Track It (again)

Pedro André Gomes
gomes.a.pedro@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2015

Abstract

Track It is a GPS application oriented to outdoor sports and activities that allows users to review and analyze records from activities previously carried out, as well as planning future ones. This application was created prior to this work for a multi-platform environment. This work presents significant improvements to the application and the reasons behind these improvements are explained as well as how they were implemented. The highlight of these improvements are the creation of a repository of open documents, allowing to resume the progress of previous sessions and search activities by area. In addition to these improvements, three new algorithms were introduced to calculate speed, detect pauses and detect the position of media items on the map view according to the GPS path which they are associated with. These algorithms were tested with good results that show potential for improvement and still more precise results.
Keywords: GPS, pause, speed, multimedia, coordinates detection

1. Introduction

1.1. Motivation

Track It is a Global Positioning System (GPS) geared to outdoor activities and sports that allows to review activities previously carried out or plan future activities, through the display of various types of charts, graphs and tables showing collected information or which has been introduced during planning.

The original Track It arose from the growth in popularity of this genre of applications, especially among athletes, coaches and amateurs in general sports and outdoor activities. This trend can be attributed mainly to the increased robustness and functionality of GPS devices and due to the decrease of their price.

This work aims were to review the features and behaviors of Track It and expand or improve them as needed, adding new features so as to enrich the software in order to stay competitive.

GPS devices have been becoming increasingly robust and precise, enabling the collection of a wide range of raw data in large volumes. This enables enthusiasts of outdoor sports and activities to collect increasingly comprehensive information, allowing better data analysis to the practice of sports and competitions.[8]

However, the software currently used to display and analyze the data collected is still far from able to meet the needs that a user can have apart from analysis of physical fitness. The majority of this software is limited and many are dedicated to a single type of activity, with problems that range from missing features to others that require multiple steps with different software.

The work presented in this paper was carried out on the initial version of the application, which already supported some of the more common features among applications with GPS support for outdoor sports and activities (such as data analysis and planning future activities), making its goal to improve, fix and add features to this software in order to significantly improve its abilities .

1.2. Previous Development

Track It was developed in Java[1], using a Swing interface[2]. It enables the visualization of paths on different map types from various sources[3]. Furthermore the software is able to import several paths,

[1] Oracle (`www.java.com`)
[2] Oracle (`http://www.oracle.com/technetwork/java/architecture-142923.html`)
[3] using Google Maps, Bing Maps, Here Maps, OpenStreet Maps, MapQuest Maps, ArcGIS Maps and military maps

without allowing their simultaneous viewing, edit them by manipulating their trackpoints or joining courses together. It also presents charts, where users can select which values to display, and tables with the corresponding data.

The application architecture consists of three components, a front-end and a back-end, along with an additional module responsible for file import/export. This architecture allows the separation of data processing and the user interface and the support of additional file formats, when such a need arises.

The front-end contains the user interface module, and for the purpose of supporting a set of synchronized views, including maps, charts, data tables, view summary and details as well as a browser that allows the user to exchange views and the records that are displayed. This module needed to be upgraded with the aim of giving greater support to the possible actions, change the selection mode of the axes of the charts and support adding and viewing multimedia elements such as photographs.

However the back-end, in its state previous to this work, only contained the implementation of some algorithms needed for the various features of the software and as such, still did not have an internal database, preventing the creation and maintenance of collections of activities. Furthermore, some of the implemented algorithms lacked depth, were too simple to achieve reliable results and could harm the results of the following calculations.

The previous version of the application had no support for any type of media, not even text notes to sections in general or particular points.

1.3. Goals and Research Topics

This study aimed to identify and explore existing problems and features of current software, including Track It itself, and propose appropriate solutions for each case and test them. The proposed goals were:

- Allow the registration of courses and activities in the application through a database.
- Create a search function for registered activities and courses.
- Simultaneously present several courses or activites and assign them different properties (e.g. color).
- Study and test algorithms for determining pauses (breaks) in a recorded path.
- Study and test a precise algorithm for speed calculation.
- Permanently add multimedia content to an activity or course and represent it.
- Estimate the geographical position of multimedia elements lacking such information.

The storage of paths (courses or activities) in the application will allow the user to resume previous work. The issues in this area go from how best to save the data to how to incorporate it in the software behavior so that it doesn't become a hindrance.

