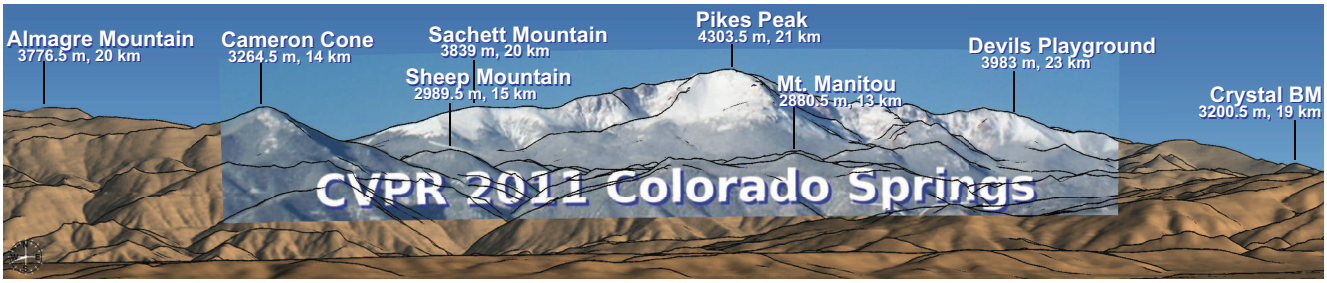
Automatic Photo-to-Terrain Alignment for the Annotation of Mountain Pictures

Lionel Baboud* Martin Čadík* Elmar Eisemann† Hans-Peter Seidel*

*Max-Planck Institute Informatik, †Telecom ParisTech/CNRS-LTCI



Abstract

*We present a system for the annotation and augmentation of mountain photographs. The key issue resides in the registration of a given photograph with a 3D georeferenced terrain model. Typical outdoor images contain little structural information, particularly mountain scenes whose aspect changes drastically across seasons and varying weather conditions. Existing approaches usually fail on such difficult scenarios. To avoid the burden of manual registration, we propose a novel automatic technique. Given only a viewpoint and FOV estimates, the technique is able to automatically derive the pose of the camera relative to the geometric terrain model. We make use of silhouette edges, which are among most reliable features that can be detected in the targeted situations. Using an edge detection algorithm, our technique then searches for the best match with silhouette edges rendered using the synthetic model. We develop a robust matching metric allowing us to cope with the inevitable noise affecting detected edges (*e.g. *due to clouds, snow, rocks, forests, or any phenomenon not encoded in the digital model). Once registered against the model, photographs can easily be augmented with annotations (*e.g. *topographic data, peak names, paths), which would otherwise imply a tedious fusion process. We further illustrate various other applications, such as 3D model-assisted image enhancement, or, inversely, texturing of digital models.*

*{lbaboud, mcadik, hpseidel}@mpi-inf.mpg.de

†elmar.eisemann@telecom-paristech.fr

1. Introduction

The internet offers a wealth of audio-visual content and communities such as Flickr and YouTube make large amounts of photos and videos publicly available. In many cases, an observer might wonder what elements are visible on a certain shot or movie. Especially for natural scenes, the answer to this question can be difficult because only few landmarks might be easily recognizable by non experts. While the information about the camera position is (at least roughly) known in many cases (photographer's knowledge or camera GPS), the camera orientation is usually unknown (digital compasses have poor accuracy).

The principal requirement is then the accurate alignment (registration) of a given photograph or video with a 3D georeferenced terrain model. Interestingly, such a precise localization would be useful in many contexts. Services such as Google StreetView could be extended in an automatic fashion to natural environments by exploiting user-provided shots. Further, the photo can be used to texture virtual terrains such as those in Google Earth. Also, annotations, derived from an annotated 3D terrain model, could be added automatically (highlighting important landmarks) which is of interest when describing or planning a field trip. Because of such applications, cameras start being equipped with GPS in order to automatically track photo locations.

We will focus on a special class of content taken in mountain regions, and provide a solution to automatically derive the orientation that was used for a given shot, assuming that the viewpoint location is known accurately enough, as well as the cameras's intrinsic parameters (*e.g.* field-of-view). It is often complicated or even impossible to access

these regions with cars or robots, making user-provided images an interesting way to collect data. Furthermore, users also benefit from our solution, as it enables them to enhance (and even augment) their photos with supplementary data.

The input of our approach is a single photograph or a video and an indication of where it was taken. Our algorithm then automatically finds the view direction by querying the position against a reference terrain model that we assume to have at disposition. The latter is a smaller constraint because satellites can provide very reliable terrain elevation maps even for less accessible regions. Once the view is matched, we can transfer information from the reference model into the photo.

Our main contribution is the robust matching algorithm to successfully find the view orientation of given photo. This task is far from trivial and many previous approaches attempting to match up an image and 3D content can exhibit high failure rates (Section 2). The reason why our algorithm (Section 3) provides a working solution is that we can exploit the special nature of terrains. Mountain silhouettes are relatively invariant under illumination changes, seasonal influence, and even quality of the camera, therefore we detect these features and make them a major ingredient in our matching metric (Sections 4, 5, 6). Finally, we illustrate the robustness and usefulness of our approach with several of the aforementioned application scenarios (Section 7) before concluding (Section 8).

