Exploiting User Generated Content for Mountain Peak Detection

Roman Fedorov
Politecnico di Milano,
Piazza Leonardo 32, Milan,
Italy
roman.fedorov@polimi.it

Davide Martinenghi
Politecnico di Milano,
Piazza Leonardo 32, Milan,
Italy
davide.martinenghi@polimi.it

Marco Tagliasacchi
Politecnico di Milano,
Piazza Leonardo 32, Milan,
Italy
marco.tagliasacchi@polimi.it

Andrea Castelletti
Politecnico di Milano,
Piazza Leonardo 32, Milan,
Italy
andrea.castelletti@polimi.it

We present a system for the classification of mountain panoramas from user-generated photographs followed by identification and extraction of mountain peaks from those panoramas. We have developed an automatic technique that, given as input a geo-tagged photograph, estimates its FOV (Field Of View) and the direction of the camera using a matching algorithm on the photograph edge maps and a rendered view of the mountain silhouettes that should be seen from the observer's point of view. The extraction algorithm then identifies the mountain peaks present in the photograph and their profiles. We discuss possible applications in social fields like photographs peaks tagging on social portals, augmented reality on mobile devices when viewing a mountain panorama, and generation of collective intelligence systems (such as environmental models) from massive social media collections (e.g. snow water availability maps based on mountain peaks states extracted from photographs hosting services).

User Generated Content(UGC); collective intelligence; passive crowdsourcing; environmental models; geotagging; mountain identification; peaks detection.

1. INTRODUCTION

Recently the Web has become a publishing platform of massive user personal media content, mostly photographs and videos. User generated content has been exploited as a form of "passive human computation", in which the collective intelligence of a large group of users is harnessed for the resolution of complex tasks (Quinn and Bederson 2011). In this article we present a collective intelligence application from user-generated photographs, which relies on the fact that nowadays many photographs are geo-tagged to predict environmental variables. The availability of geo-tags in photos results from two current trends: first, the widespread habit of using the smartphone as a photo camera; second, the increasing number of digital cameras with an integrated GPS locator and Wi-Fi module. The impact of these two factors is that more and more personal photographs are being published on social portals and more and more of them are precisely geo-tagged (NPD Group 2011).

A time-stamped and geo-located photo can be regarded as the state of one or more objects at a certain time. From a collection of photographs of the same object in different moments, one can build a model of the object, study its behavior and evolution in time, and build predictive models of properties of interest.

The amount of available user generated media content on the Web nowadays is reaching unprecedented mass: Facebook alone hosts 240 billion photographs and gets 300 million new ones every day (Doherty and Smeaton 2010). This massive input allows for collective intelligence application of unprecedented reach and in innumerable domains, from smart city scenarios to land and environmental protection. However, a problem in the implementation of collective intelligence applications from visual user-generated content (UGC) is the identification of the objects of interest: each object may change its shape, position and appearance in the UGC, making its tracking in a set of billions of available

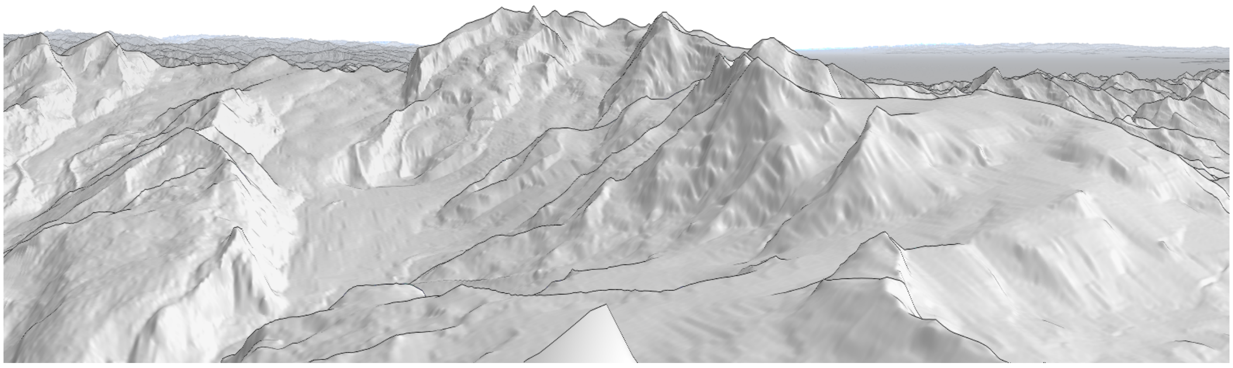

Figure 1: *An example of a rendered mountain view to be matched with the photograph.*

