last night, etc.). This approach was used in [17] for semantic enrichment of lifelogs and, after that, segmentation.

Amongst these contexts, existing research [8] recognises the role played by location clues in memory recall. Although not a focal point of recall, they support recall over a prolonged period of time (which lifelogs are); trajectory reminders (locations visited before and after) help recall through inference; and location information might be useful when navigating through vast lifelogs. Furthermore, location-related contexts are a good indicator of activity. For example, being in a *restaurant* suggests *having meal* and in a *supermarket* suggests *shopping*. Previous work has been done on lifelog location data to provide structure to lifelogs [6,14], or to make meaning of food and physical activity behaviours [2]. In this paper, we further explore location contexts and propose a method of preprocessing, enriching, and segmenting lifelogs based on integrating location as a source of evidence. Regarding segmentation, there may be multiple events happening at the same location, and lifelogs could be further segmented based on other analyses in that location, which is beyond the scope of this work.

Generally, the processing of location data is based on GPS coordinates collected from a wearable device or a smartphone. Clustering methods, including variations of K-Means or DBSCAN [7,12,21,25] are used to detect places of importance, defined as clusters that contain more than a threshold of continuous data points [12]. These clusters are identified as 'stays' or 'stops'. Further characterisation of detected locations could be based on manual annotations [14,24] and automatic reverse geocoding to assign place identifiers to geographic coordinates [16]. Amongst these, location names and types attained from reverse geocoding can provide rich semantics for lifelogs. However, it requires highly accurate signals that most conventional GPS devices lack. One way to address this is to exploit lifelogs' multimodal nature and incorporate cues from other modalities, such as images. The success of computer vision models in various fields recently, especially in lifelog retrieval [1,11,22], has motivated us to explore ways to leverage vision to identify semantic locations from lifelog GPS data. Therefore, in this paper, we follow a conventional pipeline of GPS segmentation with the novelty of employing a pretrained text-image embedding model with the aim of enhancing the segmentation quality.

2 Related Work

2.1 GPS Trajectories Segmentation

GPS data have been widely collected by portable devices and smart phones. Much work has been done to segment GPS trajectories into *episodes* or *events* according to segmentation criteria such as *stops* or *moves*. This is equivalent to identifying the locations of interest from continuous GPS data. By doing this, semantics is added to the raw trajectories, allowing more complex analyses.

The most popular approach to location identification involves a density-based clustering algorithm to detect spatially connected GPS points with a distance threshold. These are variants of DBSCAN [7,12,21,25] or OPTICS [26] with more constraints, especially those related to time. For instances, [26] used both spatial and temporal aspects to define the density for the clustering algorithm. In [7], the authors proposed that all points in a cluster should be sequential in temporal order and should have an even distribution of direction changes. Hwang et al. [12] apply a temporal check after detecting spatial clusters to decide whether continuous data points belong to the same cluster. Some work applies a second step to fine-tune the location identification result by smoothing stop/move values [12], filtering out nonactivity stops (such as being stuck in traffic) using SVMs on stop attributes [7], or using improved clustering by fast search and identification of density peaks [5]. One drawback of density-based methods is that they require data to be collected with high frequency and accuracy [7]. Furthermore, a stop location could be divided into multiple smaller groups due to data drift [5]. This work attempts to address these shortcomings with the support of visual data.

2.2 Text-Image Embedding Models

Recently, pre-trained models have attracted considerable attention in different fields. The core idea behind pre-training is to gain knowledge from a massive amount of data, then transfer what has been learnt to various downstream tasks. These models have achieved impressive performance in computer vision tasks [4] such as image classification and object detection; and natural language processing tasks [15,20] such as machine translation and natural language inference.

The fusion of computer vision and natural language processing allows pre-training to be more scalable [3,19] by removing the need of a pre-defined set of object classes (e.g., 100 classes in ImageNet). Some notable examples are Contrastive Language-Image Pre-Training (CLIP) [18] and A Large-scale ImaGe and Noisy-text embedding (ALIGN) [13], which were trained on 400m/1B image-text pairs. These models embed images and texts and then utilise contrastive loss to minimise the cosine distance of matched image-text pairs and maximise that of nonmatched pairs. In particular, CLIP's zero-shot performance surpasses many baselines across different datasets [18]. Regarding lifelog, CLIP models have been integrated into lifelog retrieval systems and outperformed prior state-of-the-art techniques in the field [1,11,22]. In this paper, we focus on extending the applications of CLIP in lifelogging by complementing GPS data with a vision module.

