

LVA 126.099 / SS 2010
Research Group Cartography



Location Based Services:

MyTripSystem V2 - Report

Matevž Domajnko
Zbyněk Janoška
Peter Lanz

1. Introduction

'MytripSystem' is a tool for geotagging images using WorldWideWeb environment. It has been created as a semestral project for course 'Location Based Services' at TU Wien. The main idea is to geographically locate photos using GPS track. Typical user is a tourist, who wants to easily locate his photos and share his experience with other people.

The system has been developed at 2009 and further changed and extended at 2010.

User is required to possess GPS device and digital camera. He takes photos along his trip and his GPS device records his route. Both GPS track and photos are uploaded to the MyTripSystem environment and the system combines them (based on their time stamps). Data are visualized using Google Maps API. Visualization using Yahoo! Maps is also possible, although advanced visualization is accessible only with Google Maps API.

Microsoft Visual Studio has been used to program the MyTripSystem environment and Google Maps API (using JavaScript) has been used for visualization.

2. [Geotagging](#)

2.1 [What is geotagging?](#)

Geotagging is the process of adding geographical identification metadata to various media such as photographs, video, websites, or RSS feeds and is a form of geospatial metadata. These data usually consist of latitude and longitude coordinates, though they can also include altitude, bearing, distance, accuracy data, and place names.

(Wikipedia)

2.2 [How to geotag an image?](#)

There are more ways to add a spatial information to the image. Easiest way is to store the location directly in image metadata. Typically, photos are stored in a JPEG format. JPEG metadata are stored in an EXIF (Exchangeable image file format) or XMP (Extensible Metadata Platform). Some cameras already have GPS chips build in and location is stored directly in the EXIF file. Also many types of cell phones include GPS chips and provide some tools to automatically geocode the pictures taken with the cell phone.

However, the most typical situation is, when user owns digital camera without GPS chip and therefore the location must be stored externally, in a GPS track.

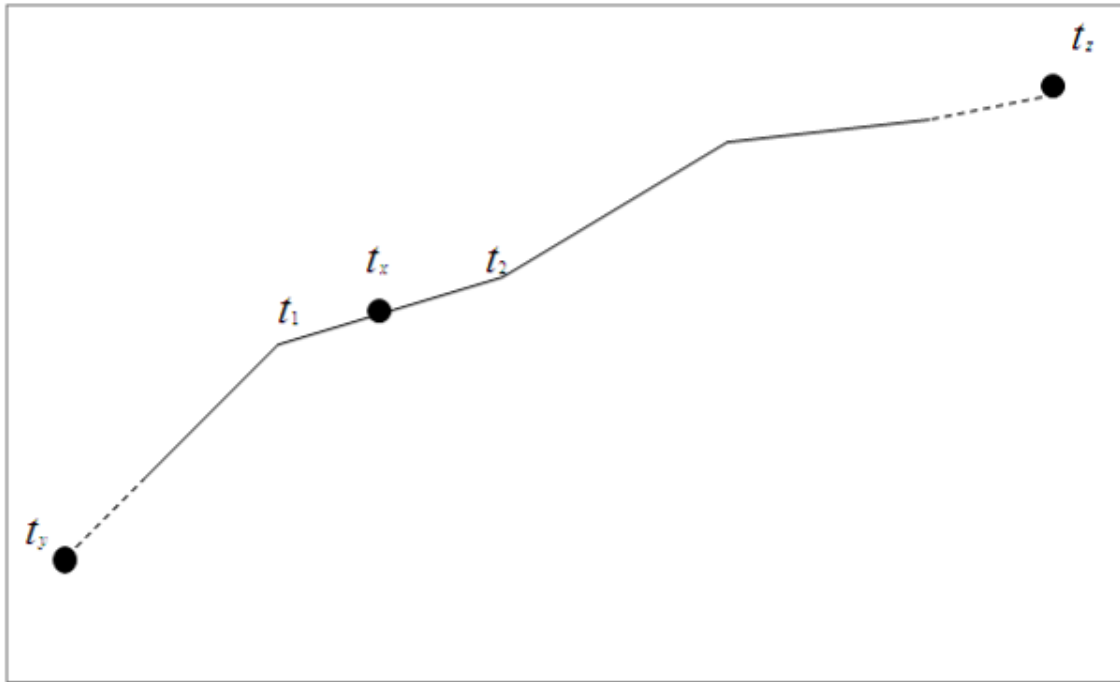
Since every digital camera stores the time, when the picture has been taken, the easiest way to append the location to the image is through the time stamp.

