Mobile Guide Applications Using Representative Visualizations

Omar Choudary
UPB/IRIT, Univ. of Toulouse
2 Rue Camichel, 31071
Toulouse, France
choudary.omar@gmail.com

Vincent Charvillat
IRIT, Univ. of Toulouse
2 Rue Camichel, 31071
Toulouse, France
charvi@enseeiht.fr

Romulus Grigoras
IRIT, Univ. of Toulouse
2 Rue Camichel, 31071
Toulouse, France
grig@enseeiht.fr

ABSTRACT

Recent developments of hardware capabilities on mobile devices have made Location Based Services(LBS) very popular. This triggered an increasing need of rich visual content in mobile guide applications.

This paper presents two original approaches for using representative visualizations: artistic views which represent an arbitrary deformation made by an artist and perspective views(3D-like) obtained from 3D models. Both approaches are based on learning GPS-to-image relations. We show an efficient use of the *thin-plate spline* for registering GPS coordinates with images. We also show the implementation of our guiding system on two mobile platforms.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Enhancement—*Registration*

General Terms

Algorithms, Design, Performance

Keywords

mobile guide, LBS, visual representation, artistic views

1. INTRODUCTION

The past decade has seen a massive growth in the number and the capabilities of mobile devices (PDAs, smart-phones etc.) and associated applications such as mobile guides and LBS [5].These applications offer navigation support and information delivery services, as users want to know where they are, what is interesting near them and they want to have this information whenever they need it.

Today's mobile devices have WLAN, GPS and also some graphic acceleration. These new communication and positioning capabilities have fueled a tremendous interest around LBS. LBS have made mobile guide applications more attractive as they provide accurate location as well as surrounding information (e.g. shops or hotels within close distance).

Providing users of mobile guides with intuitive, easy ways of understanding their position in the environment is very important. They need to mentally map what they see in the application with the reality. Traditional mobile guides such as *Google Maps* only show simple 2D maps with street names without any visual help of the environment. This can make the process of finding a specific place difficult.

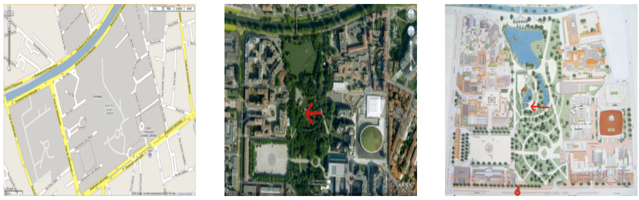
It is clear that users would benefit from visual content richer than 2D maps. For this purpose research has been conducted on 3D mobile guide applications [7]. They show a 3D model that tries to match the real environment as closely as possible. By combining LBS with 3D models these applications aim at matching user position in the real world with camera position in the model, giving the sensation of being inside the application in the same place as in reality. The problem with 3D applications resides on current hardware technology. Currently only a few mobile devices have graphic acceleration and most of them have small memory. 3D mobile applications have reported small rendering rates (less than 1 fps) even if optimizations and offline computations were used. We argue that, within a few years, as hardware on mobile devices will improve, these rich visual applications will become common place.

In between 2D and 3D one can identify intermediate approaches where simple photos or Google Earth images are used in mobile guide applications. Unfortunately photos are not good enough to be used as maps in applications. For example, in a top view of a park (Figure 1) trees cover paths affecting user's ability of finding his position on the image.

In the following section we propose two new intermediate approaches for mobile guide applications using representative views of the environment. In the first approach we use artistic images to represent outdoor environments(e.g. parks) and in the second approach we use perspective images, taken from 3D models, that give the same visual sensation as 3D models and fit perfectly in small areas or venues(e.g. stadiums) (see Figure 3).

2. VISUAL APPROACHES

In mobile guide applications one of the most important problems is how to display the environment and the current position to the user. In this section we present two different and innovative approaches to display the environment and we explain the methods used to convert from GPS to



(a) (b) (c)

Figure 1: ***Difference between 2D map(a), aerial photo(b) and artistic view(c) of Compans, Toulouse***

pixels coordinates in order to accurately show the current user position in the image. In the rest of the document we'll consider that GPS coordinates are expressed in latitude(φ), longitude(λ) and altitude(h) using the WGS84 datum.

