

# Heterogeneous information integration for touristic augmented reality mobile apps

Darian Frajberg, Piero Fraternali, Rocio Nahime Torres

Politecnico di Milano,  
Dipartimento di Elettronica, Informazione e Bioingegneria  
Piazza Leonardo da Vinci, 32, Milan, Italy  
`{first.last}@polimi.it`

**Abstract.** Mobile Augmented Reality (AR) apps promote a new way of marketing the touristic offer of a territory, by overlaying useful information directly on top of what the user sees. These apps analyze the sensor readings (GPS position and phone orientation) and possibly also the camera view, to understand what the user is watching and enrich the view with contextual information to enable knowledge acquisition and exploration. Developing mobile AR apps poses several challenges related to the acquisition, selection, transmission and display of information, which get more demanding in mountain applications where usage without Internet connectivity is a strong requirement. This paper discusses the experience of a real world mobile AR application for mountain tourists, from its design to the feedback from the community of users.

## 1 Introduction

Mobile outdoor Augmented Reality (AR) applications are an emerging category of solutions implemented in mobile terminals (especially mobile phones, but also envisioned for the forthcoming consumer-grade smart glasses) that enrich an outdoor experience (e.g., trekking or star gazing) by overlaying useful information onto the device camera view. A prominent class of such applications has been published for mountain tourism (e.g., PeakLens<sup>1</sup>, PeakAR<sup>2</sup>, etc). These applications permit the user to receive information directly onto the objects in view, especially mountain peaks, but also about other points of interest, such as shelters, trekking paths, and more. Mobile outdoor AR applications pose several technical challenges: 1) they must understand the current context and activity of the user, e.g., what he is looking at; 2) they must find the appropriate information pertinent to the user's current activity and view; 3) they must overlay the retrieved information onto the device screen in a way that is adequate to the user's experience. Besides these requirements, also non functional properties are relevant: the application must work in real-time, on a device that normally has low computation power and strict energy consumption constraints, and, especially in the case of outdoor apps, functioning must be guaranteed also in

---

<sup>1</sup> <http://www.peaklens.com>

<sup>2</sup> <http://peakar.salzburgresearch.at>

absence of Internet connectivity. Taken together, all such requirements make the development of outdoor mobile AR apps a non trivial task.

In this paper we report the ongoing experience with PeakLens [5], an outdoor mobile app for mountain lovers that features a computer vision module based on Deep Learning for the extraction of the natural mountain skyline from the camera view of the mobile phone; such skyline is aligned with the skyline extracted from a virtual panorama, computed from the GPS position of the user, the orientation of the mobile phone view, and a Digital Elevation Model (DEM) of the Earth. To enrich the user’s experience, the app harvests the coordinates and information of peaks from multiple data sources, determines which peaks are in view in the current situation, projects the 3D coordinates of the visible peaks onto the 2D view of the mobile phone, and overlays the peak names and other meta-data (e.g. altitude, distance) onto the camera view in the proper screen positions.

Specifically, this paper focuses on the classes of information that can be used to augment the camera view and on the problems posed by their management. The paper is organized as follows: Section 2 surveys the related work; Section 3 illustrates the architecture and components of a tourism mobile AR app for mountain peak identification; Section 4 discusses the classes of information relevant for tourism mobile AR apps and their underlying management problems; Section 5 briefly reports the experience and feedback gathered from the publication of PeakLens; finally, Section 6 concludes and gives an outlook on the future work.

## 2 Related work

**Augmented reality applications.** AR is a well established research area in the Human-Computer Interaction field, which has recently gained momentum due to the introduction by major hardware vendors of consumer-grade AR devices. Furthermore, a recent trend shows mobile devices being used as low cost AR platforms, thanks to the improved standardization (most AR software can now be used without ad hoc hardware), increased computational power and sensor precision [6]. The survey in [1] recaps the history of AR, introduces the essential definitions of the discipline, and positions it among other related technologies. The authors also propose design guidelines and examples of successful AR applications and highlight directions of future research. An important branch of the discipline is outdoor AR mobile apps development. Several works address the problem, usually to identify [4] and track [8] points of interest in urban scenarios. Although mobile AR solutions already exist (e.g. Wikitude<sup>3</sup>), they generally rely only on compass sensors or on the a priori known appearance of certain objects.