photographs an extremely difficult task. In this paper we aim at realizing applications that generate collective intelligence models from user-generated media content based on object extraction and model construction in a specific environmental sector: mountain conditions study. We harness the collective effort of people taking pictures of mountains from different positions and at different times of the year to produce models describing the states of selected mountains and their changing snow conditions over time. To achieve this objective, we need to address the object identification problem, which for mountains is more tractable than the general case described before, thanks to the fact that mountains are among the most motionless and immutable objects present on the planet.

1.1. Problem Statement

Given a user-generated geo-tagged and time-stamped photograph, the goal of this work is to determine whether the photo contains a mountain panorama, identify the visible mountain peaks and their corresponding edges, and then extract the visible part of the identified mountain.

1.2. Application Examples

The algorithm of peak detection we present can be used in applications where the purpose is to identify and tag mountains in photographs provided by users. The algorithm can also be used for the creation of mountain models and of their related properties, like, for example, the presence of snow at a given altitude.

Two representative examples of the first type of usage are:

- Mountain peaks tagging of user-uploaded photographs on photo sharing platforms that allow anyone browsing that photograph to view peak names of personal interest.
- Augmented reality on mobile devices with real-time peaks annotation on device screen while in camera mode.

An example of usage for environmental model building is the construction of a model for correcting ground and satellite based estimates of the Snow Water Equivalent on mountains peaks (which we describe in the Conclusions and Future Work section).

2. RELATED WORK

While several of recent papers has studied how to apply analyzing techniques to datasets of social multimedia in order to retrieve collective intelligence about the world and its changing in time (Lazer et al. 2009) only a small part of them deals with environmental disciplines, and another part faces the requirement of identifying the objects in the phogoraph. Jin et al. (2010) use photographs uploaded by users to detect the popularity and the people opinion (positive or negative) about the object captured in it. Zhang et al. (2012) use geo-tagged Flickr content to build an environmental model of the Earth with its snow and vegetation cover maps but without object identification procedure considering the planet itself at the same object to be processed in each photograph. Baboud et al. (2011) instead present a technique for moutains identification given a geo-tagged photograph focusing the results however on applications for annotation of the mountains on photo and video sources and not on environmental models building.

3. DATASETS AND MODELS

The proposed technique is used to match a photograph with a rendered mountain panorama from the location from which the picture was presumably taken. Therefore, the peak identification technique takes as inputs a model for generating rendered views of mountain panoramas and a photograph dataset.

3.1. Panoramas Renderer

The rendered image can be generated from digital elevation data sources. In our work we use a web tool "Panorama Generation"* that, given a geographical point, generates a rendered image of a mountain panorama (Figure 1) of the Alps, Pyrenees and Himalayas mountain range systems and the list of names of the peaks present in the panorama. [1]

3.2. Photographs Datasets

The input test data has been limited to photographs of the Alps and two test datasets were defined:

- A smaller dataset with around 40 photographs manually geo-tagged by their photographers.
- A larger dataset with 500 photographs extracted from photograph hosting services.

As sources for the second dataset we chose the Flickr and Panoramio sharing websites. On one hand, Flickr is one of the largest photo sharing platforms that allows queries combining geographical position and image title, with the possibility to retrieve the precision of the geo-tag of a given photograph. On the other hand, Panoramio provides only geographical position search but is a geolocation-oriented hosting service. All Panoramio photographs are supposed to be panoramic, so there is no need for querying the image titles to estimate the geo-location.

Considering the distinctive features of these datasets with respect to the accuracy of the geographical position of the photographs allows us to examine the importance of geotag precision and the impact of errors in positioning the observers point of view on result quality.

4. USER-GENERATED CONTENT ANALYSIS

The process of analyzing a single mountain panorama photograph in order to identify individual mountain peaks consists of several steps, shown in Figure 2. First the geo location associated with the user photo is extracted; next a 360-degree rendition of the panorama visible at that location is generated from the digital elevation model.