2. Previous Work

The problem of matching appears in several areas of research, but proves difficult in most cases. Advances in camera engineering (i.e. digital compass and GPS receivers) can facilitate the task in the future, but such data is neither available in most current cameras nor present in video sequences. Furthermore, even when available, such information is not reliable enough for an accurate pose estimation and will not be in a long time because the satellite infrastructure would need to change drastically to allow the precision we seek. Usually, existing GPS and compass-based applications only present distant abstracted depictions (*e.g.* Peakfinder (http://peakfinder.ch), Google Skymap) without considering the actual view content. The same holds for augmented reality applications, such as the Wikitude World Browser (http://www.wikitude.org). In a reasonable time frame only initial estimates of a camera pose, but not the final fine-tune registration will be available. In the context we target, orientation must be known accurately to properly discriminate distant peaks, wereas position accuracy is less crucial (negligible parallax).

Registration comes in many variants, usually, instead of matching an entire image, a first step is to restrict the search to a small set of feature points. Such feature-based (SIFT [13], SURF [1]) techniques work robustly for image to image registration, but are less usable for image-to-model registration [23]. Nonetheless, for applications such as panorama stitching [19], feature-based techniques work well and currently dominate. Unfortunately, our case is more difficult because we have to consider very differing views in a natural scene which exhibits many similar features or features that might depend heavily on the time of the year (*e.g.* snow borders). This constraint also renders statistical methods [24], that are widely used in medical image registration, less successful.

The difficulty of this task is also underlined in the photo-tourism approach [17]. Indoor scenes and landmark shots are handled automatically, while outdoor scenes have to be aligned against a digital elevation map and a user has to manually specify correspondences and similarity transforms to initiate an alignment. Similarly, Deep Photo [9] requires manual registration and the user has to specify four or more corresponding pairs of points.

In our experience, even simpler tasks, such as horizon estimation [6], tend to fail in mountain scenes. Similarly, advanced segmentation techniques [7, 16] proved futile. Maybe for these reasons, existing photogrammetry approaches for mountain imagery, such as GIPFEL (http://flpsed.org/gipfel.html), strongly rely on user intervention.

Robust orientation estimation is a necessary component of localization algorithms for autonomous robots. During missions on moon or mars, it is impossible to rely on standard GPS techniques, but satellite imagery can deliver a terrain model. Many of these algorithms rely on the horizon line contour (HLC) which is the outline of the terrain and specific feature points thereon that are matched with extracted terrain features [2, 21, 8]. Peaks of the HLC are often used as features, but might not correspond to actual peaks in the terrain due to partial occlusion (clouds, fog, or haze), terrain texture (*e.g.* snow), or an incorrect sky detection (see Fig. 3). The latter is very difficult, but particularly crucial for HLC approaches, especially when estimating visibility between peaks in the query image [8]. Learning techniques [8, 14] can often lead to successful segmentations, but they depend on the training set and implicitly assume similar query images (*e.g.* same daytime). Furthermore, even if successful, the localization of peaks in a photo is error prone [2] and can lead to a deviation in the estimate. Hence, sometimes only virtual views are tested [21], or an accurate compass is supposed [20].

Instead of peaks, using all occluding contours leads to more robustness, but previous solutions [18] needed an accurate orientation estimate and assumed that the query image allows us to well-detect all occluding contours. As for the HLC, this property rarely holds because haze, fog or lighting variations often occlude crucial features. Our approach does not penalize missing contours, and the detec-

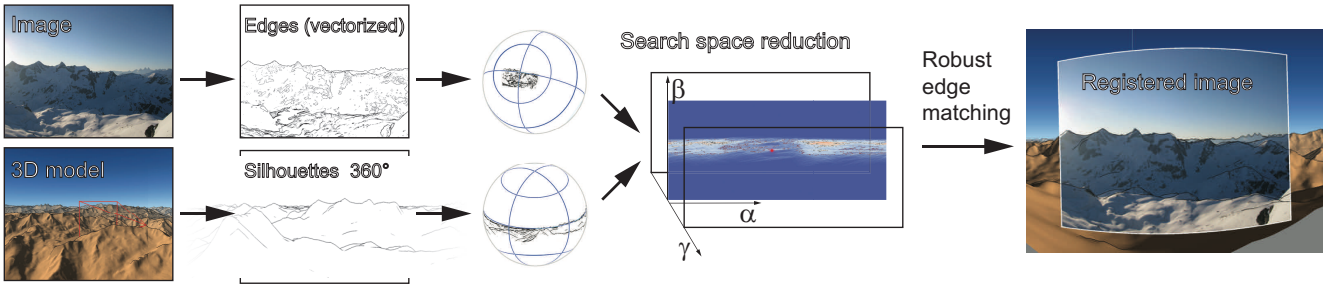


Figure 1. Overview of the proposed technique.

tion robustness does not suffer from false positives.