2.3 [How are the images geotagged within this project?](#)

In MyTrip this project, proportion method is used to geotag photos because of not having exact GPS signals at the time of photos.

Lets denote t_x the time, when image has been taken. In the GPS track, we find the points with closest time before t_x and closest time after t_x . For simplification, we suppose that the image has been taken on direct line between the two points. Then, the location of an image is calculated by simple linear interpolation.

Formulas and representative picture is given in the following figures:



$$\text{Formula.1: } \frac{t_x - t_1}{t_2 - t_1} = \frac{\text{lat } t_x - \text{lat } t_1}{\text{lat } t_2 - \text{lat } t_1}, \quad \text{Formula.2: } \frac{t_x - t_1}{t_2 - t_1} = \frac{\text{long } t_x - \text{long } t_1}{\text{long } t_2 - \text{long } t_1}$$

t_x : Time of the photo

t_1, t_2 : Times of received GPS signals

t_y, t_z : Times of photos which are not taken during the GPS track

$\text{lat } t_x, \text{long } t_x$: Coordinates of the photo

$\text{lat } t_1, \text{long } t_1, \text{lat } t_2, \text{long } t_2$: Coordinates of GPS signals

Figure 1: Basic concept of geotagging photos

In case of having photos that are not taken during recording the GPS track but some time close to it, it is also possible to see them in final maps. If time difference between these photos and start/end time of GPS track is less than 10 minutes, program represents them at start/end points of track. If time difference is more than 10 minutes, program alerts user as it can't find the location of these photos.

3. Software

3.1 ASP.NET

MyTripSystem has been programmed within the Microsoft Visual Studio development environment, using the functionality of ASP.NET and C#.

ASP.NET (successor of Active Server Pages technology) is a software framework enabling user to design and develop dynamic websites. Microsoft Visual Studio is a development environment, which had been used for building the application.

Microsoft Visual Studio Express is a set of freeware integrated development environments (IDE) developed by Microsoft that are lightweight versions of the Microsoft Visual Studio product line. The idea of Express Tools editions, according to Microsoft, is to provide streamlined, easy-to-use and easy-to-learn IDEs for users other than professional software developers, such as hobbyists and students.

With this development environment we expanded last year's application. We used existent framework and added new functionalities mostly in programming language C#.