3 Lifelog Dataset

Our method is applied to lifelog data from the fifth iteration of the Lifelog Search Challenge (LSC), LSC'22 [10]. This dataset features 18 consecutive months of multimodal lifelog data collected by one lifelogger. The data are organised in sequence, ordered by UTC time. Each data point, captured every minute, is aligned with various types of lifelog data such as music listening history, biometrics, and GPS coordinates. Furthermore, more than 725,000 point-of-view images, captured by a

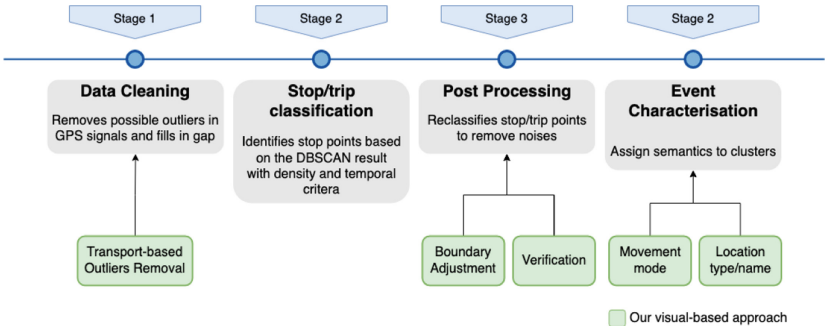


Fig. 1. Our method's workflow, following the conventional framework for location identification on GPS trajectories. Our contributions are highlighted in green. (Color figure online)

Narrative Clip wearable camera clipped on the lifelogger's shirt, are included and aligned to the minute sequence based on their time stamps. We would like to note that because the images are captured at a higher frequency (around every 30 s), a one-minute data point can be aligned with up to two images. This means that these two images, although possibly very different (for example, in two different locations), share the same GPS coordinates. We will address this in our method, described in the next section.

In this work, we are interested only in the lifelog images and GPS coordinates that come with the LSC'22 dataset. Because it is infeasible to produce a ground truth or thoroughly verify the results on the entire 18-month dataset, we choose to apply and validate our method only in the first month (January, 2019). This part contains 44,640 one-minute data points and 42,640 lifelog images. Among these, only 8,634 non-null GPS points are recorded.

4 The Proposed Method

Illustrated in Fig. 1 is our method. The main difference of this work from previous work is the incorporation of a visual module. Specifically, `L/14@336px`, the last published pre-trained CLIP model, was used to encode lifelog images.

4.1 Data Cleaning

Our proposed method follows the data cleaning and gap treatment process used in [12]. Specifically, we calculated the moving speed based on the spatial distance and the time duration between each point and its previous non-null data point. Outliers, as defined by data points having an unusually high speed, are removed. As noted by the authors, some false outliers (for example, as a result of speeding in transportation) could be removed. As we aim to reserve as many data points as possible, we improve outlier detection by incorporating the visual module to identify the transport mode and define the speed threshold for each mode.

Table 1. Texts used to assign transport modes to images.

Actual text	Label
I am sitting on an airplane	Airplane
I am in a car	Car
I am in a public transport	Public Transport
I am walking outside	Walking Outdoor
I am in an airport	Indoor
I am inside a building or a house	Indoor

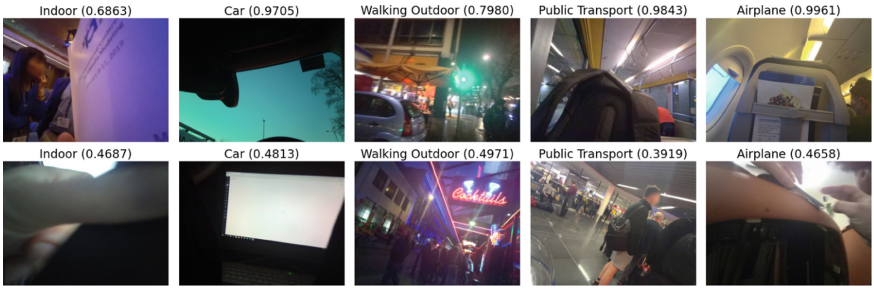


Fig. 2. Examples of transport modes classified by CLIP and their probabilities.