Interestingly, despite their negative effect on contour detection, haze and fog potentially encode monocular depth information [3] The assumptions on reflectance properties and fog/haze are relatively general and some assumptions such as a ground plane [3] fail in our context. Consequently, the resulting depth estimates are usually coarse and proved insufficient for our purposes.

The area of direct image to model registration is less developed, and most techniques assume some structural elements (*e.g.* straight lines, planes) in the input image [10, 5]. Unfortunately, mountain scenes are highly unstructured making matching very challenging which lead us to develop our approach.

3. Problem setup

Given a photograph, our goal is to estimate its pose relatively to an accurate 3D terrain model based on a digital elevation map (DEM). We assume that the camera's field of view is known, as well as an estimate p_v of the viewpoint position (accuracy is discussed in Section 7). Given these hypotheses, we are looking for the rotation $\tilde{g} \in SO(3)$ that maps the camera frame to the frame of the terrain. The set of images that can be shot from p_v is entirely defined by a spherical image f centered at p_v against which we need to match the query photo.

We target outdoor scenes that do not allow to rely on photogrammetry information, as it can vary drastically. Instead, we rely on silhouette edges that can be obtained easily from the terrain model and can be (partially) detected in the photograph. In general, the detected silhouette map can be error prone, but we enable a robust silhouette matching by introducing a novel metric (Section 4).

Because a direct extensive search on $SO(3)$ using this metric is very costly, we additionally propose a fast preprocess based on spherical cross-correlation (Section 5). It effectively reduces the search space to a very narrow subset, to which the robust matching metric is then applied. The resulting algorithm is outlined in Fig. 1.

3.1. Spherical parameterization

We start by defining some basic notations. The camera frame has its Z axis pointing opposite to the viewing direction, with X (resp. Y) axis parallel to the horizontal (resp. vertical) axis of the image. The terrain frame has its Z axis along the vertical. Rotations of $SO(3)$ are parameterized with the ZYZ Euler angles, *i.e.* an element $g \in SO(3)$ is represented by three angles (α, β, γ) so that $g = R_Z(\alpha)R_Y(\beta)R_Z(\gamma)$, where R_Y and R_Z are rotations around axes Y and Z.

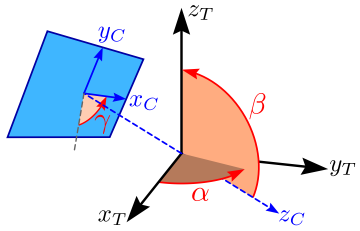


Figure 2. Terrain (x_T, y_T, z_T) and camera (x_C, y_C, z_C) frames.

The synthetic spherical image of the terrain model from p_v will be denoted f, and the spherical representation of the photograph will be denoted p. The corresponding silhouette sets will be denoted $\mathcal{E}_F$ and $\mathcal{E}_P$.

4. Robust silhouette map matching metric

We first address the more costly, but precise fine-matching. In the targeted situation, *i.e.* on photographs of mountainous scenes, results produced by available edge-detection techniques usually contain inaccuracies which can be classified as following (see also Fig. 3):

- some of the silhouette edges are not detected;
- some detected edges are noisy;
- many detected edges are not silhouette edges.

The noisy edges prevent us from using traditional edge matching techniques that often rely on features that are assumed to be present in both images. However the specificity of our problem allows us to derive a robust matching metric.

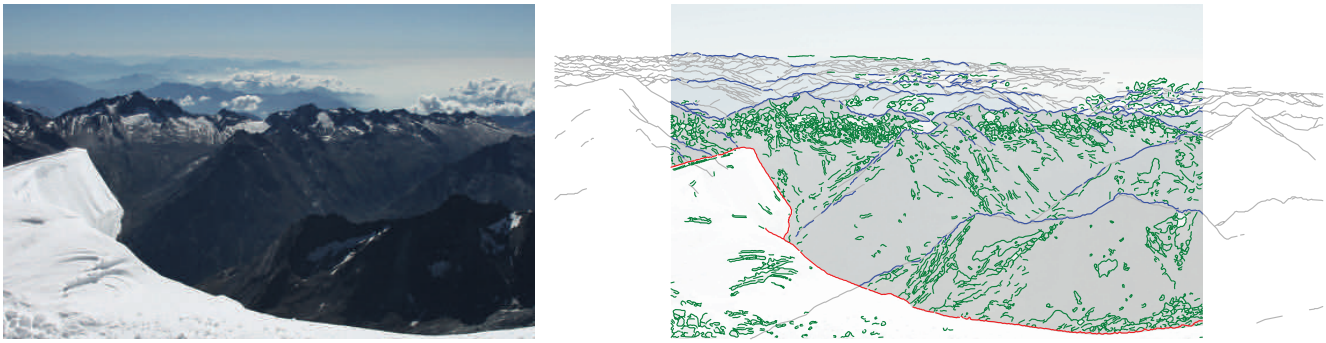


Figure 3. Types of edges detected in mountain scenes: silhouettes encoded (blue) or not encoded in the terrain model (red), noise and non-silhouette edges (green). Reference (*i.e.* synthetic) silhouettes (gray) are not always detected.