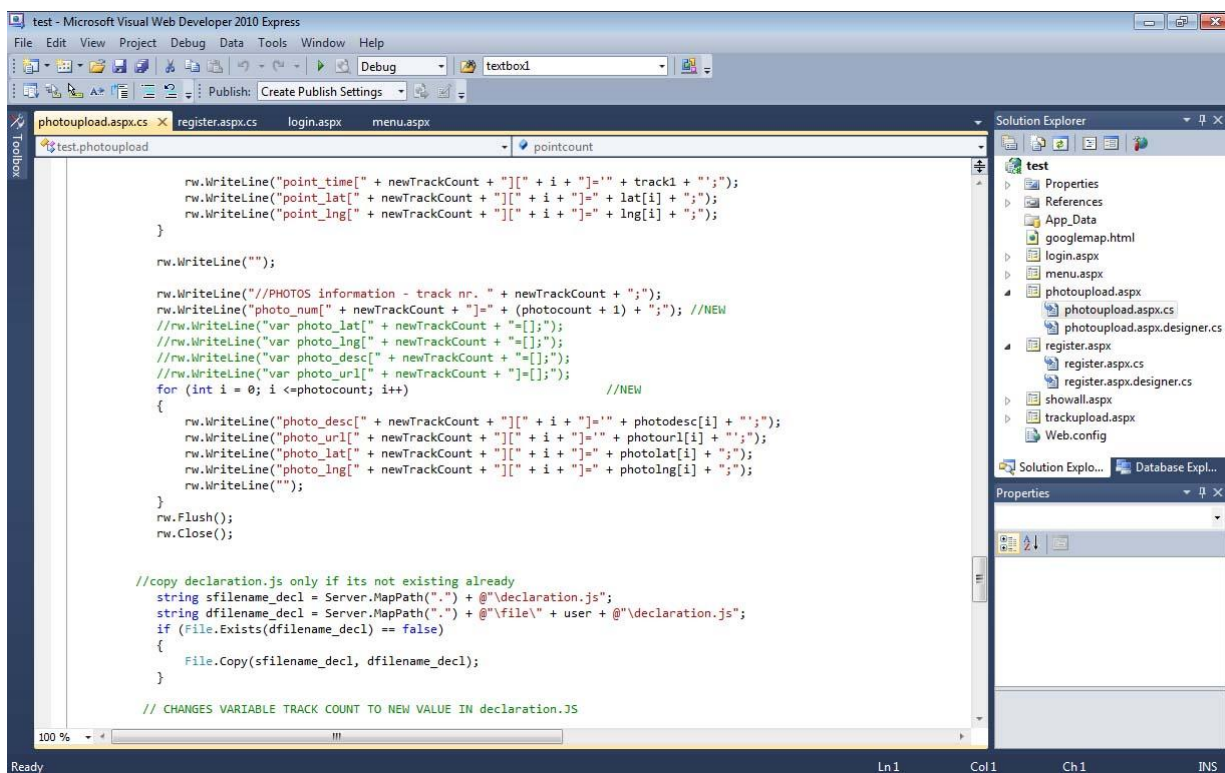


Figure 2: Microsoft Visual Studio development environment

3.2 Google Maps API

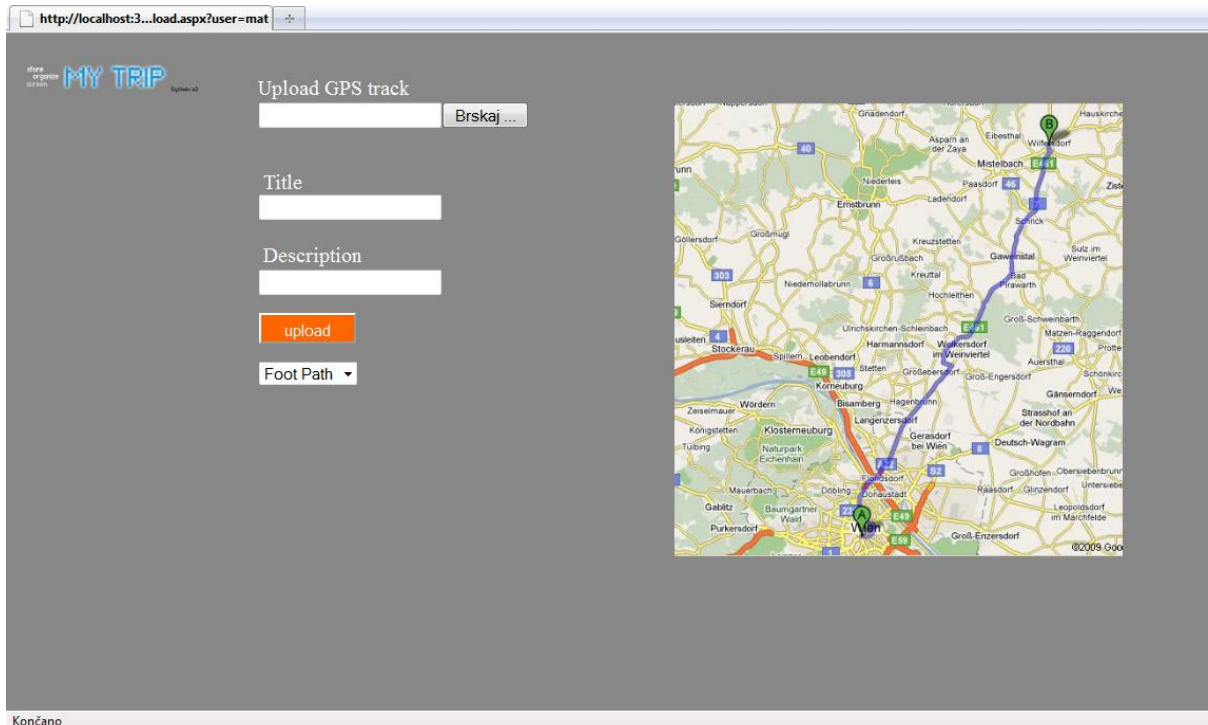
Google Maps API enables user to contain the whole Google Maps site into an external website. User can also add his own data points. JavaScript is used for calling the functionality of Google Maps and for user customization.