2.1 Artistic Views

Artistic views are a good choice for outdoor environments(e.g. parks, town centers) where an overview of a large area is needed. As it can be seen in figure 1, the environment and interesting places are easier to identify than in simple photos or 2D maps. In the artistic view the bridge (denoted by the red arrow) is clearly seen whereas in the aerial photo it is not visible and in the 2D map it doesn't exist. Most notable elements in the area are represented in an artistic way that is very easy to understand by human brain and is closest to our mental representation of space than simple images.

In this work we use *Radial Basis Mappings* (RBM), to register the GPS coordinates with the artistic images. Since the images to register are hand drawn, the deformations between the artistic view and the GPS coordinates can be only modeled by non-rigid transformations. A similar approach has been applied to a different application in medicine [2].

A $\mathbb{R}^2 \rightarrow \mathbb{R}$ Radial Basis Function (RBF) f is defined by a basis function ϕ and an $l+3$-vector of coefficients $(w_1, ..., w_l, a, b, c)$ and a set of l centres $\mathbf{q_k}$ as :

$$f(\mathbf{x}) = a\varphi + b\lambda + c + \sum_{k=1}^{l} w_k \phi(\|\mathbf{x} - \mathbf{q_k}\|) \quad (1)$$

where $\mathbf{x} = (\varphi, \lambda)^T$. It consists of a linear part parameterized by (a, b, c) and a non-rigid part, a sum of l weighted terms with coefficients w_k of the basis function applied to the distance of $\mathbf{x}$ and the centres $\mathbf{q_k}$. A possible choice for the basis function is the Thin-Plate Spline $\phi(\eta) = \eta^2 log(\eta)$.

We can then construct a $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ RBM m mapping a GPS point $\mathbf{x}$ to a point $m(\mathbf{x})$ defined by two RBF f^x and f^y sharing their centres :

$$m(\mathbf{x}) = \begin{pmatrix} f^x(\mathbf{x}) \\ f^y(\mathbf{x}) \end{pmatrix} = \bar{\mathsf{A}}\mathbf{x} + \mathbf{t} + \sum_{k=1}^{l} \begin{pmatrix} w_k^x \\ w_k^y \end{pmatrix} \phi(\|\mathbf{x} - \mathbf{q_k}\|) \quad (2)$$

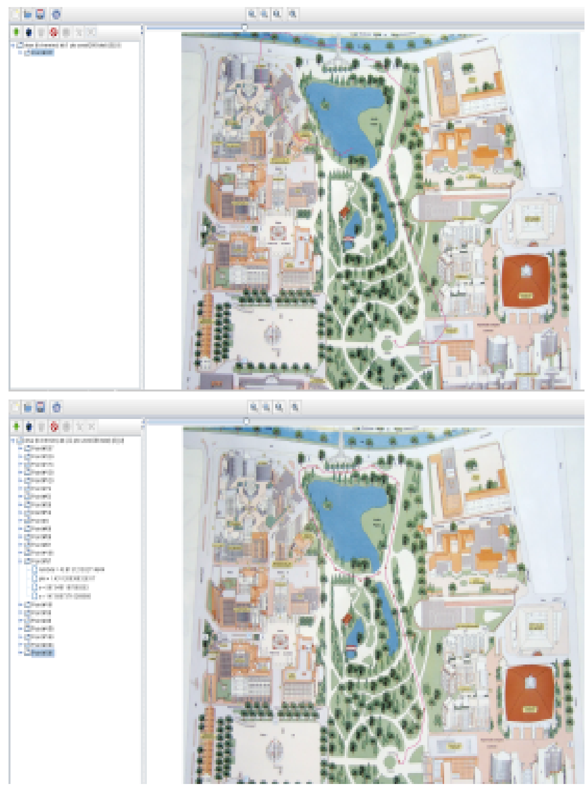
where $\mathsf{A}_{2\times 3} = (\bar{\mathsf{A}}_{2\times 2} \;\; \mathbf{t})$ defines an affine transformation represented by 2 rows of 3 parameters (a^x, b^x, c^x), (a^y, b^y, c^y) by generalization of equation 1.