Our main observation relates to the topology of silhouette-maps: a feasible silhouette map in general configuration can contain T-junctions, but no crossings. Crossings appear only in singular views, when two distinct silhouette edges align (Fig. 4). Consequently, a curve detected as an edge in the photograph, even if not silhouette, usually follows a feature of some object and thus never crosses a silhouette. This only happens if some object, not encoded in the terrain model, occludes it. The probability for such events remains low, which will render the method more robust despite potentially low-quality edges.

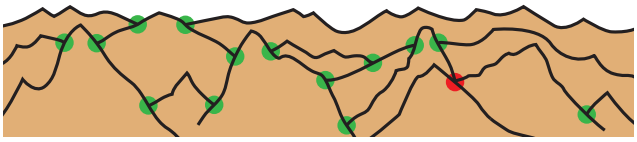


Figure 4. Specific topology of terrain silhouettes: separate edges meet with T-junctions (green), crossings (red) are singular.

To evaluate the likelihood of a given orientation g, the two edge sets (from photo and model) are overlayed according to g. Each edge e_p from $\mathcal{E}_P$ is considered independently and tested against $\mathcal{E}_F$. To account for noise, any potential matching with an edge e_f must be scanned within some tolerance ε_e. When e_p enters the ε_e-neighborhood of an edge $e_f \in \mathcal{E}_F$, four distinct cases can happen, as depicted by Fig 5. A threshold ℓ_{fit} is used to distinguish the case where e_p is following e_f from the case where it crosses it.

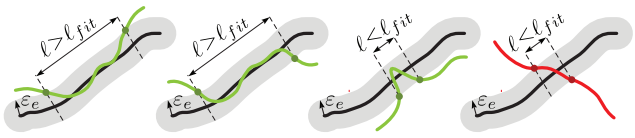


Figure 5. The four possible situations for edge-to-edge matching.

For a single edge e_p, the matching likelihood value is computed as follows:

- parts where e_p stays outside the ε_e-neighborhood of elements of $\mathcal{E}_F$ count as 0;
- if e_p enters the ε_e-neighborhood of an element $e_f \in \mathcal{E}_F$ and exits it after traversing over a length ℓ:
  - if it exits on the same side or if $\ell \geqslant \ell_{fit}$, the fitting energy $\ell^{a_{fit}}$ is added;
  - else, a constant penalty cost c_{cross} is subtracted.

The non-linearity implied by an exponent $a_{fit} > 1$ increases robustness: long matching edges will receive more strength than sets of small disconnected segments of the same total length. Finally, the matching likelihood for $\mathcal{E}_P$ under the candidate rotation g is obtained by summing the values of each of the (accordingly displaced) individual edges of $\mathcal{E}_P$.

In practice the computation is performed as follows: first, $\mathcal{E}_F$ is rasterized with a thickness ε_e into a sufficiently high-resolution spherical image; second, the $\mathcal{E}_P$ edges are warped according to g, traversed and tested against the rasterized $\mathcal{E}_F$ for potential intersections. The cost of this simple approach is $O(mn)$, where m is the resolution of the rasterized $\mathcal{E}_F$ and n the total number of segments of $\mathcal{E}_P$.

Interestingly, the metric relies on all the information available in the detected edges: even non-silhouette edges help to find the correct match by preventing actual silhouette edges from crossing them (Fig. 6). Therefore it would theoretically be possible to find the correct matches even if all silhouette edges were missed.

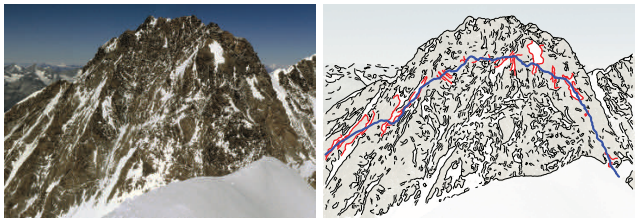


Figure 6. Detected non-silhouette edges also help the matching process: a pose of a reference silhouette (blue) is prevented if it crosses many detected edges (red).

Although this metric allows robust matching (see Section 7), it requires a dense 3D sampling of $SO(3)$, leading to

prohibitive computation times. We avoid this problem with an effective search space reduction preprocess, presented now.