**Mountain peaks applications.** Nowadays, numerous mobile applications have the goal of identifying mountain peaks with their corresponding information such

---

<sup>3</sup> <http://www.wikitude.com/app>

as name, altitude and distance, and attract millions of people around the world who are devoted to mountain activities. Considering the fact that these applications are meant to be used in mountainous areas where Internet data access is not always granted, off-line maps download management is an essential feature. Some of the best-known apps are PeakFinder<sup>4</sup> and PeakAR. PeakFinder uses the GPS position to compute from the DEM and visualize on the screen a virtual panorama with the mountain peaks positioned on it. It also exploits the compass and the users orientation, but does not analyze the real images and thus cannot overlay any information on the camera view. Regarding the management of off-line maps, it automatically downloads the required area when the application is started based on the current position (e.g. the entire Alps). PeakAR uses the camera of the device, as well as other sensors, for the projection of the peaks on the screen. However, its implementation does not perform any geometrical projection of the terrain before positioning the peaks on the screen, which means that also peaks that are masked by the terrain configuration, and thus are invisible, are still shown. The problem is somehow alleviated by a simple filter that disables the display of peaks that are beyond a distance threshold. PeakAR manages off-line areas by default, because it requires downloading all the peaks around the world beforehand. This is possible because only the 3D coordinates of the peaks are downloaded, and not the surrounding DEM points. Several other apps, similar to the above mentioned ones, use only the position and orientation sensors, which are imprecise and may induce substantial peak positioning errors. Some clear examples are ViewRanger<sup>5</sup> and PeakVisor<sup>6</sup>. ViewRanger targets trekkers and offers route guides and GPS navigation; recently it incorporated an AR function, which overlays points of interest such as peaks, towns, lakes, cliffs and glaciers, over the camera view. Positioning uses only the GPS and orientation sensors. PeakVisor allows the user to correct compass errors manually, by registering the virtual panorama with the real image captured by camera, using the sun position as a hint. Again, sensor errors may intervene: e.g., the DEM resolution is such that not always the virtual panorama generated from it matches well the camera image, which makes manual adjustment hard.

In contrast with all the previously mentioned apps, PeakLens [5] aims at providing high precision peak identification and information overlay by using artificial intelligence to analyze the frames captured by the device camera and accurately position the points of interest. Its core is the comparison of what the user sees with the 3D model of the terrain and the automatic alignment of the virtual and real mountain skylines, whereby the app can correct significant sensor errors. Moreover, PeakLens can also handle occlusions, in order to avoid displaying peaks when they are masked by an object in front of them (e.g., a bell tower, a person, or a tree).

---

<sup>4</sup> <http://www.peakfinder.org/mobile>

<sup>5</sup> <http://www.viewranger.com/skyline>

<sup>6</sup> <http://peakvisor.com>

### 3 Outdoor augmented reality for mountain tourism

Mobile AR applications dynamically augment the sequence of the device camera frames, possibly using the position and field of view determined from the GPS and orientation sensors, to superimpose information about objects present in a region of interest. They compute the on-screen position of relevant objects and associate relevant meta-data to them, at the same frequency of the input capture. The main challenges for mobile AR applications development to provide a satisfactory user experience include achieving high object positioning accuracy, fast response time, and low memory and battery consumption. In particular, applications for outdoor usage, such as mountain tourism ones, must cope with other requirements:

- *Uncontrolled viewing conditions*: the objects to be identified have no fixed, known a priori, appearance, because the viewing conditions can change due to weather, illumination, occlusions, etc.
- *Uncertain positioning*: position and orientation sensor errors make the location and field of view estimation noisy; thus the identification of the relevant objects from these signals alone cannot be assumed to be fully reliable and must be corrected with information from the camera view.
- *Bi-dimensional reduction*: although the objects' position in the real world is estimated in the 3D space, the on-screen rendition requires a projection onto the 2D surface of the camera view, based on the properties of the camera.
- *Uncertain Internet connection*: especially for rural and mountain regions.