We can estimate a RBM from n matched features (or manual landmarks) : $\tilde{\mathbf{x_i}} \leftrightarrow \tilde{\mathbf{x_i'}}$ where $\tilde{\mathbf{x_i}}$ (resp. $\tilde{\mathbf{x_i'}}$) stands for GPS (resp. image) coordinates. A center is chosen at each $\tilde{\mathbf{x_i}}$.

In order to compute the parameters needed in equation 2 we have created a helper application(see Figure 2). In this application we can load the artistic image and a set of GPS coordinates (see the unregistered traces in Figure 2 (a)). The application allows to drag and drop points of GPS traces at their correct places onto the artistic view. At least 3 points must be matched for estimating the affine part of equation 2) and then the application automatically re-estimates the non-rigid RBM mapping each time a new correspondence is established. Using our application we can both visualize the current mapping and assess its generalization capability. After we have matched enough points we can obtain the parameters that we need from the application and use them in the mobile application to transform GPS coordinates into pixel coordinates by applying the equation 2. The estimation process is simple. Given the correspondences and the selected deformation centres, we minimize a least-squares transfer error J :

$$J = \frac{1}{n} \sum_{i=1}^{n} \|m(\tilde{\mathbf{x_i}}) - \tilde{\mathbf{x_i'}}\|^2. \quad (3)$$

We must also ensure the boundary conditions [1] and get a linear least squares problem with linear equality constraints.



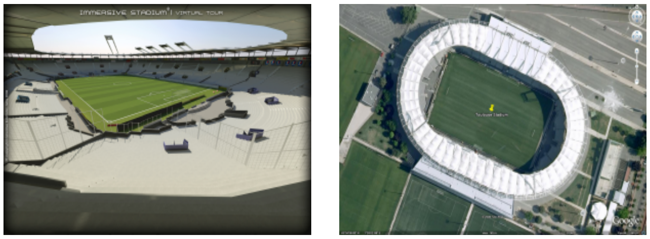
(a)

(b)

Figure 2: ***Application used to compute TPS parameters, before(a) and after(b) matching the spline (in red) with the image***

2.2 Perspective Views

Perspective views can be the ideal choice to visualize interior spaces or venues (e.g. stadiums). We used this approach in our test application for guiding inside TFC football stadium. This section presents the use of perspective images taken from 3D models (Figure 3) in mobile guide applications. One can easily notice that perspective images offer the same picture to the user as a 3D application while using



(a) (b)

Figure 3: *Perspective image(a) and aerial photo(b) of TFC stadium, Toulouse*

a simple image instead of a complex 3D model. The immersion of the user into the venue (the stadium in our case) is possible via zoom in/out and scroll left/right and top/down. We don't focus on the positioning accuracy since we consider this to be a separate problem. For our experiments GPS positioning has been enough, but if higher accuracy is needed, WLAN or other type of positioning can also be used [8].

A perspective image can be created from a 3D model by setting a camera at a specific position. After the perspective image has been created we can obtain the parameters of the camera (direction, horizontal and vertical FOV) that we'll need for the conversion of coordinates. The conversion between GPS and pixel coordinates is done in 2 main steps, firstly we convert a GPS coordinate $W(\varphi, \lambda, h)$ to a vector coordinate $\mathbf{V}(x, y, z)$ in the 3D model and then we convert $\mathbf{V}$ to a pixel coordinate $P(u, v)$.

To convert from GPS to the 3D model coordinates we first convert $W(\varphi, \lambda, h)$ to Earth Centred Earth Fixed (ECEF) coordinates and then to local East, North, Up (ENU) coordinates that match a plane tangent to Earth's surface.This conversions are described in [9].