To assign the transport mode for each data point, we utilise the images aligned with each minute. The CLIP model is used to calculate the mean image features and compare them with the transportation labels using cosine similarity. The texts and their associated labels are specified in Table 1. For the rest of the paper, we will refer to the transportation modes as the labels in this table for the sake of brevity. As seen in Fig. 2, CLIP model is remarkably reliable. However, mis-classification does occur. By considering only the transport modes with a probability greater than θ, we can increase the model's precision. From these points, the speed thresholds are identified and are illustrated in Fig. 3.

After removing possible outliers, we apply a gap treatment process, similarly to that mentioned in [12]. For gaps whose time duration is at least q minutes, we add k data points to the gap with linear interpolated time and GPS coordinates. For more details and the rationale behind the process, see [12]. Moreover, we observe that the coordinates can be missing when the lifelogger is still or walking inside a building for a period of time. This results in a large number of false gaps in the stop/trip point detection module below (since null data points are considered as `trip` points). For this reason, we also interpolate gaps whose time duration is less than q minutes and whose movement mode is classified as *Indoor*.

4.2 Stop/Trip Point Classification

Similar to [12], the spatial clustering is performed using DBSCAN. After clustering, if a track log constituting a same spatial cluster are consecutive for the

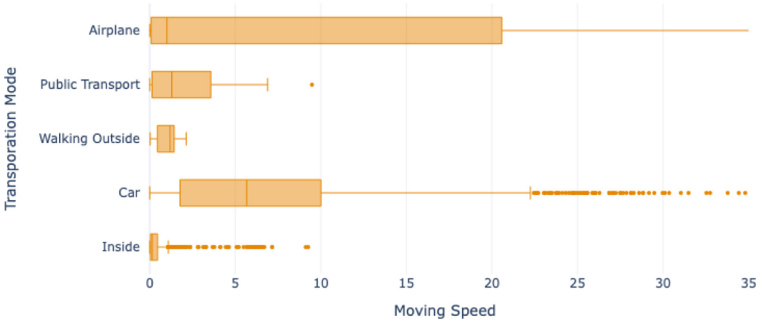


Fig. 3. Transport mode and their corresponding moving speed in the dataset. The upper speed thresholds are $1.5 * IQR$ where IQR is the interquartile range.

minimal duration of time t, then we classify them as `stop` points and assign a cluster ID to them. The rest are marked as `trip` points with a null cluster ID.

4.3 Post-processing

Smoothing: The previous method still leaves us with some mis-classified points. To address, we apply smoothing to the sequence `stop` and `trip` points. In effect, we replace the `stop`/`trip` values with the most common value of a window of three consecutive data points. Cluster IDs are also smoothed in the same way.

Boundary Adjustment: A specific step that is necessary for the vision module to work correctly in the next component is to modify the cluster boundaries to achieve a more accurate result. This is because for smaller clusters, if the boundaries are not clear, which can easily happen if we only consider GPS signals, the event can include both indoor and outdoor images, making it difficult to calculate the representative label of the whole cluster. Therefore, for each `stop` cluster, we consider expanding or shrinking it by examining whether the boundary image is labelled *Indoor* or not.

Verification: After boundary adjustment, we join consecutive data points with the same `stop`/`trip` and cluster ID values to form an event. For each event, the CLIP model is used to its images and average pooling is applied to obtain the event visual vector. This vector can be used to compare to the transport modes in Table 1 and verify the event type as described in Algorithm 1.

4.4 Event Characterisation

We then assign each event with its properties, summarised in Table 2. For example, we can get the begin time, end time, and duration based on the first and last minute IDs. Different processed are applied for `stop` and `trip` events.

Algorithm 1. Verify event type after boundary adjustments.

```
label, probability ← Compare(Event, Transport modes)
if probability ≥ θ then
    if label = Indoor then
        type ← stop
    else
        type ← trip
    end if
end if
```

Table 2. All event properties. ✓ indicates that the property is applicable.

	Property	Description	`trip`	`stop`
1	Begin Time	The first minute ID (of the event)	✓	✓
2	End Time	The last minute ID	✓	✓
3	Duration	The number of minute IDS	✓	✓
4	Visual Vector	Average pooling over all encoded images	✓	✓
5	Begin Location	GPS coordinates of the first minute ID	✓	
6	End Location	GPS coordinates of the last minute ID	✓	
7	Movement Mode	Airplane, Car, Bus, Walking, etc.	✓	
8	Centre Point	Mean GPS coordinates of all minute IDs		✓
9	Location Type	Restaurant, Airport, University, etc.		✓
10	Location Name	Name of the visited place, obtained from the FourSquare Nearbys API		✓

For `trip` events, the begin and end locations are obtained from the minute IDs. Also, we reuse the transport mode in the verification step in Sect. 4.3.

