Social Tourism Using Photos

Mark Lenz

Abstract

Mobile phones are frequently used to determine one's location and navigation directions, and they can also provide a platform for connecting tourists with each other. This paper proposes a system that uses a set of geotagged photos to automatically compute the geographic location of another photo and then augment that photo with text and images that enable collaboration and enhance navigation. The system breaks up the world into smaller regions and then computes a geographic location using one method and a complete camera pose using a more complex method, both methods utilize robust local feature matching. Tourists can then add text and images to objects in a photo which are then augmented onto other photos viewing the same object, allowing them to share information linked to specific objects in their environment. Navigation information in the form of arrows are projected onto photos pointing tourists in the direction of their selected destination. This system is particularly applicable to places such as malls, museums, and theme parks where photos can provide more information than GPS, but it can be applied anywhere.

1 Introduction

More people are using mobile phones to retrieve location-based information as more GPS-enabled mobile phones have become available. Nearly every mobile phone also comes equipped with a camera. Automated methods of retrieving geotagged photos from camera-equipped vehicles as well as Internet sites such as Google Street View and Flickr provide large photo sets that can be used with computer vision techniques to automatically create models which can provide information beyond what is provided by GPS.

Using this additional information the proposed system enables linking text and images to points in the 3D world that are then displayed on new photos taken of the same points. This allows tourists to collaborate using visual information linked to the viewed scene. So visitors to a museum can share their opinions or knowledge about a particular artwork, and a custom application or external database can provide additional contextual information such as the history of a particular piece of art. Popular places of interest as well as the path traveled can be mapped by storing the location photos are taken and what is photographed. The system is further enhanced by overlaying photos with navigation information in the form of arrows that direct tourists to their destination.

Consider the example of a theme park where visitors use their mobile phones to navigate from one desired ride to the next. Using the proposed system visitors could get a more precise location than GPS, as well as the ability to easily view and link semantic information such as ride experiences with a particular part of the ride. Visitors take photos to determine their location and then the photo can be augmented with graffiti from other visitors. If the visitor selects a particular destination the overlay can include an arrow pointing in the direction of the destination. The proposed system eases the task of navigating unknown territory and enables the sharing of experiences while traversing the environment.

2 Related Work

Several techniques have recently been developed for estimating the geographic location a given photo was taken with varying approaches and results. Im2gps [6] uses online photo collections to create a model that produces a distribution of the likelihood of the geographic location of a photo based on a scene matching algorithm. The scene matching algorithm attempts to match several different image properties such as color and texton histograms, gist descriptors [13], and the geometric context of a photo to those of the photos in the model. Instead of matching global scene information the proposed system uses local feature descriptors which can provide a more precise matching between views of the same scene as well as distinctive points in the scene which can be used to easily augment the photo. Im2gps was the winner of the "Where am I?" Contest [16] at ICCV'05 where many other algorithms were introduced for location estimation and recognition, but Johansson and Cipolla [8] developed an approach similar to the proposed system using local planar features to reconstruct camera pose.

There have been a few efforts using photos of landmarks to determine location. The landmark recognition engine developed by Zheng et al. [17] uses clusters of registered photos of landmarks from Internet photo collections to recognize landmarks in other photos. Hile et al. [7] used collections of geotagged images to provide landmark-based navigation. The underlying structure-from-motion algorithm [15] used in the later system is also used in the proposed system.

A number of methods have been developed for location-based image annotation transfers. The LOCAL [12] system uses geographic distance to transfer text between geotagged photos. The proposed system, similar to Photo Tourism [14], links annotations to local feature points so that annotations can be linked not only to geographic regions but to objects in the scene. Utilizing local feature points the system can display annotations on photos from various viewpoints as well as when the annotated object is partially occluded. The proposed system enhances these annotations by allowing both text and images to be arbitrary placed on a photo without the prior selection of a region. With the 3D scene reconstruction from the structure-from-motion algorithm annotations can exist in 3D space similar to work done by Feiner et al. [4].

Augmented reality has had a lot of interest lately in part due to the rise is in high-powered mobile phones with cameras and GPS. Augmented reality has been used for navigation directions by Hile et al. [7] where 3D arrows in the direction of the destination are projected onto photos taken by navigating users. Whereas their system uses a path for navigation and sophisticated arrows the proposed system is simpler in nature, though enhanced arrows and directions can easily be applied.

