

A Location-Based Virtual Reality Application for Mountain Peak Detection

Antonio La Salandra, Piero Fraternali, Darian Frajberg

Politecnico di Milano

Milano, Italy

name.surname@polimi.it

ABSTRACT

Location-based mobile outdoor applications are powerful tools that can engage users in social and environmental tasks and support the emerging paradigm of citizen science. In this paper we present PeakLensVR, a virtual reality location-based mobile app that enables users to capture with their mobile phone panoramic mountain images and later visualize such images, enriched with metadata about the peaks visible from the capture point, with a low-end VR device. The goal of PeakLensVR is to harness the emerging trend of geo-located augmented and virtual reality applications to foster a community of environmentally conscious users who volunteer in the collection of mountain images for environment monitoring purposes.

CCS CONCEPTS

- **Information systems** → *Mobile information processing systems*;
- **Human-centered computing** → *Virtual reality*; • **Computing methodologies** → *Computational photography*; • **Applied computing** → *Environmental sciences*;

KEYWORDS

Virtual Reality, mobile applications, location-based systems, remote sensing, environmental monitoring, mountain identification

ACM Reference Format:

Antonio La Salandra, Piero Fraternali, Darian Frajberg. 2018. A Location-Based Virtual Reality Application for Mountain Peak Detection. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3184558.3191559>

1 INTRODUCTION

The diffusion of Virtual Reality (VR) devices and sensor-equipped mobile phones is fostering immersive content consumption and sharing, not only for entertainment but also for environmental education and awareness. Nowadays, users can easily capture 3D immersive videos and images with their smartphones, post them on their favorite social platforms for others to relive their experience

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '18 Companion, April 23–27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04.
<https://doi.org/10.1145/3184558.3191559>

with a plain smartphone or a consumer-grade VR device, such as the Google Cardboard¹ and the Samsung Gear VR².

A promising field for VR mobile applications that exploit the user's location is environment monitoring, where more and more scientists seek the contribution of citizens for the collection of geo-referenced data to use in the assessment of natural processes, such as climate change [15], hydrological risk assessment[14], and alien plant species diffusion [13]. The collection of data with sufficient spatial and temporal density requires hundreds of thousands contributors, durably engaged. Virtual reality, coupled to mobile sensing of geo-referenced data, can play an important role in the diffusion of environment monitoring citizen science applications: users can collect data with an outdoor mobile application and then transform such data into immersive content sharable within their social circles. In this way, other users can relive the experience in a quasi-real way and join the community of contributors.

This paper describes PeakLensVR, a location-based mobile application for the collection of mountain images and their transformation into immersive content consumable with VR devices and sharable in social networks. With PeakLensVR, users can take 360° panoramic photos of mountain landscapes, record, while acquiring the scene, the camera position and orientation, and later visualize in immersive mode a panoramic image enriched with information about the peaks in view computed using the position and orientation saved during the capture. Metadata about the mountains (peak, name, altitude, distance from the viewer) are rendered as hotspots that the user can activate with the controller of the VR device, as shown in Figure 1. Visualization and interaction can be performed with a simple smartphone or in immersive mode with the Google Cardboard VR viewer. Annotated panoramic images are converted into the popular Extensible Metadata Platform (XMP) format, for easy sharing in social networks.

PeakLensVR is part of a research effort for the monitoring of mountain snow coverage, and hence of water availability, through the analysis of low-cost multimedia content [4, 7]. The SnowWatch Portal³ collects mountain photos, user-generated or crawled from touristic Web cams, which are analyzed to extract snow indexes (e.g., minimum snow altitude) and forecast the water stock present in the mountains in the form of snow [4]. PeakLens⁴ is an Augmented Reality location-based mobile app developed for collecting photos of mountains for the SnowWatch data set; to engage users, it identifies the peaks at which the user is looking through the mobile phone in real-time, using as input the frames captured by the smartphone camera, the user's position and the camera orientation, estimated

¹<http://vr.google.com/cardboard/>

²<http://www.samsung.com/global/galaxy/gear-vr/>

³<http://snowwatch.polimi.it>

⁴<http://peaklens.com>

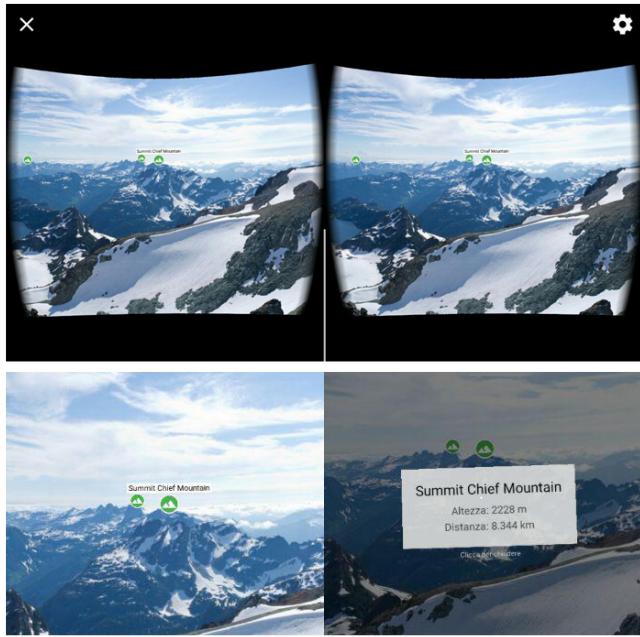


Figure 1: Visualization of a panoramic image and an activated hotspot on a visible peak

from the GPS, gyroscope, magnetometer and accelerometer [7]. To compute the positions of peaks on the phone screen, PeakLens extracts the skyline from the camera frame and aligns it in real-time with a virtual panorama built by querying i.e. the NASA SRTM [18] Digital Elevation Model (DEM) of the Earth with the position and orientation of the user. The data of the DEM server are enriched with the coordinates and metadata of the mountain peaks, mainly acquired from the OpenStreetMap⁵ (OSM) GIS system. Comparing the real and virtual skyline, PeakLens determines the on screen coordinates of the visible peaks and inserts tags with their names, altitude, and distance in the correct screen positions, augmenting the user's view in real-time.