As for `stop` events, their centre points are the mean GPS coordinates of all points. In these experiments, we fixed two personal places as the lifelogger's homes, where they frequently stay. Thus, any `stop` that is less than 100m away from these places and assigned as HOME 1 and HOME 2. For the rest, instead of using straightforward reverse geocoding to attain the *closest* location name from the GPS coordinates, VAILS exploits *nearby* locations. This addresses the inaccuracy of off-the-shelf GPS devices by making use of visual cues to choose the best match. To do this, we use the FourSquare Nearbys API[1], which provides location information of nearby Places Of Interest (POI), such as

- **POI Name**
- **POI Types:** restaurant, airport
- **Related POIs:** parent/children locations
- **POI Images:** indoor, outdoor or menu photos uploaded by FourSquare users. However, the data are far from complete.

[1] https://developer.foursquare.com/docs/places-api-overview.

For each POI returned, we form a textual description using a template of 'I am in a {*POI types*} called {*POI name*}'. After that, we can compare the visual vector of the event with each of the POI descriptions and choose the most probable POI with its corresponding name and location type.

We also experiment on two more approaches to assigning the best POI to `stop` events. The first exploits the Related POIs result from the API (called Rel-VAISL), while the second takes advantage of the POI Images (called Img-VAISL). Regarding Rel-VAISL, the process is identical with one extra step: If the probability of the most probable POI is lower than a threshold θ, we merge the POIs belonging to the same parent and take the sum of their similarity scores to re-choose the best matching parent POI. This can be helpful in cases such as when the lifelogger moves between different parts of a building and the moving speed is generally not large enough for DBSCAN to distinguish the clusters. In the second approach, Img-VAISL, if there are images available for a POI returned by the API, we encode the images and take the average vector to compare with the event vector using cosine similarity. The final similarity score for each POI is the average of image similarity and text description similarity. However, since the POI image dataset is not complete, we cannot rely solely on images. Thus, for POIs without images, we use only their text description similarity. Similarly to the pure method, the POI with the highest similarity score will be assigned to the event. In addition, Combined-VAISL, which uses both of these approaches, is also included in the experiments.

The last step is merging consecutive events of the same type having identical Movement Mode (for `trip` events) or Location Name (for `stop` events). This step aims to mitigate the problems of density-based methods mentioned in Sect. 2.1. This reduces the number of events to a more accurate result.

5 Experiments and Results

In this section, we will analyse the results of our method. Since most of the process is similar to that of [12], we reuse their parameters with slight modifications due to the difference in recording time intervals. Specifically, the minimal time duration t remains unchanged as 3 min. However, since the recording time interval, r, in our study is 60 s instead of 30 s, the $MinPts$ for DBSCAN is set to 3 instead of 5. For gap treatment, the value of q is chosen in a way that $q = t + r$ (4 min) and that of k is $k = MinPts+1$ (4 data points). For DBSCAN clustering, we also reuse $eps = 50$ m. Finally, the *theta* threshold for CLIP classifications is chosen as 0.75.

To evaluate our method, we use the location history from the Google Maps API[2] that has been manually validated by the lifelogger. The location logs contain information of visited places and transportation segments, the equivalences of `stop` and `trip` events. According to the logs, the first month of 2019 can be segmented into 206 `stop` events and 214 `trip` events in between. All properties listed in Table 2 can be interpreted from the logs as the ground truth.

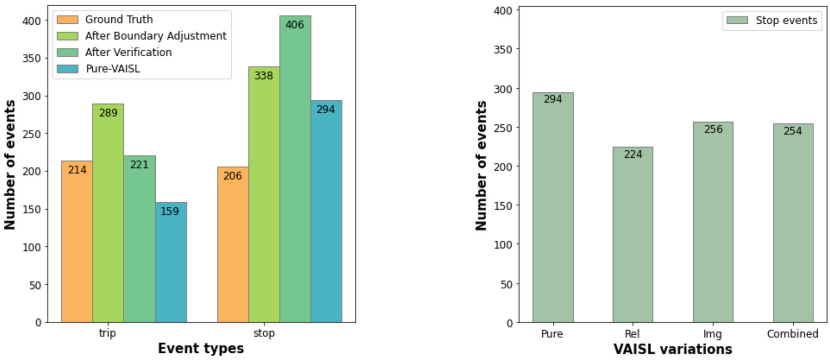
[2] https://www.google.com/maps/timeline.