3 System Overview

The objective of the proposed system is to automatically generate models that can be used to determine the geographic location of photos taken of the same scene, and then use the computed location and local features to display and modify an overlay containing additional information. This section describes the details of the proposed system. In order to reduce complexity and try to avoid feature collision the world is segmented into smaller regions described in Section 3.1. Once a region is selected, a simple method using local feature correspondence and stereo vision algorithms can be used to compute the location of a new photo taken in that region, as described in Section 3.2 which lays a foundation for a more sophisticated method using structure-from-motion as described in Section 3.3. Using the local features from the model photos and new photo, the collaborative user interface referred to as scene graffiti and described in Section 3.4 allows users to view and modify text and images linked to local features. Finally, arrows can be displayed on the new image pointing in the direction of a destination, as described in Section 3.5.

3.1 World Segmentation

A critical issue with local feature-based algorithms is the sheer complexity of modeling the entire world's distinctive features. Even with massive parallelization, reconstructing an entire city from a collection of photos can take days [1]. Another issue is that the number of false positive matches increase as the number of photos increase. To overcome these limitations the world is segmented into regions which are then modeled independently. For this paper, regions are defined as all views of a street from one intersection to the next. The regions cover most of the space traversed by people moving from one place to another and also provide visual overlap between neighboring regions. Regions can also be defined as a particular self-contained place such as a museum or theme park. This type of selection could be useful for such places where destinations such as art displays and rides are named and can be easily searched for.

Each region is independently modeled using one of the two methods described in Sections 3.2 and 3.3. The creation of models and their modification can be an offline or live process. For this paper models are created offline using images from Google Street View which are tagged with GPS locations. However, other sets of geotagged photos which have many different viewpoints of

the same scene could be used as well. Once the models are created they are stored for later queries using new photos.

In order to query the system for the geographic location of a new photo a region must first be selected. Regions can be selected in many different ways. For this paper regions are selected in one of two ways, either via GPS or the user. For arbitrary locations, GPS data is used to select a region. If the GPS location lies inside a region, for example on a street, that region is selected. If the GPS location does not lie inside of a region, for example a backyard when using street-defined regions, the region closest to the location is selected. This increases the chances that a photo taken by the user will have local features that correctly match local features in the region's model photos.

There are several ways a user can select a region, some more intuitive than others. A simple example is that of a custom application provided by a museum for their visitors to use as they navigate from one display to the next. In this case the region is the museum and has been preselected for the user. Since the region is small, geotagged photos covering the entire region can be easily taken and the number of local feature points is reasonable. When the user must directly select the region it could be selected from a list of regions or more intuitively from a map. The model associated with the selected region is then queried with new photos taken within or near the region to compute the geographic location using the method described in Section 3.2 or a geographic location and orientation using the method described in Section 3.3.

3.2 Local Feature Correspondence

The first method for using a geotagged photos to compute a geographic location simply consists of matching local feature descriptors and then determining their relative motion in order to finally compute a geographic location using the GPS coordinates of the top two matching model photos. Each photo in the model is represented by its local feature points and GPS coordinates. The feature descriptors of each pair of photos are then matched and only the points that correspond to points in at least two photos are stored as the model for later matching against new photos. Local features in a new photo are extracted when the photo is queried against the model. Then the top two matching photos are determined by matching the local features of the new photo against those in the model using approximate nearest neighbor search. Using the top two photos, an essential matrix is computed and the GPS location is triangulated. This method can be useful when speed is of high importance, model data is precise, and the camera intrinsics are known, or simply when orientation is not necessary.

The SIFT algorithm [10] is used to detect and create local feature descriptors for this paper because of its wide use and ability to independently detect the same distinctive features from a wide range of views. Also current implementations of SIFT exist[1] that are easy to extend and integrate with more

[1]The SIFT binary used for this paper can be downloaded from http://www.cs.ubc.ca/~lowe/keypoints/

complex systems. However, other local descriptors could be used such as as those compared by Mikolajczyk et al. [11] and SURF [3], a relatively efficient Hessian-based feature extractor that creates descriptors similar to those from SIFT. SIFT feature extraction is also efficient, creating thousands of descriptors for a typical photo in less than a second. SIFT uses a difference-of-Gaussian function to create scale and orientation invariant descriptions of the points and the area around them. This description can then be easily matched across views of different scale, rotation, illumination, and weather.