PeakLensVR aims at boosting the community of PeakLens users and SnowWatch contributors enabling the recording of *panoramic 360°* scenic images and their replay in *immersive mode* with a mobile phone or VR headset. Users can relive the outdoor experience made by others, preview a scenic panorama, get information about the mountains in view in an immersive and interactive way, and hopefully decide to join the SnowWatch community and contribute to data collection.

PeakLensVR reuses part of the components of SnowWatch and PeakLens, but adds new capabilities and extends existing ones significantly: it introduces a new module for 360° image acquisition, extends the peak positioning algorithm to cope with large panoramic photos constructed by stitching captured image sequences, and supports content fruition in VR mode. Its contributions can be summarized as follows.

⁵<http://www.openstreetmap.org>

PeakLensVR features a processing pipeline for creating a panoramic view from a sequence of images tagged with the position and orientation of the camera; the pipeline consists of sensor initialization, tilt control, sensor-based images alignment, content-based alignment refinement, color correction, seam estimation, image blending, and panoramic image generation; the modules of the pipeline are optimized for the limited resources of mobile devices; compared to a classic stitching algorithm of a digital camera, PeakLensVR records, for each image in the captured sequence, the orientation of the camera estimated from the gyroscope and accelerometer and exploits it in the composition of the panoramic view.

To provide an engaging user experience, PeakLensVR extends the peak positioning approach of PeakLens to cope with large 360° images. The peaks visible in the captured panorama are identified, their position in the cylindric view is determined, and hotspots are inserted in the view in correspondence of each peak. An advanced *content-based* algorithm uses image processing techniques and external geographical data sources for achieving higher precision in the positioning of peaks onto the panoramic image, going beyond the limited accuracy of position estimation based only on the device orientation derived from the sensors. The acquired panoramic images enriched with peak positions can be viewed in the full screen mode of a smartphone or in immersive mode with the Google Cardboard VR viewer. Hotspots are inserted in correspondence of the peak positions and the user can interact with them to obtain metadata about the peaks in view.

Finally, we evaluate the accuracy of peak positioning in panoramic images, by comparing the peak positions computed with the sensor-based and content-based algorithms to *ground truth* peak positions manually established with the help of an online service, developed for that purpose, which displays the panorama visible from the user's location, highlighting peak names and positions. The evaluation shows that, due to the inherent inaccuracy of today's consumer grade mobile phones in sensing orientation, content-based peak positioning delivers superior precision and can also deal with distortions introduced by the image stitching process.

2 RELATED WORK

PeakLensVR is a Virtual Reality (VR) application. VR is a computer generated environment that humans can interact with a sense of presence similar to that of real life [17], thanks to ad hoc viewers (helmet or eyeglasses) for projecting the digital world, headphones for 3D spatial sound, and controllers for interaction. Recently, VR applications have reached the consumer market, with VR platforms using either PCs or smartphones as computational units. Smartphone solutions use the headset only as a support for the smartphone, which manages both computation and display. In [5] a detailed description of the last-generation VR can be found. PeakLensVR works on smartphone VR systems, because the environmental data collection and awareness effort that motivates the project requires deployment to a vast community of users who collaborate to the creation of the panoramic images and to their diffusion. In such a class of devices, Gear VR and Google Cardboard are the most popular solutions. PeakLensVR is developed for Google Cardboard, to exploit the wide range of supported smartphones and headsets, and thus

better reach non-professional users; ultimately, even such a simple platform proved capable of affording good enough visualization and interaction with the 360° immersive content.

VR applications based on immersive multimedia content, such as 360° photos and videos, are increasing in popularity: YouTube⁶ offers the possibility to watch videos at 360° and the same functionality has been launched by Facebook⁷ for both photos and videos. This trend has sparked interest in the community of GIS researchers, as a new means to disseminate geographical knowledge. Early studies on the integration of VR and GIS are discussed in [12, 20]. [3] surveys VR GIS applied to urban and environmental planning, design, entertainment, education and training, personal health and wellness. Virtual tours are the most common applications, e.g., supported by the Google Street View app⁸ in immersive mode with the Cardboard headset. ArcheoVR [2] integrates VR, GIS and archeology; it features a realistic 3D model of the Itapeva Rocky Shelter, a prehistoric site in Brazil; exploration is supported with hot spots placed in points of interest and audiovisual guides following the user during the navigation of the site. Trail Me Up⁹ is a Web platform that offers virtual walking tours. Content is captured with a backpack with a mounted camera that automatically shoots geo-located photos while walking. The collected photos are processed to create a virtual tour, in which hot spots are manually placed to mark points of interest, such as mountains.

PeakLensVR offers a virtual panoramic view of a mountain landscape, captured with a simple mobile phone in any place on Earth where mountains are visible; hot spots are automatically placed in the correct positions of the screen in correspondence to the summits, without the need of any manual editing, and useful information about the mountain peaks in view is automatically extracted and connected to the hot spots by querying DEM and GIS sources with the user's location and the orientation of the camera recorded during the image capture.

3 PROCESSING PIPELINE

Figure 2 overviews the processing steps and the application modules that compose PeakLensVR.