Simultaneously display of multiple records will allow to compare several records of the same path or even independent recordings obtained in the same zone, so they can be analyzed or even joined in one new path. The major concern in this situation is how to visually present the different paths on a map, allowing for their clear distinction.

The ability to add georeferenced(geotagged) multimedia content provides a better visual representation of the context of a recorded track. However, to actually place these items correctly on a map, these multimedia items must have location information (e.g. according to the Exif[1] specification) or, where such data is not available, they must have their date and time of creation in order for their position to be estimated by the algorithm proposed in this paper.

2. State of the Art

In this section we analyze existing software products, focusing on those who have similar features to this work, or related to the desired features. These applications are available to the general public and stand out positively in one or more features.

Additionally, this analysis intended to cover the major platforms used today: Windows, MacOS, Linux, and mobile apps (for Android, iOS, BlackBerry, Windows Phone, etc.) and, in the latter case, gives emphasis to those that run simultaneously as web services, and select those applications that represent the features typically supported by systems that work with GPS records, such as review, planning and searching for new routes.

The analyzed applications were:

- Sports Tracker[4]
- Endomondo[5]
- Strava[6]
- GPSies[7]
- Sports Tracks[8]
- Ski Tracks[9]
- Map my Ride[10]

The features that we focused upon for this analysis were:

- Planning future activities
- Search for other routes
- Simultaneously display of multiple routes (for comparison, etc ...)
- Add multimedia elements to recorded activities
- Estimate the position of multimedia elements without geolocation data
- Detecting pauses or less relevant sections

The planning of futures activities cannot always be done from previously recorded routes and creating a new one from can take a lot of time. However it is noteworthy that in order to reduce this difficulty various applications give suggestions to their users be it by following the shortest route through the streets between two points or using existing sections of other activities (the most popular ones) that pass close to two given points.

Support for adding multimedia content and its subsequent presentation on the map is a feature that has been gaining weight in these applications, especially among those who place more emphasis on sharing activities within their user community. This feature arises primarily in applications available on smartphones, which can capture images at the same time that the path is recorded. There are only few cases where the GPS application ensures that these are marked with the geolocation. This means that importing photographs of different activities is unusual and adding photos without geolocation information is non-existent.

Searching for activities is also a feature that is more common among applications that try to build communities. Search functions range from simple text and tag search to more complex ones by proximity or region. Sometimes searching by region is presented as a map where the user can drag the map and zoom in or out to determine the area to search.

A feature that is clearly unusual is the detection of pauses or sections of less interest, even though the only application with similar functionalities is Ski Tracks, which distinguishes, though not always successfully, ascents by mechanical means.

3. Application Enhancement

The following section lists the more relevant changes and additions that were made to Track It throughout this work.

[4]Sports Tracking Technologies (`http://www.sports-tracker.com/`)
[5]Endomondo.com (`http://www.endomondo.com/`)
[6]Strava (`http://www.strava.com/`)
[7]GPSies (`http://www.gpsies.com/`)
[8]Zone Five Software (`http://www.zonefivesoftware.com/sporttracks/`)
[9]Core Coders (`http://www.corecoders.com/applications/ski-tracks/`)
[10]MapMyFitness (`http://www.mapmyride.com/`)

3.1. Features
3.1.1 Storage

One of the major absences in the previous version of Track It was the lack of persistence between multiple sessions. In order to correct this, it was decided to change the applications architecture by adding a database and internal knowledge on whether a course was created and never saved or if it had unsaved changes, thus avoiding unintended loss of data.

The use of the database allows Track It to keep tabs on open documents without the need to create additional, and unnecessary, copies of said files, while also allowing for the storage of additional information about these documents, such as bounding box coordinates.

The database was created using the SQLite[11] relational database management system. This system remains embedded within applications that use it, while providing tools to handle various types of data with fewer restrictions and greater ease. This is due to the fact that it is a database based on independent files compared to systems that use server services for the same purpose.[3] In addition, the whole application can be moved within the same file system and it's also possible to create backup copies as the database is a single file.

The database is a file created by default by the application in its home folder. This file contains tables of files with GPS information of routes, as well as tables relating to the multimedia content, photographs which were attached to tracks, in this case.