Fig. 1: PeakLens interface: compass orientation (bottom band); photo shooting command (upper button); peak scrolling and peak page indicator (bottom button and 3 circles).

A generic architecture for outdoor mobile AR applications has already been proposed in [5] and applied to the implementation of the PeakLens mountain tourism app (Figure 1); it is based on four main sub-systems: 1) the *Sensor Manager* orchestrates concurrent signal acquisition from the camera, the orientation sensors, and the GPS. 2) The *Data Manager* provides the initial positions of the points of interest to display and the meta-data for enriching them. 3) The *Position Alignment Manager* updates the positions of the objects adapting them to the current camera view, tracking the phone movements and reacting to sensor updates, e.g., more precise GPS fixes. The points of interest are projected in real-time by computing the 2D positions from orientation sensor readings, while simultaneously object 3D positions are updated in background based on the mountain skyline extracted from the camera frames, the virtual skyline created from the DEM, and the alignment between the mountain and virtual skylines. Besides, the algorithm detects occlusions produced by irrelevant objects (such as trees, cables, people, roofs, etc) avoiding to visualize masked points of interest. 4) The *GUI* draws objects and their meta-data at the computed on-screen coordinates, at the same frequency in which the frames are captured.

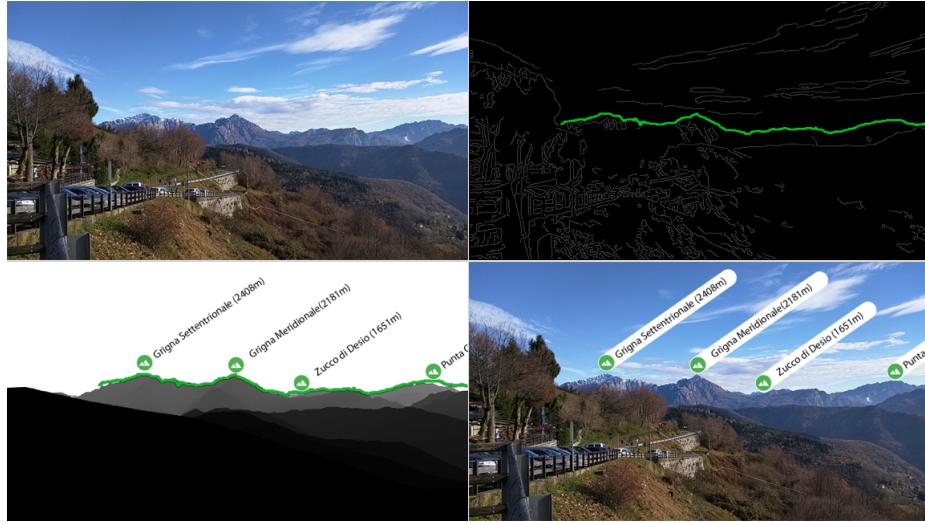


Fig. 2: An example of the image-to-terrain alignment: input frame (top left), skyline detection (top right), alignment between corresponding panorama and skyline detected (bottom left), final result (bottom right).

Figure 2 shows an example of the main steps of the image-to-terrain alignment and peak visualization: the input frame image is captured by the camera of the device (top left); the skyline from the frame image is detected with artificial intelligence (highlighted with green color, top right); the alignment between the virtual panorama from the DEM and the skyline from the camera view is executed (bottom left), so to correct the position of the visible peaks; and peaks

are projected onto the screen with the corrected 2D coordinates (bottom right).