Next, the direction of the camera during the shot must be estimated so that each photograph pixel column can be associated with its degree orientation with respect to the camera position. Then, the mountain tagging is performed on the photograph by using the list of visible peaks on the rendered model.

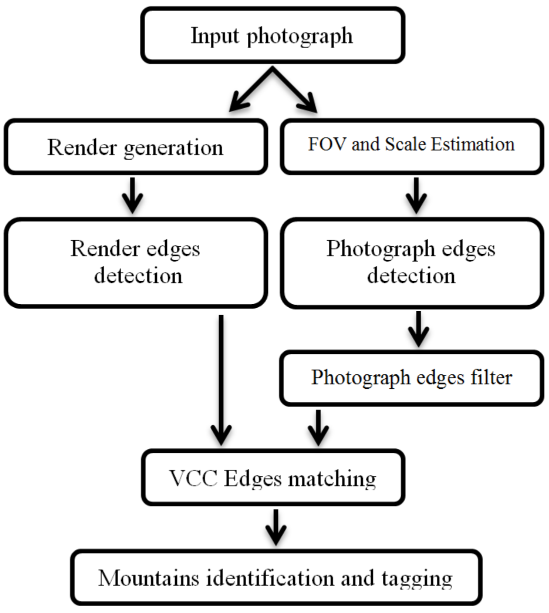

Figure 2: *Simple UGC process schema.*

4.1. Field Of View and Scale Factor Estimation

Given a user generated photograph and a synthetic rendered panorama image, it is necessary to scale the two images in such a way that the items (mountain systems in our case) have the same pixel sizes, in order to be matched with a non-scale-invariant matching algorithm. We can formulate this problem as that of making the ratio between the Field Of View (FOV) and the width of both images equal. This substantially reduces the problem to the estimation of the photograph FOV.

To estimate the photograph FOV we need the focus length (l) with which it was taken and the sensor width (w) of the digital camera that made the shot:

$$FOV = 2 \arctan \frac{w}{2l}$$

Once we have the photograph FOV, knowing that the FOV of the rendered panorama is 2π and assuming that the height/width ratio of the objects is the same both for the photograph and for the render (which depends on the quality of the renderer, and is actually true in our case), we can find the right scaling by a simple proportion. For example, if we want to keep the original render image (of width r) and adapt the photograph (of width p) to it, we should just scale the photograph by a factor of:

$$f = FOV \frac{r}{2\pi p}$$

[1]*`http://www.udeuschle.de/panoramen.html`

EXIF specification provides the focal length of a photograph but not the sensor details of the camera, so the algorithm processes only images with EXIF information about the focal length and the camera model, the sensor width of which is then searched for in an ad hoc developed database containing photo cameras sensor data.

Scaling estimation is a purely mathematical procedure, and the quality of the results depends directly on the precision of the geo-tag and the accuracy of the rendered model (with exact GPS location and a render based on correct elevation data the scaling produces perfectly matchable images).

4.2. Render and Photograph Edges Detection

The color appearance of the mountains can change significantly depending on the different times of the year and weather conditions, so standard color- and local-features-based image similarity scores are not adequate for our purposes. The most significant property of a mountain is its shape, and in order to match the shapes of the mountains we have to extract their edges.

We consider an edge map (both for photograph and for render view) as a 2D vector of complex numbers, where the absolute value of a number is equal to the edge strength in that point (in the range from 0 to 1) and the argument is the direction of the edge. In order to reduce noise, only the points above a certain absolute value threshold are considered. The association between the edge strength and its direction is used for edge matching.

4.3. Photograph Edges Filtering

Edge matching algorithms are developed to be robust to noise, so they tend to reduce the impact of noise edge points (edges of the photograph that do not belong to a mountain profile and so will not be present in the render view) on the matching results. Mountains, however, tend to be very edge-poor objects and most photographs have mountains in the background, with some other edge-rich objects in the foreground. In such cases, noise edges prevail over the actual ones.

This phenomenon is clearly illustrated in Figure 3, where, even with a quite clear photograph, 90% of all edge points are noise (the foreground is composed by a couple of rocks and the mountain reflection in the lake). This percentage grows up to 98% with the presence of detailed objects in the foreground, such as persons, buildings and trees. These levels of noise make mountain edges statistically insignificant even with the most sophisticated algorithms. Therefore, using some edge filtering process is an absolute necessity.