4. Implementation of new functionalities

4.1 Grouping of Tracks

All tracks uploaded by the user are stored in a database. To allow the user to better organize those tracks, we decided to implement functionality in the original MyTripSystem which enables every user to define the “type” of each track.

For this, we added a drop down menu to the form “track upload” which can now declare each uploaded track being either a “foot path”, a “bike tour” or of “other” kind (Figure 3).



The screenshot shows a web browser window with the URL `http://localhost:3...load.aspx?user=mat`. The page title is "MY TRIP". The main content area is titled "Upload GPS track". It contains a file upload field with the text "Brskaj ...", a "Title" input field, a "Description" input field, an orange "upload" button, and a dropdown menu currently set to "Foot Path". To the right of the form is a Google Map of the Vienna region, showing a blue track overlaid on the map. The map includes labels for various locations like Mistelbach, Gaweinstal, and Wien.

Figure 3: Form “track upload” inclusive newly added track type option.

Correspondingly, the database file (“mydb.mdb”) received some changing: an additional column called “type” was added to the table listing all tracks. Those preliminary steps were essential to create an environment in which the system can access and work with tracks according to their type. Having done this, the next project goal was possible to accomplish:

4.2 Visualization of all user’s tracks in one map, organized into groups of track type

Last year’s system worked in that way that for every track it created a separate folder containing one file “googlemap.html” and one file “data.js”. The data.js file provides information about coordinates and timestamp of photos and the track; the googlemap.html file reads this information and visualizes the track using Google Map API interface and functionalities. We adopted described workflow of these two files into our system but with an important change: now, for each user a folder is created this specific user. Additionally, a file with variable definitions was added (“declaration.js”), which acts as a second source for the html file.

These changes in application structure were accompanied with changes in the html forms. As a

consequence of creating the option to visualize all users' tracks in one map, the option to show just one track in a map was removed. Moreover, the user is navigate back to main menu after having added a track in case the user wants to add more tracks in one session. Nonetheless major part of changes in code was necessary in the googlemap.html, where a loop was established to load one track inclusive photos after the other. Having done this, the interface of the track visualization could be edited:

4.3 Choosing groups of tracks by type to visualize

Checkboxes above the Google Map are offering the user now to switch ON or OFF tracks belonging to one of the three possible types of tracks (Figure 4). The visualization was further enriched with a logo and formatting issues, respecting corporate design considerations. At the top, the user is welcomed with his username, below an overview about track types and actual uploaded user tracks is given.