When a user takes a new photo it is queried against the model and its local features are matched against those of the photos in the model to find two other views of the same scene which are then used to determine the location the photo was taken. The approximate nearest neighbors algorithm by Arya et al. [2] is used to match feature descriptors between pairs of photos. For each pair of photos in the model, a k-d tree is constructed to store the 128-dimension descriptors from one photo. Then the tree is searched for a neighboring descriptor using the match constraint that the Euclidean distances between the two nearest neighbors must have a ratio less than 0.6, as described by Lowe and Snavely et al. [15, 10]. Using the distance ratio constraint instead of a threshold increases the chances of finding matches without a substantial increase in the number of incorrect matches. The two model photos with the most matches are then used to compute a location for the query photo.

Given the pair-wise matched feature descriptors from the query photo and the top two matching model photos, the system can determine the relative motion between the three views to finally compute a location. The relative motion between a pair of photos is encoded in their essential matrix. So the essential matrix is computed for each pair of the three photos.

The essential matrix maps normalized coordinates of a point in one photo to the normalized coordinates of the same point in another photo [5]. Normalized image coordinates are image coordinates with the specific camera details removed. Given the camera intrinsics matrix K, the normalized coordinates are $\hat{x} = K^{-1}x$, where x is the coordinate vector of a point in an image and $\hat{x}$ is the normalized image coordinate vector. The eight-point algorithm is used with RANSAC to compute the essential matrix E using at least eight matching feature points [5]. Then to ensure the internal constraints of the essential matrix are met, that is its tow singular values must be the same and $\det(E) = 0$, the singular value decomposition (SVD) of E is modified by replacing the two singular values with their average to give the closest essential matrix in Frobenius norm [5]. Although photos with known intrinsics where employed for the method in this paper, there are many techniques for estimating the camera parameters, some of which are detailed by Hartley and Zisserman [5]. However, part of the speed and simplicity of this method is based on the assumption that the intrinsics are known and do not have to be computed.

Since the essential matrix is composed of the rotation and translation between the two photos $E = [\mathbf{t}]_\times R$, where $[\mathbf{t}]_\times$ is the matrix representation of the cross-product with t, SVD can be used to compute the motion [5]. The SVD of $E = U\Sigma V^T$ where all three matrices are 3 by 3 and U and V are orthogonal.

Then with the matrix

$$W = \begin{matrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}$$

the rotation between the two cameras $R = UW^{-1}V^T$, and the translation t is the last column of U. However, the solution is not unique because the sign of E cannot be computed directly and thus there are four possible motions: (R, t), $(-R, t)$, $(R, -t)$, and $(-R, -t)$. Fortunately, there is only one solution which projects the same point in both photos into the 3D scene in front of both cameras [5]. So the relative motion between each pair of cameras is determined using one of the matched feature points appearing in both photos.

The relative motion is computed up to an unknown scaling. So the absolute scale between the GPS coordinates of the two model images and simple trigonometry is used to compute the global position. Transforming all three translation vectors into a common coordinate frame forms a triangle. Finally, using the geographic distances of the model photos a GPS location can be triangulated for the query photo.

3.3 Structure-from-Motion Reconstruction

The method described in the previous section computes a GPS location, but it does not compute a global orientation. This requires a more complex algorithm using structure-from-motion, however it provides the additional benefit that the photos can come from cameras with unknown intrinsics. For this method an open source package by Snavely et al. called Bundler[2] is used to reconstruct each region. The same model photos and local SIFT features from the previous section are passed to Bundler which then performs 3D registration of the photos. Since Bundler uses SIFT features to match photos it can register photos from a wide range of overlapping views. The main advantage to using Bundler is its use of sparse bundle adjustment to efficiently refine computed camera poses. Sparse bundle adjustment algorithm by Lourakis and Argyros [9] is implemented in Bundler using a modified version of another open source package called sba[3]. The use of bundle adjustment provides better estimation of camera intrinsics and distortion which allows the system to use photos from nearly any camera.