3.1 Scene acquisition

Scene acquisition lets the user capture a panoramic view of the surrounding environment by taking a series of photos with the smartphone camera. Also latitude, longitude, camera field of view, and orientation are captured to monitor the smartphone position during the shooting and support the stitching of images into a 360° panorama.

Initialization To work properly, the magnetometer must be calibrated, therefore a message that invites the user to move the smartphone in an 8-shaped pattern is presented at the beginning of the recording session. The movement forces the magnetometer to reset and return the correct cardinal direction. After calibration and

⁶<http://www.youtube.com>

⁷<http://facebook360.fb.com>

⁸<http://www.google.com/streetview/>

⁹<http://www.trailmeup.com>

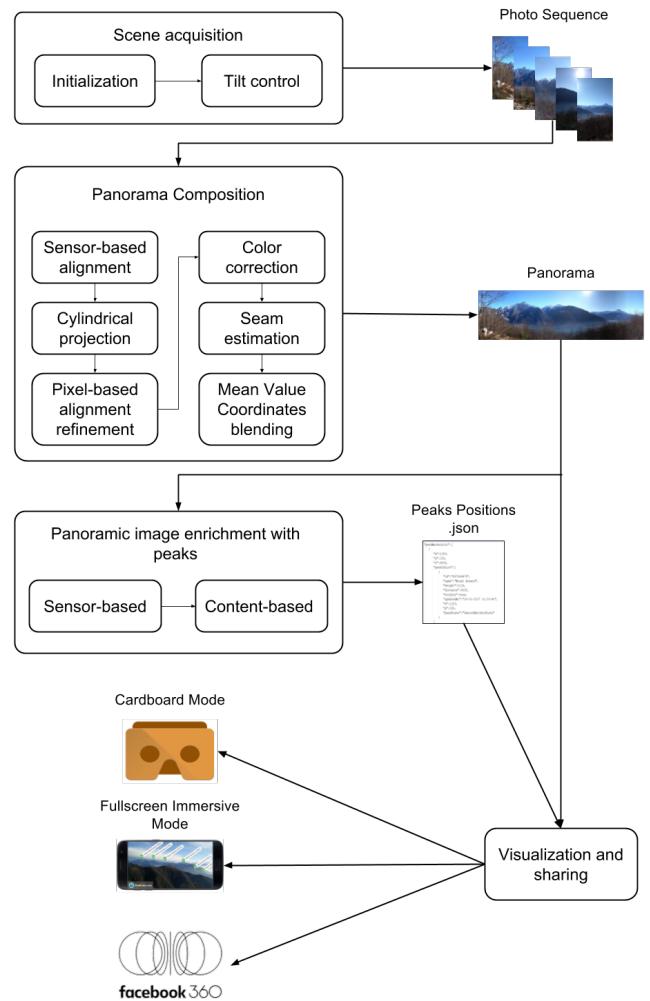


Figure 2: Overview of PeakLensVR main processing steps and modules

GPS activation, the interface to shoot photos is displayed. Moreover, the latitude, longitude and field of view of the device are saved.

Tilt control A panoramic scene is acquired by rotating the smartphone clockwise around the vertical axis. To control the process and the smartphone orientation, azimuth, pitch and roll are used. Two values are used for the azimuth: relative and absolute. The *absolute azimuth*, i.e., the angle between the device compass direction and the magnetic North, is computed from data provided by the magnetometer and accelerometer; it is stored to be later exploited in the peak identification phase. The *relative azimuth* refers to the phone position when the accelerometer is activated and is used to control the device orientation during the acquisition and to align photos during panorama generation. It is computed only from the gyroscope and accelerometer, because the magnetometer suffers from inaccuracies due to magnetic field variations. To generate cylindrical panoramas, one photo in the series is taken every

10 degrees w.r.t. the relative azimuth, while the user rotates the phone clockwise around the vertical axis. For each photo, relative azimuth, pitch and roll are captured. The absolute azimuth is also saved separately, for later use. Without a tripod, users often rotate around their vertical axis instead of around the vertical axis of the smartphone, which causes parallax errors that need to be compensated. The same applies for pitch and roll, as movements that might compromise the final result are detected: if the smartphone is tilted laterally or frontally by more than 25° acquisition is stopped. When an acceptable 360° rotation is detected or the user presses a button to stop the capture, the panorama composition starts.

3.2 Panorama composition

After the images and the sensor data have been captured, Peak-LensVR composes the cylindrical panorama. First images are aligned w.r.t. the reference surface chosen for the composition. Next, they are stitched together to produce a panorama without noticeable artifacts. Alignment and stitching are sequential: only one image at a time and the current partial panorama are kept in memory to reduce required space. The alignment process prepares the images for merging onto a cylindrical view. Images are first rotated by applying a perspective transformation based on sensors data to compensate tilt errors during the acquisition. Then, they are projected onto the cylindrical reference surface and a pixel-based alignment post-processing is applied, to refine the relative image positions estimated from the sensors data.

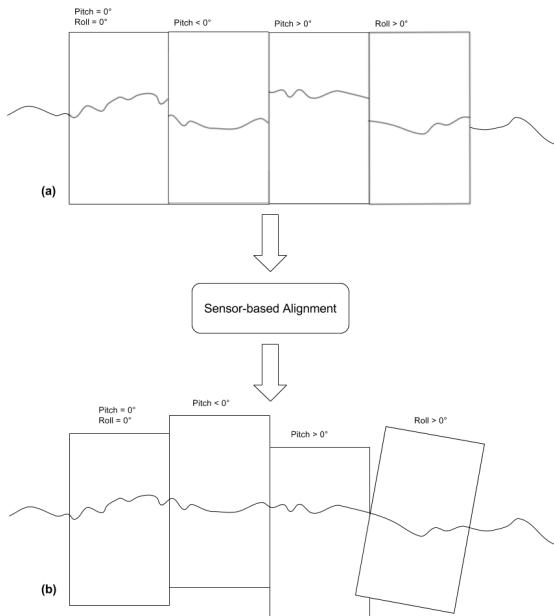


Figure 3: Image sequence before and after sensor-based alignment.