5. Spherical cross-correlation for search space reduction

To address the problem of the high cost implied by a dense sampling of $SO(3)$, we move to the Fourier domain. It is well known that the cross-correlation between two $n \times n$ images can be computed in $O(n^2 \log n)$ using the fast fourier transform (FFT). This has recently been extended to spherical images [11]. The spherical cross-correlation of two complex-valued spherical functions f and p is defined on $SO(3)$ as:

$$\forall g \in SO(3), \quad f \star p\,(g) = \int_{\mathcal{S}^2} f(\omega)\overline{p(g^{-1}\omega)}d\omega,$$

and can be evaluated in $O(n^3 log(n))$ for n^2-sampled spherical functions via FFT algorithms on $SO(3)$ [11].

We could directly apply this to our problem by sampling the two silhouette-maps on the sphere and computing the cross-correlation of these two binary-valued maps (1 on edges, 0 elsewhere). The main problem here is that it completely disregards the relative orientation of edges. With our noise-prone detected edge-maps, the maximum cross-correlation value would be found where most edges overlap, which would only work if the detected edge-map contained all and only the silhouette edges.

5.1. Angular similarity operator

Our goal is to integrate edge orientations in the cross-correlation. The orientation information can be kept by rasterizing $\mathcal{E}_F$ as a 2D real-valued vector field $\mathbf{f}(\omega) = (f_x(\omega), f_y(\omega))$, being the tangent vectors of the edges where they appear, and zero elsewhere (Fig. 8). We define the *angular similarity operator* $\mathcal{M}(\mathbf{f}, \mathbf{p})$ as follows:

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

where (ρ_f, θ_f) and (ρ_p, θ_p) are the polar representations of $\mathbf{f}$ and $\mathbf{p}$ (see Fig. 7). The value produced by this operator is:

1. positive for (close to) parallel vectors,
2. negative for (close to) orthogonal vectors,
3. zero if one of the vector is zero.

The matching likelihood between two spherical functions $\mathbf{f}$ and $\mathbf{p}$ can be expressed as:

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

so that values of ω where edges closely match are counted positively while those where edges cross almost perpendicularly are counted negatively. Furthermore, values of ω where either $\mathbf{f}$ or $\mathbf{p}$ has no edge do not affect the integral.

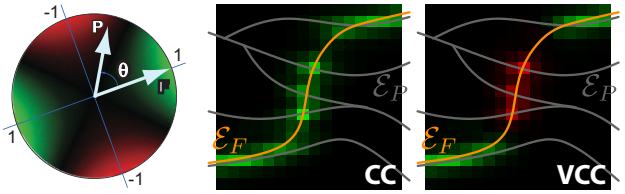


Figure 7. Left: $\mathcal{M}(\mathbf{f}, \mathbf{p})$ as a function of $\mathbf{p}$ (for a fixed $\mathbf{f}$). Classical cross-correlation (CC) disregards orientations, wereas our vector-field cross-correlation (VCC) properly penalizes crossings.

5.2. Spherical 2D-vector fields cross correlation

In order to be used as a matching likelihood estimation, this integral would need to be evaluated for any candidate rotation g, by rotating $\mathbf{p}$ accordingly. However, now that $\mathbf{p}$ values are vectors, we need to take the effect of the rotation into account. Because we defined the transformation of the camera relative to the world frame, we can show that the expression of $\mathbf{p}$ under a rotation $g = (\alpha, \beta, \gamma)$ is:

$$R_{\gamma+\frac{\pi}{2}}.\mathbf{p}(g^{-1}w) \quad \text{with} \quad R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

The formula stems from the fact that in the ZYZ euler angles parametrization we are using, the γ angle corresponds to the rotation of the camera around its viewing direction (the $\frac{\pi}{2}$ offset reflects that a horizontally-looking camera with a zero γ value is tilted by $\frac{\pi}{2}$).

Our operator needs to be modified as follows to take the rotation of $\mathbf{p}$ into account:

$$\mathcal{M}_g(\mathbf{f}, \mathbf{p}) = \rho_f^2 \rho_p^2 \cos 2(\theta_f - (\theta_p + \gamma + \frac{\pi}{2})),$$

and for a candidate rotation g we then define the matching likelihood between $\mathbf{f}$ and $\mathbf{p}$ as follows:

$$\text{VCC}(\mathbf{f}, \mathbf{p})(g) = \int_{\mathcal{S}^2} \mathcal{M}_g(\mathbf{f}(\omega), \mathbf{p}(g^{-1}\omega))d\omega.$$

5.3. Efficient computation

Using the representation of 2D vectors as complex numbers, VCC can be expressed as one spherical cross-correlation operation. Indeed, $\mathcal{M}(\mathbf{f}, \mathbf{p})$ can be rewritten as follows,

$$\mathcal{M}(\mathbf{f}, \mathbf{p}) = Re\left\{\hat{f}^2\overline{\hat{p}^2}\right\},$$

where

$$\hat{f} = \rho_f e^{i\theta_f} \quad \text{and} \quad \hat{p} = \rho_p e^{i\theta_p}.$$