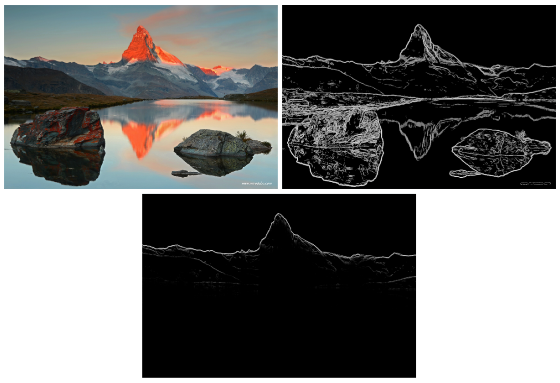
Figure 3: *Example of original edges (right) extracted from a photograph (left) and the result of edge filtering (bottom).*

We use a very simple but efficient heuristics based on an intuitive idea: mountain edges are placed usually on the top of the other edges. Therefore, the algorithm weighs each edge point by prioritizing the uppermost points. The main disadvantage of this method regards the effect of clouds highlighting. Fortunately, mountains clouds are edge-poor objects (usually only the boundary is extracted), so they produce little noise that can be managed well by the matching algorithm.

4.4. Vector Cross Correlation Edges Matching

The matching algorithm we use is based on the vector cross correlation (VCC) technique proposed inBaboud et al. (2011). The key idea of this technique is to find the best overlap of two rasterized images by considering the cosine similarity factor instead of simple correlation (Figure 4). The result of the edge detection algorithm is a 2D real-valued vector where each value is defined by ρ (the strength/absolute value of the edge in the corresponding point of the input image) and θ (the direction/argument of that edge). Let $p(\omega)$ and $f(\omega)$ be the 2D real-valued vectors generated by edge detection of the photograph and the panorama render respectively. We define the likelihood between two vectors as

$$\int_{S^2} M(f(\omega), p(\omega))d\omega$$

where $M(f,p)$ is the angular similarity operator:

$$M(f,p) = \rho_f^2 \rho_p^2 \cos 2(\theta_f - \theta_p)$$

This cosine factor is introduced in order to handle edge noise by penalizing differently oriented edges: the score contribution is maximum when the orientation is equal, null when they form a $\frac{\pi}{4}$ angle, and minimum negative when the edges are perpendicular. This penalty avoids that random noise edges contribute in a positive way to a wrong match position (a step in this direction was already made during the edge filtering).

Also a small simplification with respect to the original proposal was made. The authors worked in R^3 space, interpreting the two images as rasterized sphere images with the same radius rotating one inside the other, and the problem of matching as finding the three angles that define the right position of one sphere with respect to the other. We decided to simplify this model by removing one dimension due to the fact that almost all photos are shot perpendicular to the horizon with very small deviations. Therefore we consider that there is no need to rotate the photograph with respect to the render image and we interpret our model as two cylinders with the same radius, one inside the other, that can be rotated around their own axis and moved along the axis. Our problem in this way reduces to finding the right position, characterized by two values: the rotation of one cylinder with respect to another, and the shift between the axes. These values can be intuitively regarded as the coordinates of the photograph placed on the render image.

We define $\hat{f}$ and $\hat{p}$ as 2D Fourier transforms of respectively $f(\omega)$ and $p(\omega)$. The VCC computation equation becomes

$$M(f,p) = Re\{\hat{f}^2\bar{\hat{p}}^2\} \tag{1}$$

The removal of the third superfluous dimension reduces the computational effort incurred by VCC, which can now be efficiently computed with a 2D Fourier transform method. The algorithm is also responsible for determining if the best-found match is actually a correct one or the photograph cannot be properly matched to the render view.

If the best matching score is lower than a determined threshold the photograph is rejected as not containing mountain panorama or not processable.

4.5. Mountain Identification and Tagging

Once the photograph is matched to the render, the list of peaks present on the render with their names and coordinates allows us to identify the corresponding peaks on the photograph. Since there are usually some differences in edge maps between the photograph and the render, we do not simply tag the photograph with a peak in the point where the render supposes it to be, but we apply a further matching process.