5.1 Event Detection Results

After finishing our proposed method, we assign each minute point with the `stop`/`trip` values. We compare these values to those extracted from the ground-truth location history. The classification confusion matrix can be seen in Table 3 with an accuracy score of 0.9663 and a Kappa index of 0.8155.

Table 3. Error matrix of stop/trip values from the ground truth and our result

		Ground Truth	
		Stop	Trip
VAISL	Stop	**37594**	718
	Trip	717	**3611**



(a) Compared to the ground truth.

(b) Different variations of VAISL.

Fig. 4. Number of events detected at different steps. Despite having the identical move events, VAISL's variations result in various numbers of stay events.

Regarding the number of segments, after boundary adjustment, there are 338 `stops` and 289 `trips` detected. The verification step converted a considerable amount of `trip` segments to `stop`. After identifying event types and merging identical consecutive episodes, VAISL's result contains 294 `stops` and 159 `trips`, as summarised in Fig. 4. We also observe different numbers of `stop` events for VAISL variations, with the highest figure from Pure VAISL. By merging places that belong to the same building or organisation together, Rel-VAISL results in 224 `stops`. Moreover, similar figures are obtained when using the POI images returned by the FourSquare API.

5.2 Event Characterisation Results

Trip Events. As for moving periods, the movement modes provided here are of only WALKING, IN_PASSENGER_VEHICLE, and FLYING. Thus, for an easier comparison, we transform the VAISL's movement modes (in Table 1) from *Walking Outdoor* to WALKING, *Airplane* to FLYING, and *Car*, *Public Transport* to IN_PASSENGER_VEHICLE. The *Indoor* label can be ignored as we have already changed it into a `stop` event in the verification step.

Out of 159 `trip` events detected, 128 of them are assigned a correct transport mode. The remaining 31 events are inspected manually and most of the mismatches are due to the imprecise boundaries of the groundtruth. Image-wise, the classification achieved 0.9017 accuracy in this subset.

Stop Events. Regarding the `stop` events, we are interested in evaluating the Location Name property. We could not do this automatically due to some discrepancies in the location names provided by Google Maps API and FourSquare API. Some examples are Dublin Airport vs. Dublin Airport (DUB) and Citywest Shopping Centre vs. Eddie Rocket's (which is inside the shopping centre). Therefore, we asked the lifelogger to manually verify the detected `stop` events. Due to the large amount of data and the limited time we have with the lifelogger, we were only able to verify 14 out of 31 days. We chose to exclude three locations where the lifelogger spent most of the time to get a less biased view of the result.

Table 4. Analysis on VAISL's results on 14 days.

Variations	#stops	#correct	Accuracy
Pure	65	42	0.63
Rel	65	**45**	**0.69**
Img	67	41	0.61
Combined	65	44	0.68

The results are summarised in Table 4 for the approaches mentioned in the previous section and slight differences are observed. The highest performing option is Rel-VAISL which exploits POI relationship information. This helps correct mistakes in Pure-VAISL where the segmentation between different indoor places are not well-adjusted, resulting in choose a wrong POI. On the other hand, the idea of using user-uploaded POI images has not proven useful, most likely because the process favours POIs with images much more than others. Thus, Img-VAISL and Combined-VAISL both have lower accuracy than their counterparts, Pure-VAISL and Rel-VAISL, respectively.

6 Discussions and Conclusion

This paper described an automatic method for organising and enriching lifelogs based on location contexts. The advancement of embedding models has allowed us to perform a better analysis of lifelog data. By integrating images with GPS data, this work further confirms the importance of its visual aspects in lifelogs, especially in segmentation. We believe having a well-segmented lifelog with accurate location semantics can add value to different lifelog applications. However, because we only have access to one lifelogger's data, it would be interesting to see how this applies to other lifelogs. Future work on this could include personal location identifications. Personal locations are user-specific; some examples include one's office and relatives' homes. In this work, we fixed two places as the lifelogger's homes before performing event characterisation. It would pose a challenge when this number increases, as these locations cannot be returned from any reverse geocoding services. Additionally, how to effectively evaluate location analysis on a large lifelog dataset remains a difficult question.