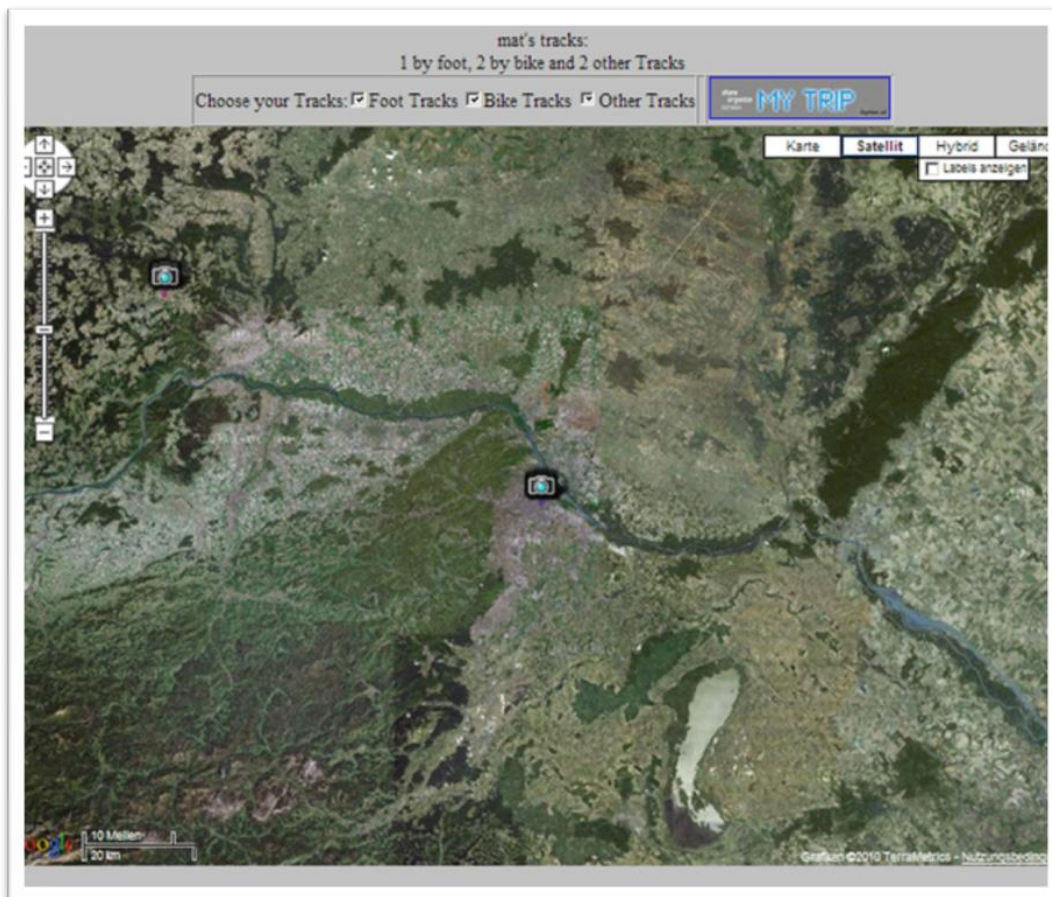


Figure 4: Presentation of tracks using Google Maps API.

Tracks are drawn in three different colors, corresponding to their type. After putting together the application so far, we saw the next goal in finding and integration an algorithm that smoothes uploaded tracks in order to improve presentation's appearance.

4.4 Track smoothing algorithm



Figure 5: Tracks with smoothing algorithm (right) and without (left).

Resumed shortly, our track smoothing algorithm looks for the angle between three consequent points of the track and calculates the distance between every consequent two points. If the angle is above a defined value (default set to 30 degrees) and the distance is lower than a predefined value, the application leaves out this point and tries to continues with the next one.

This smoothing algorithm is able to reduce the 'hard error', when one of the points is too away (caused mostly by technique) and also when there is cluster of points (user standing too long at one place), these points will be reduced to only one. This algorithm does not do cartographic generalization, it is only meant to reduce possible errors.