This leads to the following VCC formulation:

$$\begin{aligned}\text{VCC}(f, p)(g) &= Re\left\{\int_{\mathcal{S}^2} \hat{f}^2(\omega)\overline{\left(e^{i(\gamma+\frac{\pi}{2})}\hat{p}(g^{-1}\omega)\right)^2}d\omega\right\} \\ &= -Re\left\{e^{-i2\gamma}\ \hat{f}^2 \star \hat{p}^2\ (g)\right\}.\end{aligned}$$

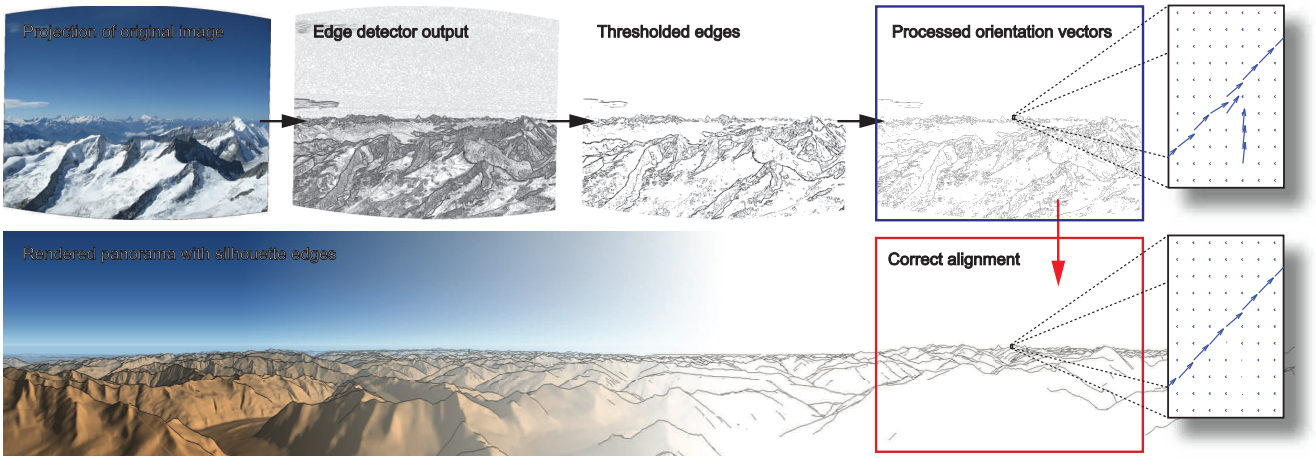


Figure 8. Detection and processing of edges into orientation vectors (blue frame), used to find the optimal registration (red frame).

In other words, we expressed the computation as a cross-correlation between $\hat{f}^2$ and $\hat{p}^2$, that is weighted by $-e^{-i2\gamma}$ and reduced to its real part. The dominant cost of the matching space reduction is therefore the cross-correlation computation, *i.e.* $O(n^3 \log n)$.

6. Implementation details

Terrain model We experimented with two terrain datasets: 1) coverage of the Alps with 24 meters spaced samples (http://www.viewfinderpanoramas.org); 2) National Elevation Dataset (USGS, http://ned.usgs.gov), covering the United States at thrice bigger resolution. Experiments showed the importance of considering the Earth's curvature when rendering the synthetic panoramas.

Image processing The input photograph is first remapped to a rectilinearly projected RGB image with known FOV, using the camera's intrinsic parameters (read from the attached EXIF data, assisted by a camera database if necessary). We then apply the *compass* edge detector [15], parameterized by a radius σ, producing separate maps for edge strengths (Fig 3) and orientations, that are easily combined into a vector field of tangent vectors (Fig. 8). This edge detector has the particularity of fully exploiting the color information, unlike classical ones that handle only grayscale images. The result is then thresholded (parameter τ) to keep only significant edge. The edge map $\mathcal{E}_P$ (a set of vectorized lines) is finally extracted by thinning [12] and vectorization [4]. The following parameters were used without further need of dynamic adaptation: $\sigma = 1$, $\tau = 0.7$.

Panorama processing Generating silhouettes from the 3D terrain data is a classical computer graphics problem for which several options exist. Exploiting the GPU, we apply raycasting to render the silhouettes into a 2D cylindrical image, which is then vectorized into an edge map $\mathcal{E}_F$.

Efficient matching Because $SO(3)$ has three dimensions, the robust matching metric still needs to be evaluated on many sampled rotation candidates, even after the search space reduction process. Nonetheless, each evaluation being independent, the overall process is highly parallelizable making a GPU mapping possible that cuts down the computation time from several hours to a few seconds.