Supposing that once the photograph is matched with the render view, the peak on the photograph (if present and visible) will be in proximity of the corresponding peak in the render, the following procedure is computed for each peak present in the render view. A fragment of edge maps (Figure 5) is extracted from both the photograph and the render, by weighing edge strength with a kernel function centered in the point where the render peak is placed. In this way, two fragments of edge maps in proximity of that point are obtained. Then, the same matching technique as before is applied: if the score is above a certain threshold, then the peak is identified and tagged; if not, the peak is considered not visible on the photograph. The use of a kernel function for the extraction of peak neighborhoods helps to position the tag as closely as possible to the exact point of the photograph peak.

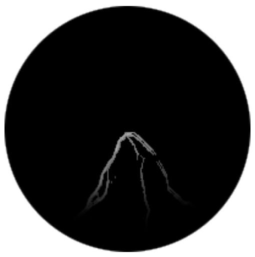

Figure 5: *An example of peak neighborhood edges extraction with a kernel weighting function.*

5. IMPLEMENTATION DETAILS

5.1. Panorama Render Generation

As mentioned, we use a web service for the generation of rendered panorama view with a zoom factor equal to 1, elevation exaggeration equal to 1, and altitude equal to $auto + 0$.

5.2. Edge Detection

For edge extraction we used the Compass (Ruzon and Tomasi 1999) component that, given an image, returns an edge strength 2D vector and a vector for the orientation of edges. The Compass algorithm is applied both to the photograph image and to the panorama render with standard deviation equal to 1. Once extracted, the edges are filtered, discarding those with strength less than 0.3 in both cases.

5.3. Edge Filtering

Each column vector of the photograph image is split in segments separated by non-edge pixels. If a segment contains more than k pixels, it is split in segments of k pixels. Each pixel belonging to the i-th segment is then multiplied by

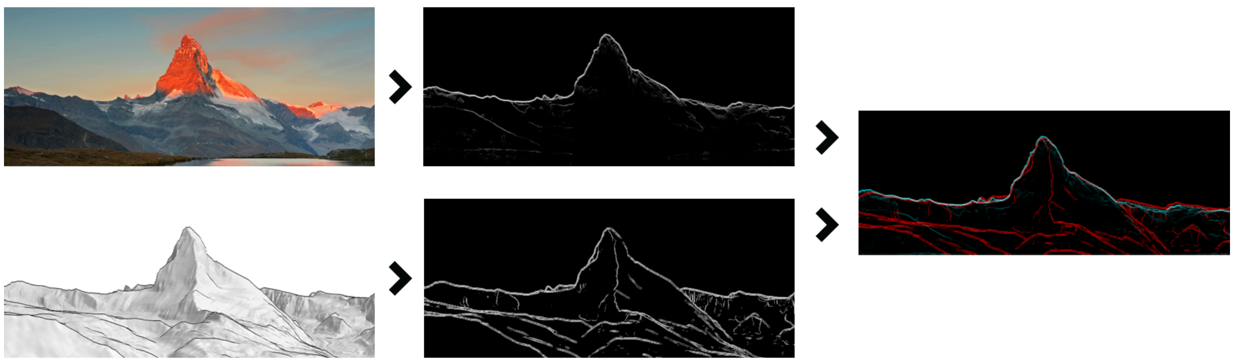

Figure 4: *An example of the VCC Matching: photograph is matched with mountain profiles in order to get the best overlap.*

$$b^{i-1}$$

We used $k = 2$, $b = 0.7$ chosen by trial and error method.

5.4. VCC Edges Matching

The edge matching is performed by applying Equation (1) using the 2D Fast Fourier Transform (FFT). Also, the scaling of the photograph does not always lead to perfect results. Therefore, due to the small computation effort of the 2D FFT, we tried different scale levels in the range between 80% and 120% with respect to the calculated scale factor, and choose the one that returns the best matching score.

5.5. Peaks Identification

Given d the distance between a point and the peak of the mountain and fixed r the radius of peak neighborhood, the triweight kernel function used for extracting the neighborhood edges around a mountain peak for all points such that $d < r$ is:

$$f(d) = \left(1 - \left(\frac{d^2}{r}\right)\right)^3$$

We use $r = 100$ chosen by trial and error method.