Bundler first matches the features between each pair of photos in the model and then prunes geometrically inconsistent features. Then an initial pair of photos is selected by first finding a homography between each pair of matching photos using RANSAC. The photo pair with the lowest percent of inliers but with at least 100 matches is selected as the initial pair [15]. This is to ensure that the initial pair has a substantial amount of matches but not a narrow baseline to avoid the case when a pair of photos are identical. Bundler then searches for the unregistered photo with the most matches to registered feature points and then registers the photo by estimating the camera's intrinsic and extrinsic

[2]Bundler can be downloaded from http://phototour.cs.washington.edu/bundler/
[3]The modified sba is distributed with Bundler.

parameters using the direct linear transform technique [5]. Each of the photo's feature points are registered if the point is geometrically consistent and visible in at least one registered photo.

At each iteration the bundle adjustment algorithm is executed to refine each camera pose. The bundle adjustment algorithm is an optimization problem that seeks to simultaneously minimize the reprojection error of the registered feature points and camera poses, thus refining the entire 3D scene with the additional information from each new photo. This is the critical difference between the previous method, providing a refined camera pose necessary for realistic augmentation. The Bundler algorithm continues until no more photos capture at least twenty registered feature points. Further details of the Bundler algorithm are described in the paper by Snavely et al. [15].

SIFT features are detected and matched in the same manner as the previous method when a new photo is taken and queried against the model. The geographic location can also be computed as was done in the previous method using the camera poses from Bundler without computing essential matrices for the query photo. Given the orientation of the first model photo R_1 and the rotation between it and the query photo R_{1q}, the orientation of the query photo $R = R_1 R_{1q}$. Another procedure for determining location using this model could be to register the query photo with Bundler, but the proposed method has been shown to have similar results using less computation time [7].

3.4 Scene Graffiti

Scene graffiti consists of text and images that users can place on photos. The graffiti are then automatically displayed on new photos of the same scene or object by querying the system. Because graffiti objects are linked to local feature points, both proposed methods for determining geographic location allow for scene graffiti. Since it is assumed that multiple users will be using the system, the graffiti is shared when a user photographs an object or scene with graffiti added by another user. The graffiti objects linked to the features in the scene are displayed on the user's photo. So if a museum visitor has a comment about a specific part of a sculpture, they can place and automatically link the comment to the specific part of the sculpture for visitors to see. This enables collaboration between users in situations such as finding a good restaurant, avoiding boring theme park rides, and exploring a new environment.

To add text or an image to the scene the user first photographs the scene. Then the user places the image or string of text on the photo at the position desired. The graffiti object is then linked to all of the feature points under the object. A normalized offset between the top-left pixel of the graffiti object and the top-left feature point covered by the object is stored with the object as well as the normalized width of the region of feature points. The normalized distances are used to scale the graffiti object to cover the same region in future photos. If a graffiti object is not placed over any feature points the two closest feature points in the photo are used. At least two feature points are needed to determine the appropriate scale for the graffiti. It is assumed that the top-left

Figure 1: The smiley face was added to the photo on the left and then displayed on the photo on the right based on matching feature points.

feature point of a linked graffiti object in one photo will also be the top-left feature point in another photo. However, this does not account for mirrors, upside down photos or occluding objects.

When a new photo is taken the set of graffiti objects associated with the matched feature points are displayed. Similar to the annotations in the Photo Tourism system [15], graffiti is displayed if at least one of its linked feature points are visible in the photo. Graffiti is also not displayed if the scale is too large or too small that it covers the photo or can barely be seen. Graffiti that are between five and eighty percent of the photo size are displayed. If the top-left feature is not visible in the photo the graffiti object is aligned to the top-left visible feature and the scale is estimated. Graffiti can than easily be moved or removed with the changes visible in all new photos. This method avoids occlusions and handles multiple viewpoints by linking graffiti objects with a set of local feature points. Since all of the local features stored in the model appear in more than one model photo it can be assumed that they represent distinctive features of objects in the scene, and so linking graffiti to features is in essence linking them to objects in the scene.

3.5 Navigation Overlays

To enhance the navigation experience the proposed system includes overlaying photos with 3D arrows pointing in the direction of the destination. Camera orientation is needed to provide intuitive navigation directions. Only the second method for determining the geographic location of a photo, as described in Section 3.3, provides complete camera pose information. With the resulting camera pose the direction is computed and arrows are augmented on the photo in the direction of the destination. The user can continually take photos to navigate to their destination.