7. Results

Our approach was implemented on a Dell T7500 workstation equipped with two six-core Intel Xeon processors, one GeForce GTX 480 GPU, and 23GB RAM. With our simple implementation, the overall process takes around 2 minutes, critical parts being compass edge detection (around 1 min.), spherical cross-correlation (less than one minute, with sampling bandwidths of 1024 for $\mathcal{S}^2$ and 256 for $SO(3)$) and final matching metric evaluation (around 20 s. with the GPU implementation). Of a collection containing 28 photographs randomly chosen from Flickr, 86% were correctly aligned by our technique (interestingly, VCC was already maximized at the correct orientation for 25% of the tested examples). We examined two different mountainous regions (Alps in Europe and Rocky Mountains in USA) and found that our approach performs similarly. The matching is generally very accurate, *i.e.* below $0.2°$ (Fig. 1, 9 and 10). Small deviations mostly correspond to imperfections of the 3D model. Experimentally, an accuracy below a few hundred meters for the viewpoint is sufficient.

7.1. Applications

Annotations Our solution enables us to mark a certain peak in all given photos if it is visible. This is a difficult and tedious task that often can only be performed by experts.

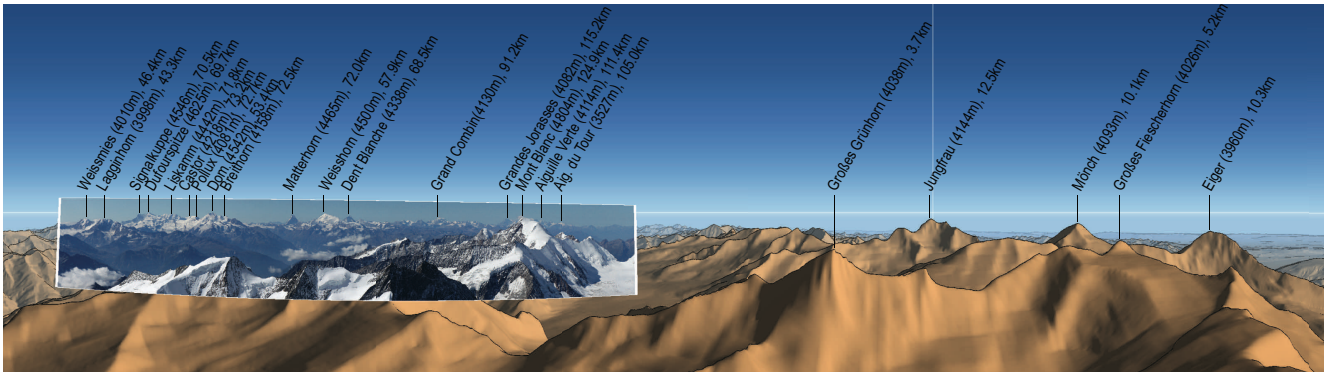


Figure 9. An example of annotated panorama image superposed on synthetic panorama.

By testing the visibility of the corresponding mountain in the 3D terrain model, we can easily decide what part of it shows in the photograph, and how far it is from the camera position. Some results are illustrated in Fig. 9 and 10.

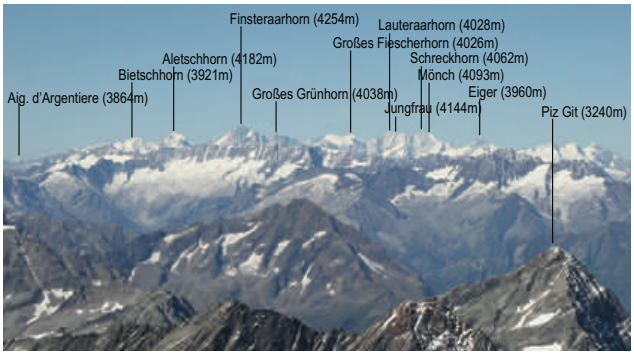


Figure 10. Annotated photo created using the proposed technique.

Augmented reality We can also achieve augmented views of the mountain landscape. Here, we add paths, landmarks and other 3D objects into the 3D terrain model. By transferring only the visible pixels of these models, we can add them into the photograph. Furthermore, we can relight them according to the shot. For this, we can either rely on the time stamp of the photo to deduce the position of the sun and weather conditions from according databases. Alternatively, we can optimize the sun position and illumination by comparing the lit terrain model to the captured photo. We rely on a simple model with a point light (sun) and ambient occlusion (sky). The optimization process is 1D and converges quickly.

Texture transfer Using our approach, photo collections can also easily be used to transfer texture information into a 3D mountain model such as those of Google Earth. Having found the corresponding camera view, it is enough to apply a projective texture mapping (including a shadow map test) to derive which part of the scene was actually visible and could benefit from the image content.

Photo navigation Similarly to photo tourism [17], we can add the photos into the 3D terrain model to enable an intuitive navigation. This allows illustrating or preparing hikes, even when relying on photos of others.

Image Enhancement and Expressive Rendering Using the underlying 3D terrain model, we can enhance an existing image or achieve non-photorealistic effects. E.g., we can perform informed model-based image dehazing (Fig. 11), enhance certain objects, or even mix the view with geological data (*e.g.* using USGS metadata).

Video Matching On a frame-by-frame basis, we can also optimize video sequences. Which is relatively fast because the search space is reasonably reduced by assuming a slow displacement. One could also initialize the search with the frame that gave the highest response in the first search step, but in practice, we found that unnecessary [1].