6. RESULTS

First, testing was performed on the dataset with manually precise geo-tagged photographs in order to assess matching correctness (i.e. it corresponds to a global maximum), and whether the correct match corresponds to a significant local maximum in the score array (by a significant local maximum we mean one that belongs to the set of top-N local maxima, for some N). Notice that the VCC matching algorithm is a fast and light technique that can be used for determining the best local maximum as possible matching positions and then checking them with more sophisticated heuristics like "Robust silhouette map matching metric" illustrated in Baboud et al. (2011) that gives better results but is much more computationally heavy and can't be applied to the whole search space. So we consider the presence of a score maximum (even if not the global one) in the right matching point as a positive result (Table 1).

Table 1: *Results of matching testing on the first dataset.*

Result	**Positive rate**
Correct match identified as global maximum	62%
Correct match identified among the top-10 local maximum	81%

The main reasons for matching failure is incorrect scale estimation due mainly to an inaccurate geo-tagging and a difference between the photograph and the model edges (for the same reason as wrong geo-tags). Tests performed on larger datasets crawled from photograph hosting services give significantly lower matching rates. Based on the previous observations, we conclude that the precision of geo-tags of photographs on sharing websites is nowadays quite low. A very common problem with manually tagged photographs (such as those from Panoramio) is that the users tag as the shot point the peak of the mountain, which is the subject of the photograph but not the actual shot point. Given a photograph with a correct geo-tag instead, the algorithm shows to be accurate and robust to noise like foreign objects placed in foreground or clouds and other noise due to the bad weather situation. Results also tend to be more accurate with increasing the FOV of the input photograph and with increasing the distance between the mountains and the camera.

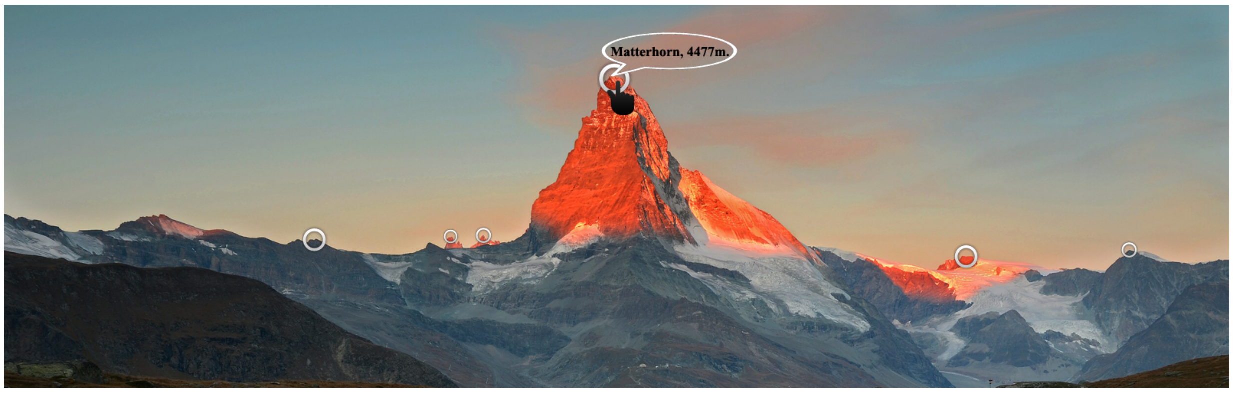

***Figure 6:** An example of the VCC Matching: photograph is matched with mountain profiles in order to get the best overlap.*

7. CONCLUSIONS AND FUTURE WORK

We developed and implemented a working algorithm that is able to recognize and tag mountain peaks on a photograph given its shot position with a good accuracy. The algorithm proved robust to noise edges and to irrelevant objects occluding the mountains such as people, trees and buildings in the foreground and bad atmospheric conditions such as the presence of clouds , but was not able to deal with significant geotag inaccuracies. A future line of work regards improving the robustness to geo-tagging errors, which could be tackled by introducing a robust silhouette map matching metric (Baboud et al. 2011) for finding the best match between some of the best positions proposed by VCC, or developing a technique robust with respect to inaccuracies in image scale estimation.

The precision of photograph geo-tags from hosting platforms was lower than we expected, but we assume that the problem will reduce significantly in future with the current increase of smartphones photographs and GPS photo cameras: in 2010, according to NPD Group (2011); Oracle (Oracle), 52% of the people declared that their smartphone would replace their digital camera, and 27% did it the same year, whereas, in 2011 43% said they actually replaced the photo camera with a mobile device.