Since each GPS file may contain one or more paths, they are individually recorded in the database and thus the same file may appear in several different records, distinguished by the primary key Filepath and their respective Name. Apart from these fields there are also values relative to the bounding box of each path, i.e. latitude and longitude, minimum and maximum.

The table on multimedia content (photos in this case) contains the Filepath and the name of each image. It also allows importing as foreign key the name(ParentName) and filepath (as Container) of the path they are associated with, the latter two fields and Filepath serving as primary key to the table. In addition the table also stores latitude, longitude and altitude recorded in the Exif[1] data.

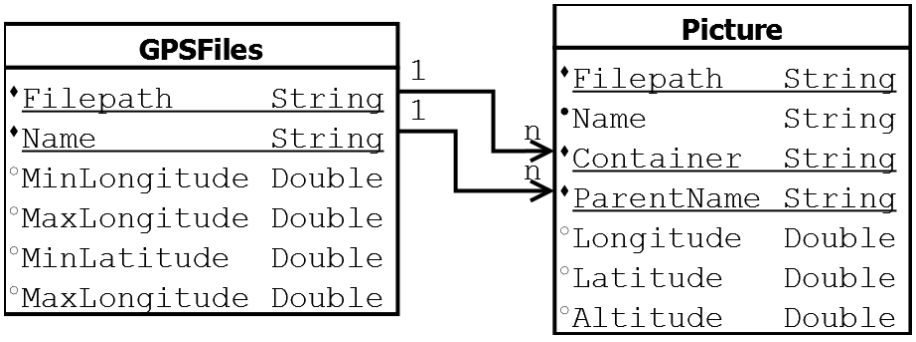

Figure 1: Database scheme currently used in Track It.

In this way, each time the software is started, it searches the database for the open files list of the previous session, as well as associated photos. This way it is possible to maintain all paths and their photos over several sessions.

It should be noted that in cases where the application can not locate a file, when the file has been moved or had its name changed, for example, then the respective records are deleted so as to prevent that database contents end unnecessarily polluted.

3.1.2 Path Search

Finding paths is a great asset, especially when the application has multiple documents open simultaneously. For this purpose it was decided to create a new program mode to show which paths are inside a given area.

For this purpose the user should make visible the area he wants to search on the map view and then choose the new search mode area and then select the area by clicking and dragging the pointer in order to draw a box over the desired area. From this box that the user draws on the interface, four coordinates are

[11] SQLite (`https://sqlite.org/`)

acquired, taken from the point where the user first clicked and the point where the button was released, with these two points representing a bounding box.

A bounding box is the concept of four limits or boundaries connected together, forming the image of a box, this is represented by two points (the top left vertex and lower right vertex or the lower left vertex and the upper right vertex).[2] From these two points it is possible to compute the coordinates of the other two points and from these the coordinates of points belonging to the edges, or boundaries, of the same box. Thus it is possible to know if any one point or set of points is within these limits.

With these coordinates Track It queries its database for paths within these boundaries. Results are presented to the user by buttons, which are placed at their corresponding start point on the map view. These buttons are clickable and when pressed zoom in on the corresponding course or activity.

3.1.3 Presentation of Multiple Paths Simultaneously

Displaying multiple paths at the same time is useful when you want to compare them or their values (be it time, distance, speed, among others), or when the user wishes to perform operations that require more than one course, such as joining two as a single course.

In order to allow this, a new menu was created that shows all loaded paths. When this menu is shown all other selection operations are blocked by the application. Every selected path in this menu is shown simultaneously on the map view, with the ones out of the area shown on the map made invisible.

The biggest problem in this situation is the distinction of several paths overlaping each other. The solution to this problem was to allow the user to select the color of any path to be shown. This color can be chosen on a color chooser or picked from available color swatches, with the selected color remaining for the duration of the current session. Related to this, the option to select an initial default color for paths was also added to the application's preferences, this default color being overriden for each path by the selected color in the multiple selection menu.

3.1.4 Adding Multimedia Content

The hability to add geotagged pictures is becoming increasingly popular and it also allows for better visual representation, be it for the need of visual cues or in order to review a past performance.

To work correctly, the option to import photographs containing Exif data needs that imported files contain the date/time stamp when the image was captured and, optionally, information about its geolocation. Imported pictures are associated with a path and are shown at their position on the map view. In cases where there's no available geolocation information pictures are initially put at the middle point of their associated path.