The results compared with the initial track appearance can be seen in Figure 5. Having accomplished these technical goals, the design issue was attempted.

4.5 Design considerations

Improvements in application's appearance and design were an important pre-defined goal as well having in mind two considerations: first, an overall improvement of user's experience, and second enhancements in usability and application's user friendliness.

First goal was tried to reach with elaborated design and animation gadgets (e.g. Flash...), second goal we chased by creating a size-variable interface. Moreover, colors and building blocks were chosen and assembled in a concerted way. Last, we came together to scale down the interface size since is the whole screen area is never needed due to simple and not overloaded forms.

Due to limited resources all those design consideration could not be realized by far. As an example we would like to point out the "login" form which received most design-related treatment (Figure 6).

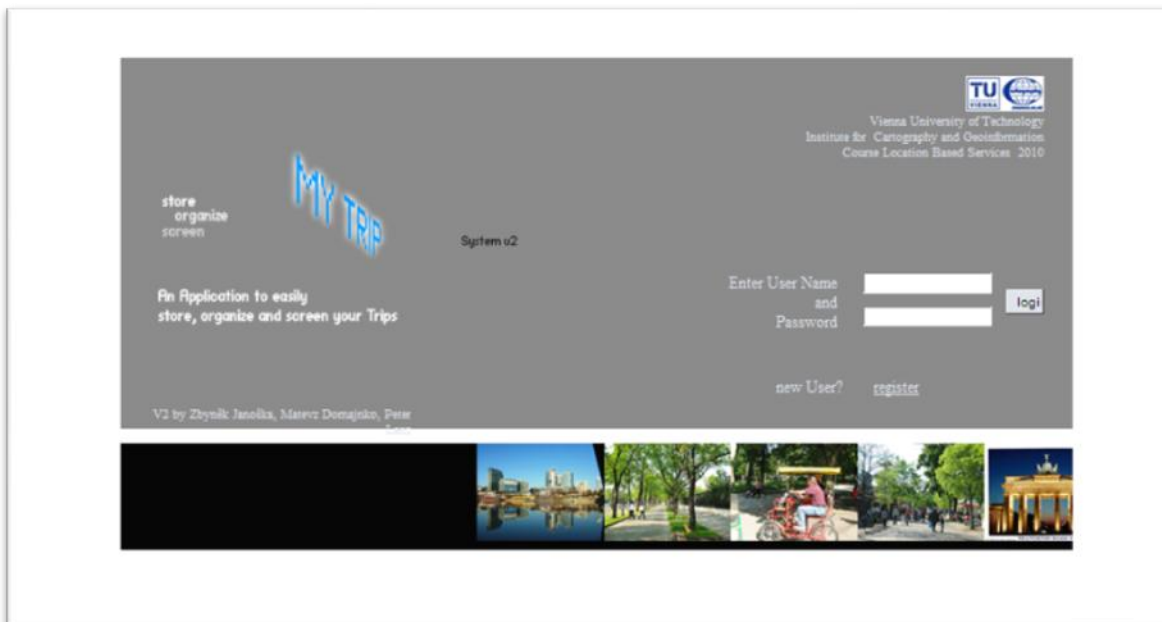


Figure 6: Form "login" showing new design.

5. Conclusion

With increasing number of people using internet not only for finding information, but also for expressing themselves, sharing, and creating their own web content, geographical information systems bring a great possibilities. MyTripSystem is a tool, which enables user to geotag photos to geographical location in a very easy and user-friendly way. This solution is to be user primary by tourist, visitors etc., but can be easily used for example by municipality for presentation of a region. MyTripSystem provides a service to share personal information using maps and an opportunity for communication using World Wide Web environment.

6. References

- **ASP.net:** <http://www.asp.net> (retrieved on 2009-04-27)
- **Google Maps API:** <http://code.google.com/intl/en/apis/maps/> (retrieved on 2009-06-27)