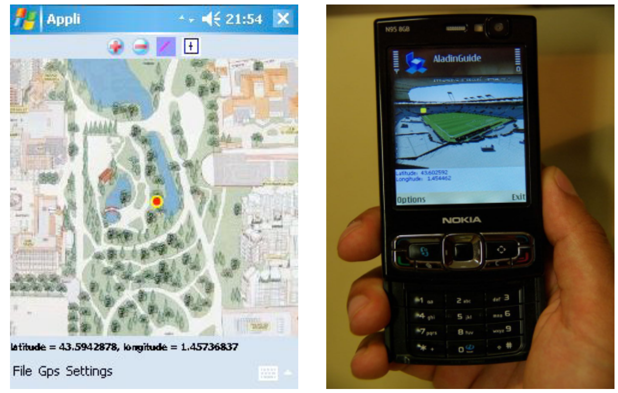
After obtaining (x, y, z) in the ENU system we have to apply a rotation in order to match the direction of the Y axis of the 3D model with the North(N) axis of the ENU system.This is a simple rotation applied only to the x and y values. The angle of rotation can be given by the 3D model or it can be determined if we know 2 points along the Y axis in the 3D model. We also have to apply a scale transformation in order to adjust real dimensions to the 3D model size. As with the angle, the scale factor can be given by the 3D model or can be obtained if 2 points are known(we compare the distance between them with the real distance).

Having the coordinates $\mathbf{V}(x, y, z)$ in the real model we obtain pixel coordinates $P(u, v)$ using a perspective projection:

$$[su, sv, s]^T = \mathsf{P}[x, y, z, 1]^T \qquad (4)$$

where s is a non-zero scale factor. The projection matrix P can be constructed [6] from the camera parameters *center* $C(X_C, Y_C, Z_C)$, *target* $T(X_T, Y_T, Z_T)$, horizontal field of view(FOV) f_X, vertical FOV f_Y, image width W, height H.

If camera parameters are unknown, a projective transformation using a set of correspondence points between the 3D model and the perspective image may provide a P estimate. Moreover, if neither the camera parameters nor the 3D model are known (if for example we take an arbitrary image from a website) we can still obtain the transformation between $W(\varphi, \lambda, h)$ and $P(u, v)$ using a similar approach to that described in the previous section. We only need a set of correspondences between GPS and pixel coordinates. Depending on the number of correspondences used the accuracy of results may be different. We strongly encourage the use of the first method if the 3D model is known.



(a) (b)

Figure 4: *User Interface of implementations, artistic (a) and perspective views(b)*

3. IMPLEMENTATION AND RESULTS

In this section we explain the implementation of the applications used to test the two approaches and we also show the obtained results. Both approaches have been tested on mobile devices and we observed a real-time execution, without noticeable delays in position update. Applications have been running for hours, the main limitation being battery life caused by the GPS receiver and the computations.

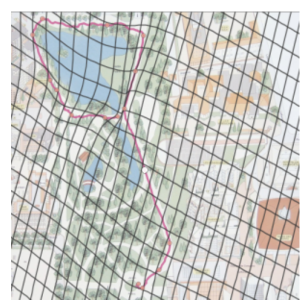


Figure 5: *Grid deformation for artistic views*

3.1 Artistic View application

To test the artistic image approach we created two applications, an offline Java application used to compute the RBM parameters and an online application running on the mobile device (see Figure 4). For portability reasons, the online application runs in JavaMe on a HP iPAQ 6915.

The APIs used for the online application are part of *JSR 216 Personal Profile* and *JScience* for the elastic model. Offline application uses the Lapack++ library(C++) for the mathematics computations and we integrated it into the Java application using JNI.

The online application communicates over a serial port with a GPS receiver and parses the NMEA frames. The offline application(see Figure 2) computes the parameters for

Table 1: Perspective View Results

Number of Points	20
Absolute Error in pixels (16 pixels/meter)	2
Relative Error	0.1%

the RBM and saves them to a file that is loaded into the mobile application to make the conversions between GPS and pixel coordinates in the artistic image. The offline application computes RBM parameters instantly for 100 control points(correspondences) and it takes 50 ms for 500 points.

On Figure 5 we see the result of a RBM which maps GPS coordinates onto the artistic view. The transformation is estimated from the correspondences shown on Figure 2. The resulting parameters are used to deform a grid of iso-latitude and iso-longitude lines. One can notice that a non-rigid mapping is indeed necessary to cope with the artistic deformations. To evaluate how well the RBM method performed for artistic views, a registration error (obtained from manual landmark selection) has been computed. These errors are below 3 pixels in average and confirm good visual results for a 240x240 view. Moreover, the figure clearly shows that a regularization of the transformation could naturally smooth the transfer of unstable GPS coordinates. It is well known that shrinking the non-rigid coefficients is very simple by ridge regression of linear TPS [3].