7.1. Future Applications

The algorithm can be used on photo sharing websites to tag mountains on user photographs allowing not only a better user experience during photograph viewing (as exemplified in Figure 6), but a content-based search procedure that can find photographs containing a mountain even if its name is not specified in the title or description.

A promising use of our algorithm regards augmented reality applications for smartphones and mobile devices. This would indeed eliminate the problem about geo-tagging precision thanks to built-in GPS locators, consume little bandwidth (mountains are usually distant from the observer and the rendered view changes very slowly while the observer is moving, so it needs to be updated rarely), and the reduced computation capacity may be compensated by the built-in compass, which gives an indication of observers direction of view, so the matching procedure can be done only on a reduced fragment of the rendered panorama.

An interesting challenge and an opportunity to apply the research to environment-related problems is the calculation of Snow Water Equivalent (SWE): in snow cover dominated basins, the accurate estimate of the SWE, i.e., the amount of water contained in the snow pack, is key to improve the anticipation capability of decision making in operational flood control, water supply planning, and water resources management. SWE is usually estimated by spatial interpolation of ground-sensed point measurements of snow depth and density conditioned on the snow-covered area retrieved by satellite images processing. Since the middle of the 1960s, a number of satellite-derived snow products have been available to complement low-density snow monitoring networks, especially at inaccessible mountainous or high latitude regions. Satellite products, however, suffer from some technical limitations that hinder their operational value in most alpine contexts: space-board passive microwave radiometers (e.g. AMSR-E) easily penetrate clouds and provide accurate estimation, but work on very coarse spatial resolution. Optical sensors (e.g. MODIS) generate high-resolution snow cover maps, yet cannot see the earth surface when clouds are present. The algorithm of peak detection provides an additional source of information to be coupled with ground and remote sensing in producing high temporal (daily to weekly) and spatial resolution SWE time series. The snow covered area on the identified mountain can be estimated from the extracted visible part on the photograph to yield a snow cover estimate also for mountain peaks where satellite data are not applicable.

8. ACKNOWLEDGEMENTS

This work is supported by the EU FP7 CUbRIK Integrating Project, `http://www.cubrikproject.eu`.

Thanks to:

- Dr. Ulrich Deuschle (`http://www.udeuschle.de`) for permission of using his mountain panorama generating web tool.
- Gregor B. (`http://www.digicamdb.com`) for providing the database of digital cameras and their sensor sizes.
- Miroslav Sabo (`http://www.mirosabo.com`) for the photograph of the Matterhorn used in this article.

REFERENCES

Baboud, L., M. Cadik, E. Eisemann, and H.-P. Seidel (2011). Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), oral presentation*.

Doherty, A. R. and A. F. Smeaton (2010). Automatically augmenting lifelog events using pervasively generated content from millions of people.

Jin, X., A. Gallagher, L. Cao, J. Luo, and J. Han (2010, October). The Wisdom of Social Multimedia: Using Flickr for Prediction and Forecast. In *Proc. 2010 ACM Multimedia Int. Conf. (ACM-Multimedia 2010)*.

Lazer, D., A. Pentland, L. Adamic, S. Aral, and A. L. Barabasi (2009). Life in the network: the coming age of computational social science. In *Science*, pp. 721–723. Science.

NPD Group (2011). Consumers Now Take More Than a Quarter of All Photos and Videos on Smartphones. `http://www.npd.com/wps/portal/npd/us/news/press-releases/pr_111222/` (retrieved June 18, 2013).

Oracle. The Future of Mobile Communications Take Two. `http://www.oracle.com/us/industries/communications/oracle-communications-future-mobile-521589.pdf` (retrieved June 20, 2013).

Quinn, A. J. and B. B. Bederson (2011). Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pp. 1403–1412.

Ruzon, M. A. and C. Tomasi (1999). Color edge detection with the compass operator. In *CVPR*, pp. 2160–2166. IEEE Computer Society.

Zhang, H., M. Korayem, D. J. Crandall, and G. LeBuhn (2012). Mining photo-sharing websites to study ecological phenomena. In *Proceedings of the 21st international conference on World Wide Web*.