Sensor-based alignment Images are sequentially warped by applying a perspective transformation based on the pitch and roll

values. The pitch of the first image is taken as reference, while the reference value of the roll is set to 0 (phone in vertical position). The first image is warped by considering only the roll; subsequent ones are warped considering their rolls and the difference between their pitch and the reference one. This approach reconstructs the movement of the camera during the acquisition. Figure 3 shows a sequence of four images taken with variable pitch and roll and the result after the application of sensor-based alignment.

Cylindrical projection The next step is to project each image onto the cylindrical reference surface [19]. First, *Forward Warping* is used to map the image onto the cylinder. Next, an *Inverse Warping* is used to map pixels onto the cylinder to the final image pixels. Finally, bilinear interpolation is used to compute the colors of the destination pixels.

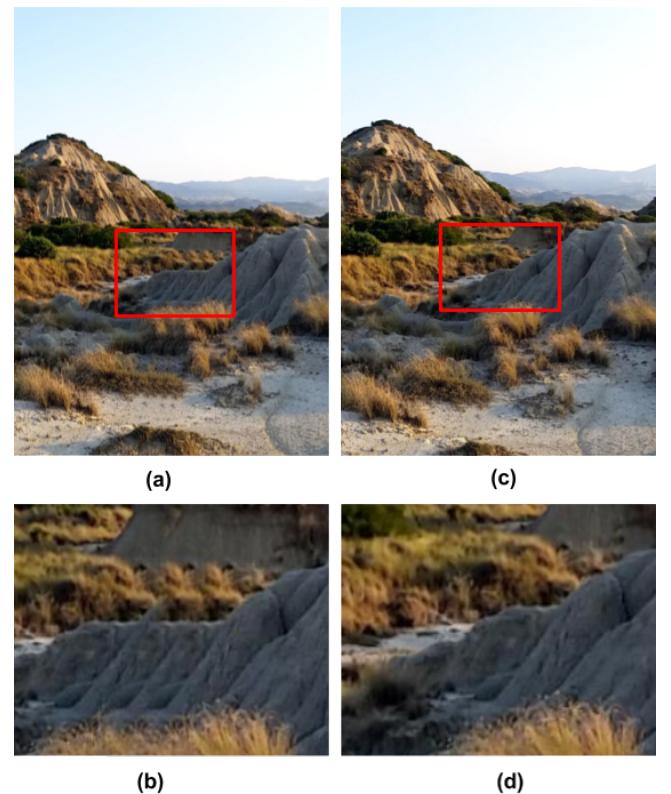


Figure 4: Pixel-based alignment refinement. a) result without pixel-based correction; b) detail of the image showing the introduced artifacts; c) result with pixel-based correction; d) detail of the image showing that the introduced artifacts have disappeared.

Pixel-based alignment refinement After the tilt angle compensation performed by the sensor-based alignment and the cylindrical warping, a sensible problem remains due to the erroneous acquisition movement resulting in a translational component, which cannot be detected by sensors and introduces a parallax error that

causes artifacts in the composed panorama. To correct such anomaly, a pixel-based alignment refinement step is used, following the approach in [19]. Pixel-based alignment compares the intensities of the images pixels and computes an horizontal and vertical displacement that refines the position of each image. Given two images, an horizontal and vertical translation is applied to their positions, the error for that offset is evaluated to test how much pixels agree, and the process is iterated. After all translations in a given interval have been tried, the one with the smallest error is chosen. The approach exploits an error metric to evaluate the quality of the alignment, and a search technique to explore the space of translations. Figure 4 shows the stitching result with and without the application of these fine grain adjustments. In Figure 4(b) the problem of repeated objects introduced by the translational component of the azimuth variation is highlighted. Figure 4(d) shows how these artifacts disappear after alignment refinement.

After warping and alignment, image stitching glues together the images to compose the panorama. First, image colors are equalized, then the best path to stitch together each image pair is identified, and finally blending is used to remove seams.

Color Correction Color differences between images may be present in case the user opts for using automatic settings for exposure and white balance during the acquisition. Color correction aims at reducing this effect; PeakLensVR implements an algorithm based on the approach described in [21]. To apply the color correction, an image (possibly the best one) is selected as the reference to correct all the others. A heuristic quality evaluation criterion that proved suited for outdoor images in the day light is the selection of the image with the median *Luma* value [16]. Next, color correction coefficients are computed for each image in the sequence, to equalize it with respect to the reference image. With the computed coefficients, images are color-corrected in an iterative way, starting from the first one in the sequence and progressing in order of acquisition.

Seam estimation Moving objects and residual alignment errors may cause ghosting artifacts in the panoramic image. Seam estimation is a technique used to find the best path along which to cut and merge images in places where they differ the least. As in [10, 11, 21], we use dynamic programming to find the optimal seams, as its limited computation and memory requirements make it particularly suitable to a mobile implementation. The dynamic programming approach solves a cost minimization problem. First, an error surface is generated by computing the pixel value squared differences in the overlapping area of two images. This surface represents residual errors between the two images to stitch. Then, the error surface is scanned row by row from top to bottom and a cumulative minimum squared difference error surface is computed. This surface contains all the possible paths to cut images. The minimum cost path is computed by traversing the cumulative error surface from the last row to the first one. The minimum value point on the last row is selected and then, at each iteration, the minimum value connected point in the upper row is selected, until the first row is reached and the path is completed.