8. Conclusions and Future Work

We presented a solution to determine the orientation of mountain photographs by exploiting available digital elevation data. Although this is a very challenging task, we showed that our approach delivers a robust and precise result. The accuracy of our solution enabled various interesting applications that we presented in this paper. Our technical contributions, such as the camera pose estimation based on edge-to-silhouette matching could find application in other contexts of more general matching problems.

In the future, we want to explore other cues (*e.g.* the atmospherical scattering, aerial perspective) that might help us in addressing more general environments and improving the edge detection part for these scenarios [22].

[1] Refer to supplemental movie for video matching examples.

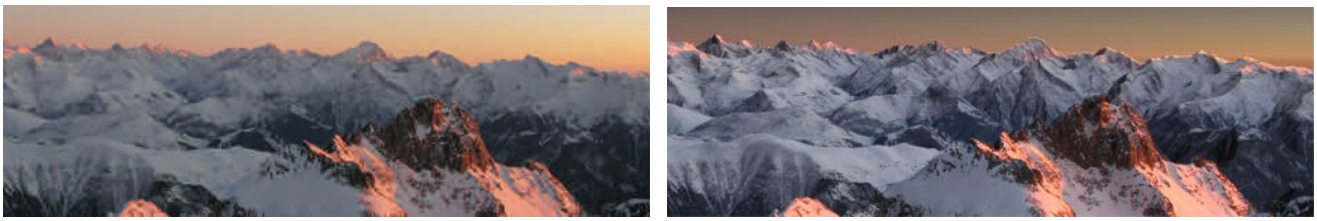


Figure 11. Application to image contrast enhancement: the original image (left) is modulated by the diffuse lighting component computed on the synthetic model (particularly profitable for distant mountains, whose contrast is affected by atmospheric effects).

Acknowledgements

Thanks to YouTube user 'towatzek' for his videos (http://www.mounteverest.at), Wikipedia user 'Nholifield' for the Mount Princeton panorama, and ColoradoGuy.com for pictures from Rockies. This work was partially funded by the Intel Visual Computing Institute (Saarland University).

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia. 42

[2] F. Cozman and E. Krotkov. Position estimation from outdoor visual landmarks for teleoperation of lunar rovers. In *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision*, Washington, DC, USA, 1996. IEEE Computer Society. 42

[3] R. Fattal. Single image dehazing. *ACM Trans. Graph.*, 27:72:1–72:9, August 2008. 43

[4] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003. 46

[5] R. Grzeszczuk, J. Kosecka, R. Vedantham, and H. Hile. Creating compact architectural models by geo-registering image collections. pages 1718 –1725, sep. 2009. 43

[6] C. Herdtweck and C. Wallraven. Horizon estimation: perceptual and computational experiments. In *APGV '10: Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 49–56, New York, NY, USA, 2010. ACM. 42

[7] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, 2005. 42

[8] P. C. N. Jr, M. Mukunoki, M. Minoh, and K. Ikeda. Estimating camera position and orientation from geographical map and mountain image. In *38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*, pages 9–16, 1997. 42

[9] J. Kopf, B. Neubert, B. Chen, M. F. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, 27(5):116:1–116:10, 2008. 42

[10] J. Kosecka and W. Zhang. Video compass. In *Proceedings of European Conference on Computer Vision*, pages 657 – 673, 2002. 43

[11] P. Kostelec and D. Rockmore. Ffts on the rotation group. *Journal of Fourier Analysis and Applications*, 14:145–179, 2008. 10.1007/s00041-008-9013-5. 45

[12] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies-a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:869–885, September 1992. 46

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 10.1023/B:VISI.0000029664.99615.94. 42

[14] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand. Geolocalization using skylines from omni-images. In *Proceedings of the IEEE Workshop on Search in 3D and Video, Kyoto, Japan*, oct 2009. 42

[15] M. A. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1281–1295, 2001. 46

[16] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 31:824–840, 2009. 42

[17] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press. 42, 47

[18] F. Stein and G. Medioni. Map-based localization using the panoramic horizon. In *Robotics and Automation, IEEE Transactions on*, pages 892 – 896, 1995. 42

[19] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006. 42

[20] R. Talluri and J. K. Aggarwal. Handbook of pattern recognition & computer vision. chapter Position estimation techniques for an autonomous mobile robot: a review, pages 769–801. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1993. 42

[21] J. Woo, K. Son, T. Li, G. S. Kim, and I.-S. Kweon. Vision-based uav navigation in mountain area. In *MVA*, pages 236–239, 2007. 42

[22] C. Zhou and B. W. Mel. Cue combination and color edge detection in natural scenes. *Journal of vision*, 8(4), 2008. 47

[23] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977 – 1000, 2003. 42

[24] L. Zollei, J. Fisher, and W. Wells. An introduction to statistical methods of medical image registration. In *Handbook of Mathematical Models in Computer Vision*. Springer, 2005. 42