## 4 Dimensions of heterogeneous information integration

The utility of an AR application depends on the quality, timeliness and relevance and usability of the information displayed on the screen. The management of the augmentation data can be characterized by the following dimensions.

**Object semantics** This dimension characterizes the purpose of the core objects published to the user. For example, in a star gazing app, the core objects are the stars and constellations, and they have the purpose of allowing the users to identify astronomical objects. In a mountain tourism app, the core objects are geographical entities such as peaks, massifs, ranges, lakes, rivers, cities and villages, which serve an identification and an orientation purpose; secondary geo-referenced objects can also be relevant, for example, web cams, cultural heritage spots, touristic services, and local events. In the design of an AR application with multiple objects of interest of different natures, it is important to define a priority ranking among objects, provided that the screen space is limited, interaction must be minimized and thus it is imperative to show most relevant objects first.

**Object provenance** Objects may come from a single source or from multiple ones (for example, peak information may be spread in more than one GIS). Multiple sources pose classical heterogeneous data management challenges, including the reconciliation of object properties with different values in alternative data sources, such as the use of alternate names or differences in the geographical coordinates or boundaries for the same geographical object. Given the tight response time constraints of mobile AR apps, reconciliation should be performed off-line. Furthermore, the usage policies of the data sources must be verified in order to ensure that their data are publicly available or exploitable under agreement.

**Object storage and availability** Most modern mobile apps are cloud-enabled and rely on data storage at the back-end and on Internet connectivity between the client and the server. However, an on-line connection cannot be given for granted, because in outdoor conditions, especially in mountain areas, Internet connectivity may be absent. Therefore, policies for objects storage and availability must be introduced, to ensure both on-line and off-line functioning. Such policies must address the download of the data required for disconnected usage, which in turn entails the segmentation of the information to cope with the cases in which the size of the entire data set exceeds the storage capacity of the phone or the complete download would result in unnecessary data transmission cost. Alternative solutions are possible: data can be segmented beforehand into fixed-size modules (e.g., countries, regions, cities and mountain ranges), or the user may be allowed to define custom areas dynamically. Caching data locally at the client side poses classical cache management problems [3]: cache expiration, validation, and replacement rules must be enacted to ensure that the arrival of new

data (e.g., more peaks available in a data source) are promptly reflected in the cached copy. Semantic cache transparency requires the user to be informed when the local copy of the data becomes outdated so as to allow him to decide whether to tolerate misalignment (e.g., when he is in the middle of an outdoor activity) or execute a cache update (e.g., if the notification arrives when at home). As usual in cache management problems, the most appropriate policy depends on the size of the cached data, on the frequency and granularity of an update, and on the tolerability of misalignment. The object semantics clearly influences the latter aspect: peak information varies slowly and mostly in an incremental way, so that the cost of misalignment is low; whereas local event information may prevent caching at all. Besides the cache maintenance issues, local data storage also entails code mobility problems [7]: if the data are subjected to intensive processing before their rendition, a balance must be found between off-line data usage and the performance overhead of processing data at the client side, which may be unfeasible or incompatible with the real-time usage requirements of a mobile AR application. For example, caching the DEM data at the client may enable the application to work also in disconnected mode, but requires sufficient computation power on the phone for extracting the virtual panorama skyline from the DEM point cloud in real-time. Deciding where to execute the code opportunistically may result in the best trade-off between conflicting requirements.

**Object data compression** When the data to be cached in the device are sizable, compression may reduce transmission time and storage occupation. For example, DEM data, typically used by mountain AR apps, consist of numerical data in massive quantity, which can be effectively compressed before transmission and caching [2]. The selection of the compression algorithm must evaluate the trade-off between the compression rate and the decompression overhead, which must be acceptable in a mobile app where low response times can be detrimental for the user experience.