Mean Value Coordinates blending After the best seam has been found, the next image is stitched along the seam to the panoramic



Figure 5: Example of an optimal seam estimated in the overlapping area of two images.

image under composition, using a blending technique to smoothen the transition. PeakLensVR uses, as the best trade-off between quality and computational requirements, Mean Value Coordinates Blending, a technique proposed in [6] for instant image cloning and used in [21] for stitching purposes. To speedup the computation, an optimization method has been used. Correction colors are computed accurately only close to the seam. Then, such accurate values are (bi)linearly interpolated for the other pixels.

3.3 Panoramic image enrichment with peaks

After the panoramic image has been composed, peak identification labels the mountain peaks that appear in it with their metadata (name, altitude, distance from viewer, etc). The process consists of two steps: first the 2D screen coordinates of the peak summits are computed using the user's position and orientation estimated with sensors data, a repository of peak coordinates (collected from such open source GIS as Open Street Map), and a Digital Elevation Model of the Earth (mainly obtained from the NASA SRTM data set [18]); then, to overcome the low precision of the compass sensor, such initial estimation is refined by comparing the mountain skyline extracted from the camera view and a reference skyline extracted from the 3D model of the Earth computed from the DEM. The algorithms used in PeakLensVR extend those developed in the PeakLens project [8], to work with 360° panoramic images.

Note that peak identification requires an Internet connection to fetch the information. If the connection is available at the end of the panorama generation phase, the module is automatically launched; otherwise, the panorama is placed in a waiting list and the peak identification functionality is executed at the next launch of the application with an active network connection.

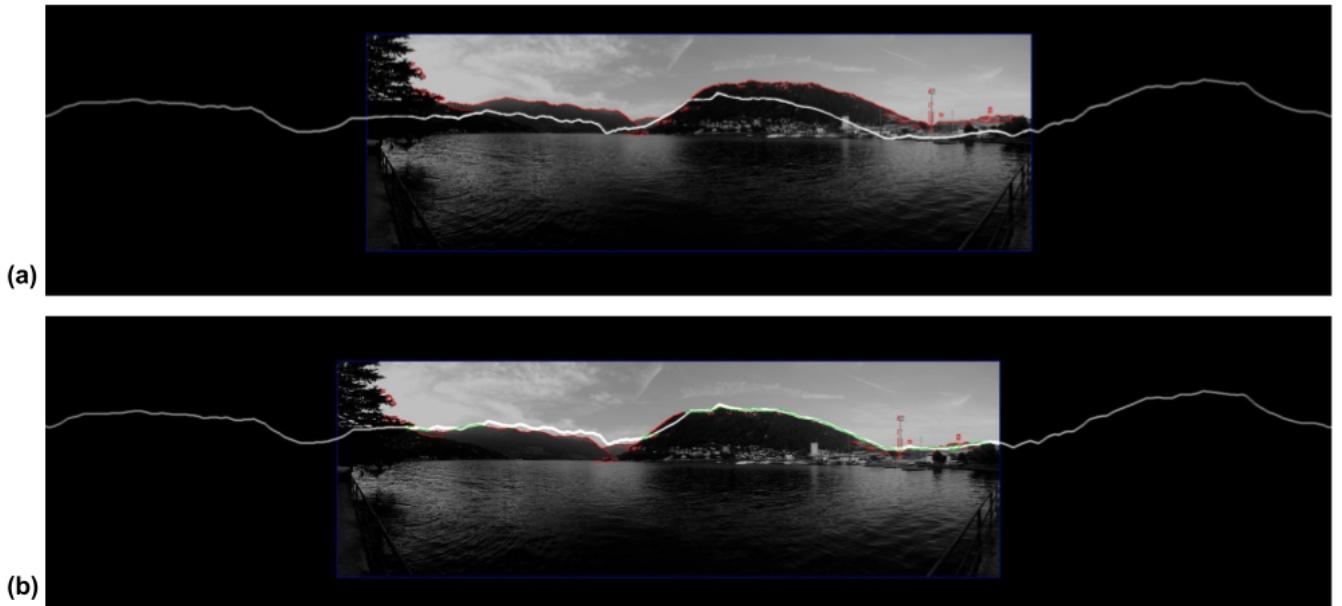


Figure 6: (a) panorama to skyline alignment with sensor data only, (b) alignment adjusted with content based matching

Moreover, this module can be easily extended so as to work with external panoramic photos retrieved from sources such as Flickr¹⁰, provided that they contain all the required metadata, which should also be ported into the application's predefined format.

Sensor-based peak positioning Sensor-based peak positioning exploits an online service that takes in input the latitude and longitude of the user and returns as output an image with the representation of the skyline visible from that point and the list of visible peaks and their positions with respect to the skyline. Sensor-based peak positioning maps the peak coordinates from the skyline image coordinate system (XY) to the panoramic image coordinate system (xy).

Mapping the panoramic image to the skyline image requires scaling the former based on its field of view so to achieve the same pixel per degree ratio, and translating it to accommodate pitch and roll. Registering the panoramic image with the skyline image permits the identification of the peaks in view.

Content-based position refinement Peak positions computed by using only sensor data may be imprecise due to sensor errors, e.g., due to the wrong estimation of the orientation of the phone caused by compass errors. Accuracy can be improved with a content-based approach, by comparing the actual camera view with the virtual skyline image retrieved from the online service based on the user's position.