To begin, the user selects a destination from any of the various possible user interfaces such as a list or a map. A list of destinations were provided by the system for this paper. Then the user photographs a representative view of

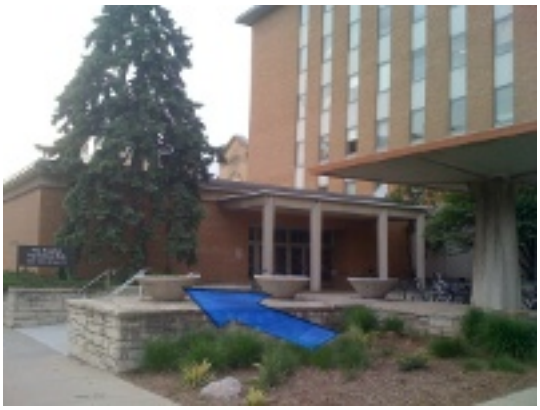

Figure 2: Arrow augmented on the photo indicating the direction of Cousins Subs.

the user's location. The photo's geographic location and camera pose is then estimated using the structure-from-motion method. Then the angle between the current location and the destination is computed, and using that angle along with the camera orientation an arrow is projected onto the photo pointing in the direction of the destination. The camera pose is also used to tilt the arrow to fit the scene in the photo more realistically as was also done by Hile et al. [7]. The user can then navigate to their destination using the augmented photo instead of a map, which alone cannot point a person in the correct direction.

This method does not provide the most intuitive navigation directions because of the naive direction computation. However, input from a path-finding algorithm could inherently be used for more intuitive and useful direction. Since navigation is not the primary focus of this paper the proposed system does not include an algorithm for finding a path from one place to another. Many techniques exist in the literature for finding paths between two places.

4 Results

The proposed system was implemented on a PC, but SIFT and other detectors can be efficiently run on many mobile phones and then the feature descriptors can be passed to a server running the model. Photos from Google Street View, with their GPS coordinates stored, were used to build the models because they are relatively easy to extract versus manually labeling photos and they cover many ranges of views. The views are of the streets of the UW-Madison campus in an urban city with more architecture than trees which contains more distinctive features than a more open landscape. The implemented test did not use GPS to select model regions. Instead the system was given a GPS location to select a region. So even though a GPS device was not used for this paper one could easily be used.

Verification of the implemented system proved difficult. Ground truth for geographic locations of photos is difficult to determine, but the nearly identical structure-from-motion method employed by Hile et al. was shown to be sufficiently accurate for navigation and augmentation purposes [7]. The GPS coordinates of the Street View photos inherently have a level of error due to the GPS system, though presumably quite small given the manner in which they are captured. So there is an unknown level of error in the models themselves. Bundler could have been aligned to the world using a few "best-fit" camera GPS locations [7, 14]. However, using automatic methods to capture higher-resolution photos with more precise GPS coordinates can be used that do not require refinement. Over 100 photos were taken using a mobile phone with a GPS device that automatically tags photos, and those photos were used as ground truth to test the system.

The first method for determining geographic location described in Section 3.2 was not as accurate as the method using structure-from-motion described in Section 3.3. Many times the computed geographic location was over 0.001 degree (about a city block) from the geotag, but many were also within 0.0001 degree (several meters). However, building the simple models required substantially less computation time than the structure-from-motion algorithm, and an entire correspondence model could be stored and queried on a mobile phone. The simple pair-wise matching of photos results in a weak estimation of the essential matrix, which is the basis for the computation of the geographic location. The second method, using structure-from-motion and bundle adjustment, produces more precise estimates of the essential matrix, and thus it results in a more precise geographic location. Every location that could be computed using the second method was within 0.0001 degree of the geotag.

Neither method worked well with photos taken from alleys and pointing away from the street. This is due to the fact that there were not enough matching features in the model photos, and it shows that simply taking photos from streets will not be enough to provide arbitrary outdoor location information. Neither method will work with photos that were taken by pointing the camera too high or too low. There just are not enough features in the sky, and the Google Street View photos are not of a high enough resolution to detect many features on the ground. Another limitation with the proposed system is that SIFT and other similar local feature descriptors cannot be matched beyond approximately 30 degrees.

The photos from Google Street View tended to have very few feature matches, which resulted in many clusters with few sparse points using Bundler. Bundler only registers photos that have enough features that match already registered features. Thus which photos get registered is dependent upon which two photos are chosen to begin the bundle. If the two photos do not have a wide enough baseline to cover other views in the model another cluster will be created. Since the Street View photos had few pair-wise matches several clusters were created for each model by iteratively running Bundler on unregistered photos until no more clusters could be reconstructed.