**Object media type** The relevant objects may have different formats: label, text, icon, URL, or a mix thereof. In outdoor applications text and images should be used sparingly and preferably not overlaid on the real-time camera view, because they may mask a too large portion of the real view. On the other hand, fragmenting the user interfaces in too many screens may also induce usability problems, because the app is used in conditions where switching among screens may be difficult (e.g., while walking or with strong light that makes screen reading and command execution difficult).

**Object visualization** The most appropriate visualization method depends on the semantics and media type of the object. Objects can be displayed as points (e.g., peaks), 2D areas (e.g., mountain ranges) and polylines (e.g., trails), possibly accompanied with icons and labels to convey some prominent attributes (e.g., an icon to suggest the object's semantics and a label to display the name and essential features). Visualization design must balance information content and visual clutter. When the relevant objects are too many to fit in the screen, policies must be designed to display them selectively based on their relevance; technical methods for doing this could be pagination and scroll mechanisms (such

as the lower button in the interface of figure 1) or filters based on object properties. Also in this case, interaction design should allow the user to obtain the required information with minimal gestures. When object selection mechanism are provided, they must be accompanied by configuration option, which provide reasonable defaults that apply without any user’s intervention. A pagination and scrolling mechanism must be accompanied by a default policy for deciding the objects that appear in the initial page: for example the most salient peaks. A filter should be accompanied by the default value of the filtering properties: for example, the distance or altitude range of the peaks to show.

Nature	Provenance	Storage/ Availability	Compression	Media type	Visualiz.
DEM	NASA SRTM	Server, client/ online, offline	Yes	Point cloud	Virtual panorama
	ASTER GDEM	Server, client/ online, offline	Yes	Point cloud	Virtual panorama
Peaks	Peakware	Server, client/ online, offline	No	Text, image, link	Point, link
	Open Street Maps	Server, client/ online, offline	No	Text	Point
	PeakBagger	Server, client/ online, offline	No	Text, image, link	Point, link
Waterbody	Waterways guide	Server, client/ online, offline	Yes	Text, image, link	2D area
	Wikidata	Server, client/ online, offline	No	Text, link	2D area, link
Alpine huts	Norwegian trekking association	Server, client/ online, offline	No	Text	Point
	Mountainhuts	Server, client/ online, offline	No	Text	Point
Castles	GPS Data Team	Server, client/ online, offline	No	Text	Point
	Wikidata	Server, client/ online, offline	No	Text, image, link	Point
Trails	Norwegian trekking association	Server, client/ online, offline	Yes	Text	Polyline
	Wikiloc	Server, client/ online, offline	Yes	Text, image, link	Polyline, link
	TrailForks	Server, client/ online, offline	Yes	Text, image, link	Polyline
Towns	Open Street Map	Server, client/ online, offline	Yes	Text	2D area
	GeoNames	Server, client/ online, offline	Yes	Text	Point
Events	Get Events	Server/ online	No	Text, image, link	2D area
	Facebook	Server/ online	No	Text, link	2D area

Table 1: Examples of data sources and their classifications

Table 1 shows a number of examples of data sources potentially relevant for the integration in a mountain tourism mobile AR app, and characterizes them using the above mentioned dimensions.

PeakLens manages a subset of the types of objects mentioned in Table 1 (DEM and peak data), harvests data from multiple sources with reconciliation done off-line at the server side, has a cloud-enabled architecture with data caching and compression, displays (at present) only objects that can be rendered as points, and provides a pagination and scrolling mechanism to cope with visual clutter.

## 5 Usage experience

PeakLens was publicly released for Android devices in the Google Play Store<sup>7</sup> in February 2017. After less than three months, the app usage is spontaneously and steadily growing in the total number of installs, daily active users, and geographical coverage. So far the application has received various ratings with an average of 4,4/5. Almost half of the ratings include reviews, most of which contained positive feedback. Negative reviews and change requests reflect both data integration issues and precision issues.