The first step is the extraction of the skyline pixels from the panoramic image. Skyline extraction from mountain images is a well-investigated problem [1]; for the purpose of the application, we have exploited the algorithm described in [8]. The skyline image

extracted from the panorama is scaled with respect to the skyline image downloaded from the online service and then the alignment in which the two skylines overlap more is searched. This process applies a template matching technique, which iteratively applies vertical and horizontal offsets and computes for each offset a matching score based on the degree of overlap of the two skylines. Finally, the offset with the highest score is used to correct the peak positions. Figure 6 shows the alignment of the panorama and skyline images by using sensor data only (a); and with the application of the matching algorithm (b).

Note that the skyline matching algorithm of PeakLensVR differs from the original one of PeakLens, because in the latter case the field of view of the camera image is much smaller than the one of the skyline (which is 360°), whereas in PeakLensVR both skylines can span 360° . Therefore, the (planar) search method of the best alignment has been modified by partially replicating the virtual skyline horizontally of a percentage of its width, defined heuristically to make a trade-off between the computational search cost and the possibility of finding the right match.

3.4 Visualization and sharing

PeakLensVR offers different ways to visualize and share content: panoramas can be viewed using the Cardboard mode with a VR headset or in the mobile phone using Fullscreen immersive mode. It is also possible to share panoramas on Facebook or with other users of PeakLensVR.

Cardboard and Immersive Full-screen Mode visualization In both Cardboard and Android Immersive Full-screen modes, the virtual environment is built by projecting the panorama and the peaks onto a cylindrical surface. The camera that generates the user

¹⁰<http://www.flickr.com>

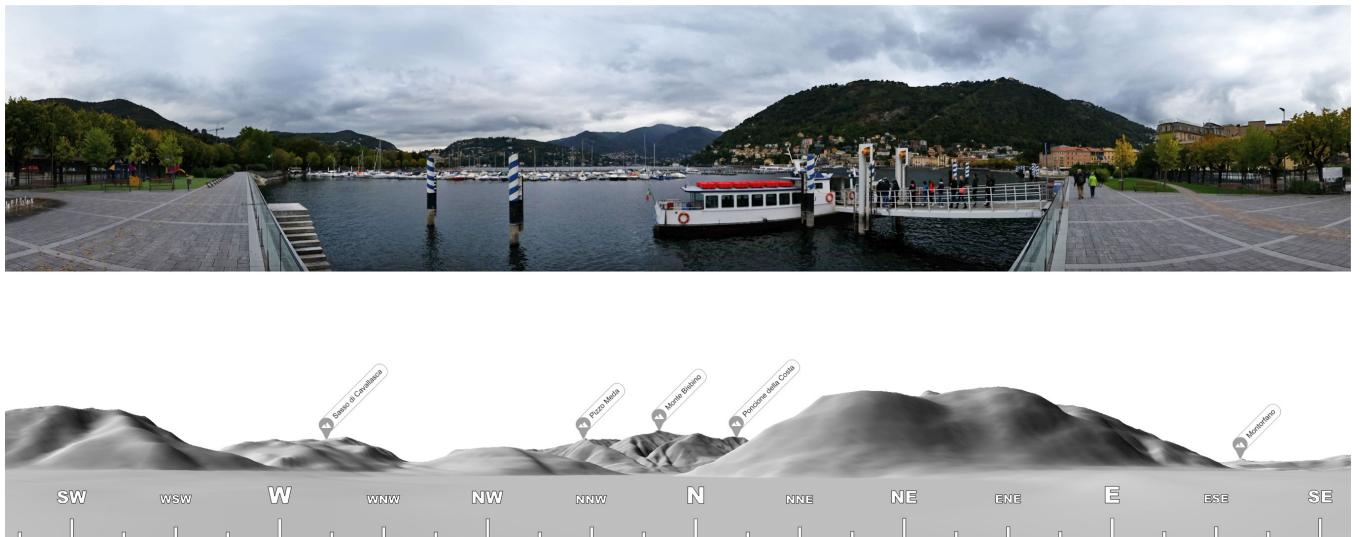


Figure 7: The captured panoramic image (top) and the corresponding virtual image used to manually define the ground truth coordinates of peak positions (bottom)

view is placed at the center of this cylinder. In Cardboard mode, it is possible to turn around and watch the scene by moving the head as if looking at a real panorama. Sensors capture movements and apply them to the camera inside the 3D scene, scrolling automatically the user view. Hot spots are placed at peak positions (see Figure 1): the user can point at a hot spot and visualize a dialog with all the metadata of the peak, by looking at it and clicking by means of a controller. In Immersive Full-screen mode, instead of clickable hot spots, labels with peak names are shown. It is possible to click and drag the panorama to scroll the scene.

Building the 3D environment requires mapping the panorama onto the 3D scene and transforming peak coordinates from the planar system of the panoramic image to the cylindrical system of the scene. First, the captured panorama, and the peak hot spots or labels are mapped to a planar reference frame that represents a complete 360° view. Then, the resulting image is projected onto the cylindrical reference frame.

Sharing PeakLensVR panoramas can be shared among users of the application. All the metadata required to invoke the peak identification service are encapsulated inside the EXIF container of the image, so that panoramas can be transferred by simple file copy, without application-specific installation procedures. At the application launch, new panoramas are detected and the service to identify their peaks is automatically invoked. Once the list of peaks has been generated, the user can visualize the panorama as if he had taken it.

Panoramas can also be shared on Facebook and visualized in the immersive mode provided by the social network. PeakLensVR can export panoramas using Facebook metadata (Photo Sphere XMP) to allow their rendering in that platform.