References

1. Alam, N., Graham, Y., Gurrin, C.: Memento: a prototype lifelog search engine for LSC'21. In: Proceedings of the 4th Annual on Lifelog Search Challenge, pp. 53–58. Association for Computing Machinery (ACM) (2021)
2. Andrew, A.H., Eustice, K., Hickl, A.: Using location lifelogs to make meaning of food and physical activity behaviors. In: 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, pp. 408–411. IEEE (2013)
3. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
5. Fu, Z., Tian, Z., Xu, Y., Qiao, C.: A two-step clustering approach to extract locations from individual GPS trajectory data. ISPRS Int. J. Geo-Inf. **5**(10), 166 (2016)
6. Gomi, A., Itoh, T.: A personal photograph browser for life log analysis based on location, time, and person. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1245–1251 (2011)
7. Gong, L., Sato, H., Yamamoto, T., Miwa, T., Morikawa, T.: Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines. J. Mod. Transp. **23**(3), 202–213 (2015). https://doi.org/10.1007/s40534-015-0079-x
8. Gouveia, R., Karapanos, E.: Footprint tracker: supporting diary studies with lifelogging. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2921–2930 (2013)
9. Gurrin, C., Smeaton, A.F., Doherty, A.R., et al.: LifeLogging: personal big data. Found. Trends® Inf. Retrieval **8**(1), 1–125 (2014)
10. Gurrin, C., et al.: Introduction to the fifth annual lifelog search challenge, LSC'22. In: Proceedings of the 2022 International Conference on Multimedia Retrieval, pp. 685–687 (2022)

11. Heller, S., Rossetto, L., Sauter, L., Schuldt, H.: Vitrivr at the lifelog search challenge 2022. In: Proceedings of the 5th Annual on Lifelog Search Challenge, LSC 2022, pp. 27–31. Association for Computing Machinery, New York (2022)
12. Hwang, S., Evans, C., Hanke, T.: Detecting stop episodes from GPS trajectories with gaps. In: Thakuriah, P.V., Tilahun, N., Zellner, M. (eds.) Seeing Cities Through Big Data. SG, pp. 427–439. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-40902-3_23
13. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. arXiv:2102.05918 [cs], June 2021
14. Kikhia, B., Boytsov, A., Hallberg, J., ul Hussain Sani, Z., Jonsson, H., Synnes, K.: Structuring and presenting lifelogs based on location data. In: Cipresso, P., Matic, A., Lopez, G. (eds.) MindCare 2014. LNICST, vol. 100, pp. 133–144. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11564-1_14
15. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692 [cs], July 2019
16. McKenzie, G., Janowicz, K.: Where is also about time: a location-distortion model to improve reverse geocoding using behavior-driven temporal semantic signatures. Comput. Environ. Urban Syst. **54**, 1–13 (2015)
17. Qiu, Z., Gurrin, C., Smeaton, A.F.: Evaluating access mechanisms for multimodal representations of lifelogs. In: Tian, Q., Sebe, N., Qi, G.-J., Huet, B., Hong, R., Liu, X. (eds.) MMM 2016. LNCS, vol. 9516, pp. 574–585. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27671-7_48
18. Radford, A., et al.: Learning transferable visual models from natural language supervision. arXiv:2103.00020 [cs], February 2021
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
20. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs], February 2020
21. Schoier, G., Borruso, G.: Individual movements and geographical data mining. clustering algorithms for highlighting hotspots in personal navigation routes. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011. LNCS, vol. 6782, pp. 454–465. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21928-3_32
22. Tran, L.D., Nguyen, M.D., Nguyen, B., Lee, H., Zhou, L., Gurrin, C.: E-Myscéal: embedding-based interactive lifelog retrieval system for LSC'22. In: Proceedings of the 5th Annual on Lifelog Search Challenge, LSC 2022, pp. 32–37. Association for Computing Machinery, New York (2022)
23. Tulving, E.: Precis of elements of episodic memory. Behav. Brain Sci. **7**(2), 223–238 (1984)
24. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with GPS history data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 1029–1038 (2010)
25. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L.: Discovering personally meaningful places: an interactive clustering approach. ACM Trans. Inf. Syst. (TOIS) **25**(3), 12-es (2007)
26. Zimmermann, M., Kirste, T., Spiliopoulou, M.: Finding stops in error-prone trajectories of moving objects with time-based clustering. In: Tavangarian, D., Kirste, T., Timmermann, D., Lucke, U., Versick, D. (eds.) IMC 2009. CCIS, vol. 53, pp. 275–286. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10263-9_24