In order to better distinguish the photo thumbnail from the map a frame was put around them, with a solid color that the user can choose. When a thumbnail is clicked, the corresponding photo is open in a new window, which adapts its size to the current environment. It's also possible to change the zoom level in this new window.

3.1.5 Selective Representation in Chart View

In previous versions of Track It, it was impossible to toggle elevation as a magnitude, while it was also possible to select other magnitudes for which there was no data available (e.g. cardiac frequency).

To avoid these situations, when an activity or course is selected, Track It checks which information they contain and only the available magnitudes are selectable. Furthermore, it is now possible to toggle elevation and speed is now the default magnitude initially presented, because it is the most common magnitude of outdoors sports and activities.

3.1.6 Keeping date/hour stamps

A problem of earlier versions of this application was that the timestamps were changed automatically whenever a course was edited. This happened, for example, when points were added or eliminated from a course.

This data destruction was done by the application that recalculated a course whenever it changed imposing an average speed throughout the course, that changed times and speeds.

As this behavior is only desirable in some situations, but not in general, the application now allows users to decide whether or not to maintain the date and time stamps while editing a course, assuming by default that timestamps should be maintained.

3.2. Algorithms

3.2.1 Speed Algorithm

Speed represents the distance traveled by an object in a given time interval.[5] It can be calculated by GPS devices, however there are some that don't do so or whose output formats omit speed. In this situation it is necessary to estimate the speed at each point captured in a path and its date and time stamp. These calculations may have errors due both to how each device registers trackpoints and time stamps, as well as the existence of sources of location errors such as drift, vegetation, tall structures, etc.

GPS drift is a phenomenon that occurs when the device that records the path is stationary but nevertheless keeps recording positions with minor differences from the preceding ones.[13] This creates an illusion of small movements or a continuous movement at an extremely low speed, when in fact the device is stationary. Due to this reason, devices may also register incorrect speed values in the files.

The previous version of Track It calculated speed according to a very simple formula. Speed was the distance between two trackpoints divided by the time it took to go from one position to the other, while not taking into account some problems that might arise in a recording, such as multiple trackpoints with the same date/hour stamp. These problems created anomalous acceleration spikes followed by decreases over a long period of time.

In order to improve speed calculation and avoid the problems presented above, it was necessary to change the rules used to calculate it. In order to facilitate the detection of pauses, which is discussed in more detail in 3.2.2, the decision was made to calculate the speed in two passes, as follows.

1. The average speed for each of the n-1 gaps (numbered from 0 to n-2) are calculated as:

$$V_i = \frac{d_{i+1} - d_i}{t_{i+1} - t_i}$$

2. In the interior points (1 to n-2), which are not limits of pause intervals, the speed is calculated using the center-weighted finite difference[9]:

$$V_i = \frac{1}{(t_{i+1} - t_{i-1})} \left((t_i - t_{i-1}) \frac{d_{i+1} - d_i}{(t_{i+1} - t_i)} + (t_{i+1} - t_i) \frac{(d_i - d_{i-1})}{(t_i - t_{i-1})} \right)$$

3. In extreme points (0 and n-1) that don't belong to pause intervals, the respective speeds are approximated by ascending or descending finite differences[9] depending on the situation:

$$V_0 = 2\frac{d_1 - d_0}{t_1 - t_0} - V_1$$

$$V_{n-1} = 2\frac{d_{n-1} - d_{n-2}}{t_{n-1} - t_{n-2}} - V_{n-2}$$

4. At points where the date/time stamp is equal to the date and time stamp of the point immediately before or after, but not equal to the two simultaneously, the speed is approximated by:

$$V_i = \frac{d_{i+1} - d_{i-1}}{t_{i+1} - t_{i-1}}$$

5. In the case where the date and time stamp a point is equal to the marks of the anterior and posterior points of the approach speed will be:

$$V_i = V_{i-1}$$

The first rule presented is applied to all trackpoints during the first pass. Only the points with a speed value above a given threshold are subject to the remaining rules. The two last rules prevent infinite speeds in the case of contiguous trackpoints with the same date/time stamp.