4 EVALUATION

The quality of the user experience with PeakLensVR depends on the fidelity of the panoramic view built with image stitching and on the accuracy of the position of peaks in the panoramic image.

The evaluation focuses on the accuracy of the positions of peaks in the panoramic view; this permits the appreciation of the fidelity of the stitched image, which contains the skyline where peaks are positioned, and of the improvement of the skyline matching algorithm with respect to purely sensor-based peak positioning.

The **data set** consists of 10 panoramic photos taken in the Como lake area in North Italy and in the Pollino National Park in South Italy. Photos have been taken in various weather conditions, with different exposures and, in some of them, with objects occluding the skyline. They have a horizontal field of view variable from approximately 96° degrees to 360° and a number of visible peaks with metadata in Open Street Map between 1 and 11. The data set has been produced by using a Samsung Galaxy s5 and 1280x720 pixel frames to generate panoramas. All panoramic images have been captured with PeakLensVR.

The **ground truth** peak positions used to compare the positions generated by the sensor-based and matching algorithms and to evaluate errors have been determined manually with the support of an online service that, given the geographical coordinates of the view point, returns a digital representation of the panorama visible in that point, enriched with the labels of each visible peaks placed in the position of its summit. Figure 7 shows an example: the virtual panorama image has been used as a guide to manually set the correct on-screen coordinates of each visible peak.

The **metrics** for evaluating errors in the on screen peak positioning are the Euclidean Pixel Error (EPE), the Euclidean Degree Error (EDE), the Azimuth Degree Error (ADE) and the Pitch Degree Error (PDE). The EPE is the Euclidean distance in pixels between the ground truth and the estimated peak position. The EDE is the

#	Sensor-based		Content-based		
	Mean	Median	Mean	Median	Improv.
1	3,97	3,68	1,72	1,80	56,72%
2	3,19	3,19	1,88	1,60	41,10%
3	2,22	2,24	1,84	1,77	17,40%
4	5,89	3,73	2,37	1,93	59,73%
5	4,57	4,57	2,20	2,20	51,73%
6	9,07	9,07	3,35	3,35	63,06%
7	4,35	4,17	2,56	2,48	41,18%
8	99,24	99,24	2,15	2,15	97,83%
9	97,29	97,71	3,41	3,41	96,49%
10	2,68	2,67	2,10	2,10	21,32%
Mean	23,25	22,99	2,36	2,28	54,66%
Median	4,46	3,95	2,18	2,13	54,22%

Table 1: Sensor-based and content-based Euclidean Degree Error EDE comparison.

same distance measured in degrees, computed using the EPE and the pixel-degree ratio of each panoramic image. The ADE is the distance along the horizontal axis in degrees and the PDE the distance along the vertical axis. ADE and PDE quantify the azimuth and pitch error, respectively. Error statistics have been computed for the peaks in each panorama. The min, max, mean and median EPE and EDE, the mean and median ADE and PDE have been computed for each panorama. The EDE has been considered for all peaks in the data set and used to compute the frequency distribution of peak positions in error classes ranging from 1 to 6 degrees.

Table 1 shows the EDE values computed for each panorama when using the sensor-based and content-based approaches and the improvement in the mean EDE with respect to the sensor estimation. These values show the fluctuations of sensor data across panoramas and the variability of errors across peaks of the same panorama. On average, errors are halved. An interesting result is the case of Samples 8, 9. Even if their initial sensor-based estimates were totally wrong, the matching algorithm has corrected the errors almost totally. There are some cases in which the improvement is not so evident, but these are also the cases in which the sensor-based error was smaller (Samples 3 and 10). Ultimately, a mean improvement of about 54% and a mean EDE of about 2 degrees (that corresponds to an EPE of 40 pixels) show that the matching algorithm improves peak positions quite substantially.

The peak frequency distribution with respect to the EDE of the sensor based and matching algorithms has been computed for the whole data set, as displayed in Figure 8: 85% of the peaks have an EDE of less than 3 degrees after the application of the matching algorithm. This value corresponds to an EPE of about 57 pixels and can be considered a good result if compared with the EDE of the sensor-based approach, for which only the 35% of peaks have an EDE of less than 3 degrees and the 17,5% have an EDE greater than 6 degrees.

For brevity, we have reported only the results of the EDE assessment, which exhibits the most relevant findings. The analysis of ADE and PDE confirms the improvement afforded by the matching algorithm,

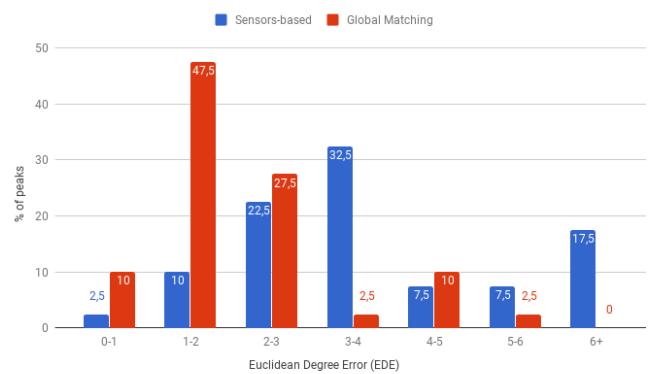


Figure 8: Classification of the frequency distribution of peaks with respect to their EDE after the sensor-based and content-based matching approaches.

5 CONCLUSIONS AND FUTURE WORK

The paper has described PeakLensVR, a location-based mobile application that allows users to shoot panoramic photos, automatically identifies the mountain peaks in the image based on their position and orientation, and enables the visualization of photos augmented with peak information in VR mode.