3.2 Perspective View application

Testing of the perspective view was done on a Nokia N95 device which has an internal A-GPS receiver able to increase GPS TTFF (time to first fix) as it uses the GSM network to obtain missing satellite information. We developed the mobile application in native Symbian C++ (see Figure 4) as it provides a rich set of APIs, applications run faster than Java or Python applications and are memory efficient. We chose Symbian C++ also because we wanted to use the *WLAN Info API* to get location information based on wireless networks even if at the end we didn't use this information for our experiments.

The application is composed of 3 modules:

- User Interface: shows the perspective image to the user and its position. This module also handles user input in order to change the view (multiple views can be used), zoom in/out and move through the image (left, right, top, bottom)
- Map Engine: makes the conversions from GPS to pixel coordinates for each particular view as previously described
- Position Engine: gives the current position. The position is obtained using the *Location API* as latitude, longitude(decimal degree representation) and altitude (meters)

We have carried out field tests at the TFC stadium in Toulouse. The 3D model and perspective views were provided by Immersive Solutions [4]. To apply the conversions described in the second section we needed two known points from the 3D model (to compute the rotation angle and the scale factor) and the camera parameters for each perspective view to create the projection matrix. These parameters can be easily found from the 3D model.

As with artistic images, this application showed good results. We used perspective views as well as a top view image of the 3D model to compare positions. We observed the position in the same place in both top and perspective views and this position also matched our real position in the stadium. We used a Matlab script in order to test the accuracy of our method on different GPS coordinates of the stadium. This allowed us to filter out the main source of data input noise: coordinate fluctuations of the GPS receiver and also the poor floating-point precision of N95. Table 1 shows the absolute and relative errors between pixel coordinates obtained with our method and expected pixel coordinates. These results prove that the method is very accurate.

4. CONCLUSION

This paper has presented 2 new approaches for mobile guide applications to visualize the environment and user position, in both outdoor environments and venues. We have showed a practical method for positioning the user inside images that are more convenient and familiar than traditional maps. We have explained our methods for converting GPS to pixel coordinates for accurate positioning. TPS has been successfully applied to match GPS coordinates with artistic images and more generally to images produced with non-rigid transformations.

Our solution provides a lightweight yet intuitive visual guiding until the majority of mobile devices will be capable of handling high-quality 3D models. Nevertheless, the methods presented will continue to be useful for accurate positioning of the camera inside the 3D scene. In our future work we intend to improve accuracy by WLAN positioning. We could also better manage the collection of views stored on the mobile. For example, a minimal set of views could be complemented by downloading new views when the user steps outside the initial area. It would also be possible to automatically switch between artistic and perspective views.

5. REFERENCES

[1] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transaction on PAMI*, 11(6), 1989.

[2] M. Davis. A physics-based coordinate transformation for 3-d image matching. *IEEE Transactions on Medical Imaging*, 16(3):317–328, 1997.

[3] T. Hastie and Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[4] Immersive-Solutions. `www.immersive-solutions.com`.

[5] I. A. Junglas and R. T. Watson. Location-based services. *Commun. ACM*, 51(3):65–69, 2008.

[6] F. Nielsen. *Visual Computing: Geometry, Graphics, and Vision*. Charles River Media, 2005.

[7] A. Nurminen. m-loma - a mobile 3d city map. In *Web3D '06: Proc. of 11th Intl. Conf. on 3D web technology*, pages 7–18, New York, USA, 2006.

[8] J. Wierenga and P. Komisarczuk. Simple: developing a lbs positioning solution. In *MUM '05: Proc. Intl. Conf. on Mobile and ubiquitous multimedia*, pages 48–55, New York, NY, USA, 2005. ACM.

[9] J. Zhu. Conversion of earth-centered earth-fixed coordinates to geodetic coordinates. *Aerospace and Electronic Syst., IEEE Trans. on*, 30(3), 1994.