With the new speed approximation algorithm, as can be seen in figure 2, the sets of points where there are likely pauses can be identified more easily. This happens because there are fewer slopes, replaced by sets of points where the speed falls and approaches the zero value. Thus the speed profile becomes more natural and avoids abrupt changes of speed, which often result from the GPS drift as mentioned above.

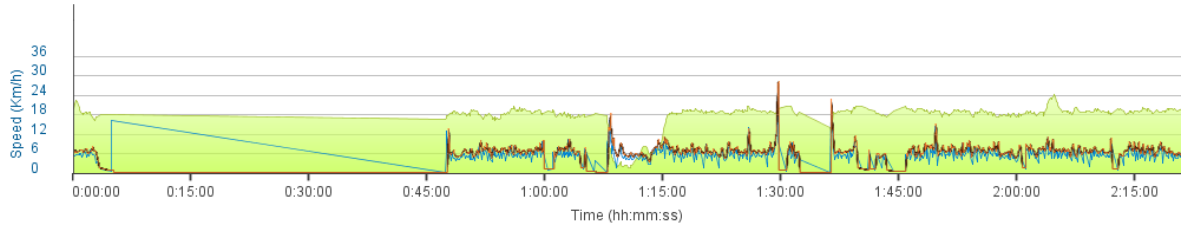

Figure 2: Overlap in the graph view to speed on the vertical axis from the previous version (in blue) with the corrected version of Track It resulting from this work (in orange).

3.2.2 Pause Detection Algorithm

A pause period is a time interval when one is stationary, or move short distances, or at very low speeds. As was explained in 3.2.1 the recording of positions and speed can be affected by GPS drift. There's also the case where GPS devices cannot correctly record the time its user stops, but does so a few moments later.

In the previous version of Track It the total pause time was only a sum, where if a point had a speed value below a threshold the time between that point and the previous one would be added to the total. This meant that it became impossible to know when a break began and ended and consequently its duration, a feature necessary for the automatic photograph placement algorithm.

It was necessary to find and maintain for each activity and course a list of its pause periods, to know the amount of time the user was stationary or moved at an extremely low speed and also when each pause started and ended.

The new algorithm is the result of the implementation of several variations, with some using weighted averages and others with simply the trackpoints values, for speed and distance. After some tests with these versions it was clear that using speed presented better results over the versions that used distance.

The final version of the algorithm for calculating the values to fill the structures responsible for keeping the information on the break times, which in turn has been changed to have better integration with the speed estimation algorithm presented in 3.2.1 follows the pseudo code shown below:

Let V_{pause} be the speed limit below which there is a pause, $T_{entrance}$ and T_{exit} output value lists, initially empty, of the type Date/Time representing the beginning and end, respectively, of each pause period and v and t speed and time values, respectively, both with n elements numbered from 0 to n-1. Pauses are detected according to the following pseudocode:

$T_{pause} \leftarrow 0$
$entering \leftarrow true$
if $v_0 \leq V_{pause}$ **then**
  $T_{pause} \leftarrow -t_0$
  $entering \leftarrow false$
  Insert t_0 in $T_{antrance}$
end if
for $i = 1; i < n; i++$ **do**
  $V_{teste} \leftarrow (v_i - V_{pause}) * (v_{i-1} - V_{pause})$
  if $V_{teste} < 0$ **then**
    $t \leftarrow (t_i - t_{i-1}) * (V_{pause} - v_{i-1})/(v_i - v_{i-1}) + t_{i-1}$
    if $entering$ **then**
      $T_{pause} \leftarrow T_{pause} - t$
      Insert t in $T_{entrance}$
    else
      $T_{pause} \leftarrow T_{pause} + t$
      Insert t in T_{exit}
    end if
    $entering \leftarrow !entering$
  end if
end for

if $entering = falso$ **then**
  $T_{pause} \leftarrow T_{pause} + t_{n-1}$
  Insert t_{n-1} in T_{exit}
end if

With the results obtained with the implementation of the pseudo code above it became possible to know the end and beginning of each break and, from these, its duration. After testing different activities it was concluded that the speed limit must be set according to the type of activity.

However since the application cannot still distinguish between types of activities the default speed limit value was set to 1,5 km/h (= 0.4166 m/s), since 1 km/h provided false positives in some tests and 2 km/h was a value that could be achieved in leisurely walks. An option to change this limit between 0 and 10 km/h was also added to the application's preferences.