In some cases, if users make erroneous movements during the image sequence acquisition or the DEM data about landforms or the GIS peak coordinates are imprecise, the generated panorama may contain errors that affect the peak positioning, not compensated by the implemented *content-based* skyline matching algorithm. This effect could be attenuated by introducing a *local* matching algorithm [7]: once the image skyline has been aligned with the virtual skyline and the peak positions have been estimated, local matching performs another refinement step. For each peak a small patch of the skyline is extracted around the estimated peak position and the skyline alignment is repeated only for that patch. This step could further improve results because it is less affected by the imprecisions of the virtual or image skyline.

A second line of development regards the rendition in VR mode of other objects of interest besides peaks. We are collecting GIS data about mountain huts and trails, which could be merged into the panoramic image by extending the peak positioning algorithm as discussed in [9].

REFERENCES

- [1] Lionel Baboud, Martin Cadik, Elmar Eisemann, and Hans-Peter Seidel. 2011. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 41–48. <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5968010>
- [2] Eduardo Zilles Borba, Andre Montes, Marcio Almeida, Mario Nagamura, Roseli Lopes, Marcelo Knorich Zuffo, Astolfo Araujo, and Regis Kopper. 2017. ArcheoVR: Exploring Itapeva’s archeological site. In *Virtual Reality (VR), 2017 IEEE*. IEEE, 463–464.
- [3] Maged N Kamel Boulos, Zhihan Lu, Paul Guerrero, Charlene Jennett, and Anthony Steed. 2017. From urban planning and emergency training to PokéMon Go: applications of virtual reality GIS (VRGIS) and augmented reality GIS (ARGIS) in personal, public and environmental health. *International journal of health geographics* 16, 1 (2017), 7.

- [4] Andrea Castelletti, Roman Fedorov, Piero Fraternali, and Matteo Giuliani. 2016. Multimedia on the Mountaintop: Using Public Snow Images to Improve Water Systems Operation. In *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*. 948–957.
- [5] Joshua Q Coburn, Ian Freeman, and John L Salmon. 2017. A review of the capabilities of current low-cost virtual reality technology and its potential to enhance the design process. *Journal of Computing and Information Science in Engineering* 17, 3 (2017), 031013.
- [6] Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. 2009. Coordinates for instant image cloning. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 67.
- [7] Roman Fedorov, Darian Frajberg, and Piero Fraternali. 2016. A framework for outdoor mobile augmented reality and its application to mountain peak detection. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, 281–301.
- [8] Darian Frajberg, Piero Fraternali, and Rocio Nahime Torres. 2017. Convolutional Neural Network for Pixel-Wise Skyline Detection. In *Artificial Neural Networks and Machine Learning - ICANN 2017 - 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part II*. 12–20.
- [9] Darian Frajberg, Piero Fraternali, and Rocio Nahime Torres. 2017. Heterogeneous Information Integration for Mountain Augmented Reality Mobile Apps. In *2017 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2017, Tokyo, Japan, October 19-21, 2017*. 313–322. <https://doi.org/10.1109/DSAA.2017.5>
- [10] Seong Jong Ha, Hyung Il Koo, Sang Hwa Lee, Nam Ik Cho, and Soo Kyun Kim. 2007. Panorama mosaic optimization for mobile camera systems. *IEEE Transactions on Consumer Electronics* 53, 4 (2007).
- [11] Seong Jong Ha, Sang Hwa Lee, Nam Ik Cho, Soo Kyun Kim, and Byungjun Son. 2008. Embedded panoramic mosaic system using auto-shot interface. *IEEE Transactions on Consumer Electronics* 54, 1 (2008).
- [12] Bo Huang, Bin Jiang, and Hui Li. 2001. An integration of GIS, virtual reality and the Internet for visualization, analysis and exploration of spatial data. *International Journal of Geographical Information Science* 15, 5 (2001), 439–456.
- [13] Alexis Joly, Hervé Goëau, Julien Champ, Samuel Dufour-Kowalski, Henning Müller, and Pierre Bonnet. 2016. Crowdsourcing biodiversity monitoring: how sharing your photo stream can sustain our planet. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 958–967.
- [14] Christopher S Lowry and Michael N Fienan. 2013. CrowdHydrology: crowdsourcing hydrologic data and engaging citizen scientists. *Ground Water* 51, 1 (2013), 151–156.
- [15] Lara Piccolo, Miriam Fernández, Harith Alani, Arno Scharl, Michael Föls, and David Herring. 2016. Climate Change Engagement: Results of a Multi-Task Game with a Purpose. In *Proceedings of the 1st international workshop on Social Web for Environmental and Ecological Monitoring*. AAAI Publications.
- [16] Charles Poynton. 1998. YUV and luminance considered harmful: A plea for precise terminology in video. (1998).
- [17] Jonathan Steuer. 1992. Defining virtual reality: Dimensions determining telepresence. *Journal of communication* 42, 4 (1992), 73–93.
- [18] U.S. Geological Survey. 2009. Shuttle Radar Topography Mission (SRTM): U.S. Geological Survey Fact Sheet 2009-3087. <http://dds.cr.usgs.gov/srtm/>
- [19] Richard Szeliski. 2006. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* 2, 1 (2006), 1–104.
- [20] Edward Verbree, Gert Van Maren, Rick Germs, Frederik Jansen, and Menno-Jan Kraak. 1999. Interaction in virtual world views-linking 3D GIS with VR. *International Journal of Geographical Information Science* 13, 4 (1999), 385–396.
- [21] Yingen Xiong and Kari Pulli. 2010. Fast panorama stitching for high-quality panoramic images on mobile phones. *IEEE Transactions on Consumer Electronics* 56, 2 (2010).