Scene graffiti transferred to photos well in most cases. The graffiti-feature

Figure 3: The location of the photo on the left was correctly computed to be the location on the map on the right. Most estimations were within meters of the phone's GPS location.

offset prevents displayed graffiti from running off of the left and top sides of photos, but it does not account for going off the right and bottom sides of photos. Placing graffiti on regions far from any feature points can produce odd results. For example, placing graffiti in the sky on one photo may cause the graffiti to be displayed on a particular object in another photo instead of the sky. However, besides these few cases the graffiti appeared where expected.

5 Conclusion

This paper proposed a system for both determining a geographic location from a photo and then augmenting the photo with text, images, and navigation information. The system is automatically created from a collection of geotagged photos using local feature descriptors. A simple and efficient method for determining GPS location which can be run entirely on a mobile device was shown to be effective enough for when GPS is not available. Building upon that method, a more complex method was proposed using structure-from-motion and bundle adjustment to provide a refined photo location and orientation. The models are then extended to include scene graffiti and augmented navigation directions that create an environment for social tourism.

The proposed system forms the basis for a fully automated augmented reality collaboration application. Several enhancements could be made to the system in the future. The GPS coordinates of the model photos could be refined using Bundler to provide more accurate geographic location estimates. The navigation overlays could be enhanced by detecting pavement and walkways to project arrows onto. However, one advantage of the proposed system is that it works where GPS does not, and it can be more precise than global or landmark photo approaches. By taking photos of where they have been, users can keep track of the path they have taken and what they have seen. Then they can visually share that information and experience with others. This paper shows how these

techniques from computer vision can be extended to enhance the collaboration and navigation experience.

References

[1] Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R. (2009). Building rome in a day. In *Proceedings of ICCV*, 2009.

[2] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM 45*(6) 891–923.

[3] Bay, H., Tuytelaars, T., Gool, L. V. (2006). Surf: Speeded up robust features. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 404–417, Springer.

[4] Feiner, S., MacIntyre, B., Hollerer, T., Webster, A. (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *Proceedings of the IEEE international symposium on wearable computers* (pp. 74-81).

[5] Hartley, R. I., Zisserman, A. (2004). *Multiple view geometry.* Cambridge University Press, Cambridge, UK.

[6] Hays, J., Efros, A. A. (2008). im2gps: estimating geographic information from a single image. *Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008.

[7] Hile, H., Grzeszczuk, R., Liu, A., Vedantham, R., Kosecka, J., Borriello, G. (2009). Landmark-based pedestrian navigation with enhanced spatial reasoning. In *Proceedings of Pervasive '09.* Springer-Verlag.

[8] Johansson, B., Cipolla, R. (2002). A system for automatic pose-estimation from a single image in a city scene. In *Proceedings of the IASTED international conference on signal processing, pattern recognition and application.*

[9] Lourakis, M., Argyros, A. (2004). *The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm* (Technical Report 340). Institute of Computer Science-FORTH, Heraklion, Crete, Greece. (Available from http://www.ics.forth.gr/~lourakis/sba).

[10] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60(2),* 91–110.

[11] Mikolajczyk, K., Schmid, C. (2005). A performance evaluation of local descriptors. *PAMI 27* 1615–1630.

[12] Naaman, M., Song, Y. J., Paepcke, A., Garcia-Molina, H. (2004). Automatic organization for digital photographs with geographic coordinates. In *Proceedings of the ACM/IEEE-CS joint conference on digital libraries* (pp. 53–62).

[13] Oliva, A., Torralba, A. (2006). Building the gist of a scene: The role of global image features in recognition. In *Visual Perception, Progress in Brain Research*, Volume 155.

[14] Snavely, N., Seitz, S. M., Szeliski, R. (2006). Photo Tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics, 25*(3), 835–846.

[15] Snavely, N., Seitz, S. M., Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, Volume 80, no. 2, pp.189-210, November 2008.

[16] Szeliski, R. (2005). "Where am I?" *ICCV 2005 Computer Vision Contest.* http://research.microsoft.com/iccv2005/Contest/.

[17] Zheng, Y. T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T. S., Neven, H. (2009). Tour the world: building a web-scale landmark recognition engine. In *Prococeedings of ICCV*, Miami, Florida, U.S.A, June, 2009.