Data integration feedback mostly refers to data storage and availability: most comments require an option to store off-line data in the SD card of the device, due to the limited space in the primary storage; the second most frequent comment requires more intuitive data visualization, because several users did not recognize the presence of a scroll button to visualize multiple subsets of the peaks and thus believed that some of the peaks in view were not identified. Although the design principle pursued so far in PeakLens is to minimize the number of commands, to cope with visualization clutter, we are studying a simple filtering mechanism with proper defaults, to use in conjunction with pagination and with a self-configurable parameter that adapts the quantity of information visualized by default to the screen density.

Precision issues concentrated on the most difficult aspect of mobile AR applications: aligning in real-time the camera view and the virtual panorama. This core function heavily depends on the reliability of the device's sensors. An error in the GPS position or in the compass orientation produce a virtual panorama that cannot be correctly aligned with the real mountain skyline, resulting in the misplacement of peaks. However, this feature is also where PeakLens outperforms other applications, which solely depend on the sensor data or on the manual adjustment of the compass orientation. The current version of PeakLens extracts the skyline from the camera view with high precision and speed and can search for an alignment not only in the field of view, but in a wider virtual panorama, which permits the automatic correction of substantial angular errors in the compass orientation.

---

<sup>7</sup> <http://play.google.com/store/apps/details?id=com.peaklens.ar>

## 6 Conclusion and future work

In this paper we have illustrated the ongoing experience of developing a mobile outdoor AR application, which integrates DEM and GIS data for the creation of an enriched view of the mountain panorama. We have also discussed the different classes of data that can be exploited to further enhance the basic information of peak position, name, altitude and distance from the viewer, and the technical challenges that managing such information introduces, due to the execution constraints of mobile outdoor AR solutions and the usability issues induced by usage during outdoor activity. Our ongoing and future work concentrates on the harvesting of all the mentioned classes of data, their encoding and compression for off-line and on-line usage, and their representation on the screen. As done for the basic peak data, we will continue soliciting user's feedback in order to create an engaging and effective user experience from a much richer corpus of information.

## References

1. Billingham, M., Clark, A.J., Lee, G.A.: A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction* 8(2-3), 73–272 (2015)
2. Cavallaro, C., Fedorov, R., Bernaschina, C., Fraternali, P.: Compressing web geodata for real-time environmental applications. In: International Workshop on the Internet for Financial Collective Awareness and Intelligence. pp. 119–128. Springer International Publishing (2016)
3. Chan, B.Y., Si, A., Leong, H.V.: Cache management for mobile databases: Design and evaluation. In: Data Engineering, 1998. Proceedings., 14th International Conference on. pp. 54–63. IEEE (1998)
4. Dähne, P., Karigiannis, J.N.: Archeoguide: System architecture of a mobile outdoor augmented reality system. In: null. p. 263. IEEE (2002)
5. Fedorov, R., Frajberg, D., Fraternali, P.: A framework for outdoor mobile augmented reality and its application to mountain peak detection. In: Int. Conf. on Augmented Reality, Virtual Reality and Computer Graphics. pp. 281–301. Springer (2016)
6. Jain, P., Manweiler, J., Roy Choudhury, R.: Overlay: Practical mobile augmented reality. In: Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services. pp. 331–344. ACM (2015)
7. Picco, G.P., Julien, C., Murphy, A.L., Musolesi, M., Roman, G.: Software engineering for mobility: reflecting on the past, peering into the future. In: Herbsleb, J.D., Dwyer, M.B. (eds.) *Proceedings of the Future of Software Engineering, FOSE 2014*, Hyderabad, India, May 31 - June 7, 2014. pp. 13–28. ACM (2014)
8. Reitmayr, G., Drummond, T.: Going out: robust model-based tracking for outdoor augmented reality. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (2006)