3.2.3 Automatic Placement of Photographs

Geotagging is a feature that has gained popularity in recent years and that allows a user to add the location coordinates where the photos were captured on the files Exif[1] information. Using this new functionality in the devices used to capture photographs, digital cameras or embedded in mobile devices, the ability to display captured photographs has become increasingly common among applications targeted for the treatment of GPS data.

Despite its popularity there are still several cases where geotagging is not possible due to the equipments inability to do so or even by user choice. The latter case can arise for security and privacy reasons especially in situations where the photos are uploaded to social networks.[6]

Currently there is software dedicated to determine the location where photos were captured from GPS records. However these programs are limited to seeking the point which has the closest date and time stamp to the time stamp of the photograph, with the coordinates from that point added to the photo's Exif information.[10][11] This requires that the clocks of both devices (camera and GPS device) are synchronized, and the pictures whose date and time stamp is outside the time interval recorded in the GPS file are discarded.

For the correct operation of this algorithm it is assumed that a person has to perform a pause to take a photograph, in accordance with what is described in 3.2.2, two or more photographs must be taken and they must be from the same device.

The algorithm aims at finding out during which break the first picture that is associated with a path was taken. This first picture is assigned to the central timestamp (for a pause between 14:10:00 and 14:20:00 the central timestamp will be 14:15:00) of that pause and will have the geolocation of the last point prior to that date and time stamp, thus being associated with a position belonging to the path. The remaining pictures are assigned new date and time stamps as well as new geolocation coordinates, according to the temporal offset that its original date and time stamp had with the date and time stamp for the first picture.

With pausesCenter the central date and time values of each pause, numbered from 0 to n, photosIntervals the interval between each photo, numbered from 0 to m and Pauses a structure that contains all the pause periods in an activity, the pause that will be assigned to the first photo and from which the others are adjusted, is determined according to the following pseudo code:

$target \leftarrow 0$
$targetAccuracy \leftarrow -1$
for $i = 0; i < n; i++$ **do**
  $accuracy \leftarrow 0$
  $middle \leftarrow pausesCenter[i]$
  for $t = 0; t < m; t++$ **do**
    $middle \leftarrow meio + photosIntervals[t]$
    if $middle \in Pauses$ **then**
      $precisao \leftarrow precisao + 1$
    end if
  end for
  if $accuracy > targetAccuracy$ **then**
    $target \leftarrow i$
    $precisaoAlvo \leftarrow precisao$
  end if

end for

This way it is also possible to correct the date and time stamps of the pictures, which is especially useful in cases where, for example, the camera had the wrong time set or was set to a different time zone.

To test the results obtained from this algorithm two records were chosen, the first obtained in the area of Albufeira in Algarve, an area where there are obstacles (tall structures), and the second record from a walk in Belém in Lisbon, an open area but close to the water, which may cause jitter in registered positions.[4][7] Each of these records was accompanied by a set of nine photos with geolocation information at the time they were taken, which allows us, assuming that the information in the files of the photos is correct, to compare the positions recorded by the device used to his capture with positions estimated by the presented algorithm.

The images were captured using the iPhone 4S Camera app, in both cases. The Albufeira path was recorded with GPS Track app and the Ski Tracks app was used to record the Belém path. Both paths were recorded in GPX 1.1 format, which doesn't record speed.[12]

With the Albufeira record the average distance after applying the algorithm, when compared to the original positions, was of 42,5m with values between 27,37 and 55,66 meters. The difference between the timestamps was of -64 seconds after the calculations.

The record from Belém presented results ranging between 3,7 and 20,05 meters, with and average of 11,71m. In this case the timestamps were set 44 seconds ahead.

As it happens with any algorithm, the results vary from case to case, as can be seen here where the case from Belém shows much better results than the case of Albufeira, due to the different conditions of the recordings, with Albufeira being an urban zone, which can cause some displacement in the GPS devices.

Other sources of error are the higher speed sections, where larger displacements occur between the original position and the one estimated by the algorithm, the other one being the use of the heuristic that the first picture is captured precisely at the midpoint of the pause assigned to it by the algorithm. Furthermore, it must be noted that the algorithm is sensitive to big pause intervals, when all the photographs are set to the same pause period and end with very similar coordinates.

Although encouraging, these results still require a more rigorous confirmation, since it's not possible to know the precision of the coordinates registered in the file's Exif information.

One change that was untested was the ability to treat pictures from different devices in separate batches.

4. Conclusions

The registration of courses and activities by the application as well as the addition of multimedia content has been achieved through the inclusion of a database in the application's architecture. This database in addition to ensuring these features, opens up a range of new options for future work, including assignment of information from the application to each path, information which enables the application to better accommodate user needs under the type of activity that is being viewed at any given moment.

The possibility to present several paths simultaneously was created and to avoid a visual clutter on the map view, the option to change the color of each of these paths was added.

The search method introduced in the application allows searches per area and serves as an example on how to use the database for operations in possible future works.

The speed calculation algorithm presented and implemented allows more precise and natural speed values at each point, avoiding peaks of excessive speed.

The pause calculation and position detection for multimedia elements algorithms show satisfactory results, where the former can be used for segmenting routes in courses and activities. As for the position detection algorithm though, as shown, it is possible to get very close to what would be the geolocation recorded by any camera, its results can also be prone to errors, particularly in the presence of large time breaks. Despite these results, further testing, with more rigorous cases and environments is still required.

Unfortunately, the implementation and study of some of the algorithms occupied more time than what was initially expected, which limited the work in other activities such as, for example, creating an interface that allows the user to choose which documents that he wishes to open when the application starts, as well as studying the utilization of relative paths for the files, GPS information and images in the database, thereby allowing to share the same file system by different machines.

As themes for future work the following subjects are of interest:

- Work with two or more paths, namely analysis and comparison of results.

- Create a new interface with the database that allows the user to choose which documents loaded on previous sessions are to open.
- Study the segmentation of courses and activities by various criteria (slope, pauses, etc.).
- The classification of paths by type of activity, with corresponding change in the values used by the application (minimum speed limit for pauses, for example).
- Include heart rate in pause detection, in particular checking it's behavior when speed decreases significantly.
- Create a formula for calculating the speed threshold for pauses, establishing ranges based on a histogram from previously recorded speed values.

References

[1] C. . I. P. Association, J. Eletronics, and I. T. I. Association. Exchangeable image file format for digital still cameras: Exif version 2.3. *CIPA DC-008-Translation-2012*, Dec. 2012. 1.3, 3.1.1, 3.2.3

[2] D. R. Caldwell. Unlocking the mysteries of the bounding box. `http://www.stonybrook.edu/libmap/coordinates/seriesa/no2/a2.htm`, urldate = 2015-03-20. 3.1.2

[3] S. Consortium. About SQLite. `http://www.sqlite.org/about.html`. 3.1.1

[4] A. Donets. *Using Single Receiver GPS Observations to Analyze the Dynamic Motion of Large Engineering Structures*. PhD thesis, University of Melbourne, 2011. 3.2.3

[5] R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics*. Addison-Wesley, Reading, Massachusetts, 1963. 3.2.1

[6] G. Friedland and R. Sommer. Cybercasing the joint: On the privacy implications of geotagging. In *Proceedings of the Fifth USENIX Workshop on Hot Topics in Security (HotSec 10)*, Washington, D.C., august 2010. 3.2.3

[7] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications, Second Edition*. Artech House mobile communications series. Artech House, 2005. 3.2.3

[8] J. M. B. Lopes. Designing interaction for outdoor sports performance analysis. *SIACG V Ibero-American Symposium in Computer Graphics*, pages 47–50, 2011. 1.1

[9] D. Lynch. *Numerical Partial Differential Equations for Environmental Scientists and Engineers: A First Practical Course*. Springer, 2005. 2, 3

[10] A. Schneider. Geotag. `http://geotag.sourceforge.net/`. 3.2.3

[11] TopoGrafix. Easygps. `http://www.easygps.com/help/geotagging.asp`. 3.2.3

[12] TopoGrafix. GPX 1.1 schema documentation. `http://www.topografix.com/GPX/1/1/`. 3.2.3

[13] F. Yan, Y. X. Gong, and Z. K. Feng. Study on the position drift of gps single point and improving the accuracy of positioning. *Applied Mechanics and Materials*, 333-335:492–498, 2013. 3.2.1