Pattern Recognition 47 (2014) 2880–2893

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Estimating the camera direction of a geotagged image using reference images

Minwoo Park [a,*], Jiebo Luo [b], Robert T. Collins [c], Yanxi Liu [c]

[a] Research and Engineering, ObjectVideo, United States
[b] Department of Computer Science, University of Rochester, United States
[c] Department of Computer Science and Engineering, The Pennsylvania State University, United States

ARTICLE INFO

Article history:
Received 11 July 2011
Received in revised form
29 September 2013
Accepted 3 March 2014
Available online 15 March 2014

Keywords:
GPS
Camera geometry
Geo-tagging
Viewing direction

ABSTRACT

Millions of smart phones and GPS-equipped digital cameras sold each year, as well as photo-sharing websites such as Picasa and Panoramio have enabled personal photos to be associated with geographic information. It has been shown by recent research results that the additional global positioning system (GPS) information helps visual recognition for geotagged photos by providing valuable location context. However, the current GPS data only identifies the camera location, leaving the camera viewing direction uncertain within the possible scope of 360°. To produce more precise photo location information, i.e. the viewing direction for geotagged photos, we utilize both Google Street View and Google Earth satellite images. Our proposed system is two-pronged: (1) visual matching between a user photo and any available street views in the vicinity can determine the viewing direction, and (2) near-orthogonal view matching between a user photo taken on the ground and the overhead satellite view at the user geo-location can compute the viewing direction when only the satellite view is available. Experimental results have shown the effectiveness of the proposed framework.

1. Introduction

With the explosion of photos and videos on the Internet, dealing with the large amount of unorganized visual data has become immensely challenging. To address this problem, one fast-emerging phenomenon in digital photography and community photo sharing is geo-tagging. The presence of geographically relevant metadata with photos and videos has opened up interesting research avenues in the multimedia research community for visual recognition of objects, scenes and events. For example, significant performance improvement in event recognition from photos can be achieved through the fusion of user photos and satellite images obtained using the global positioning system (GPS) information [1,2], while image annotation and image exploration can be enhanced using geotagged photos on the Internet [3,4].

However, the current GPS data only identifies the camera location while the interesting scene in the photo may not be at the specified geo-location; in fact it is often in the distance along an arbitrary viewing direction. Viewing direction data provided by a mobile device with a digital compass is typically unavailable, or otherwise error prone because the digital compass is sensitive to motion and magnetic disturbances. The importance of camera location and viewing direction has been recognized by many portable device manufacturers, such as Apple, Nikon, Nokia,[1] Ricoh, and Samsung, who have introduced (prototype) digital cameras and mobile phones that come with a GPS receiver and a digital compass.

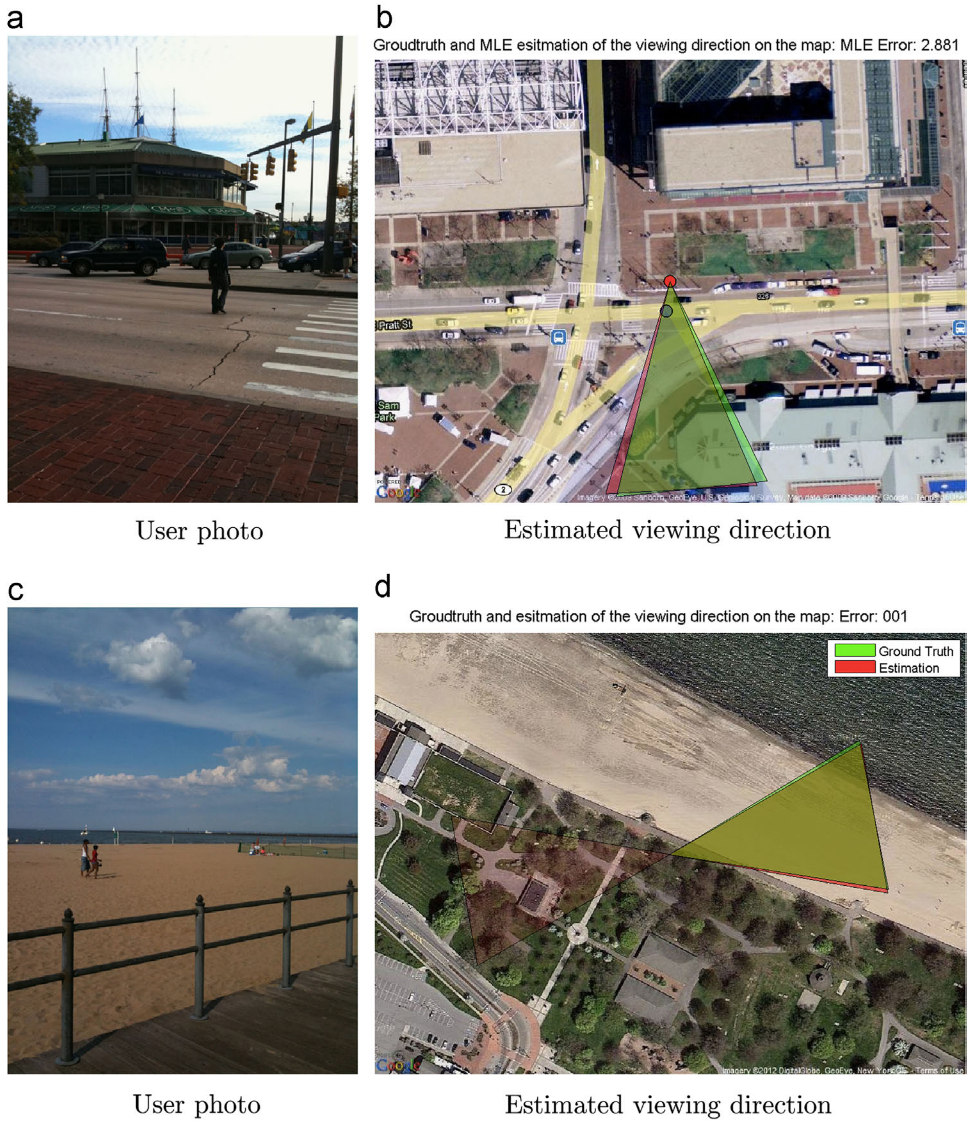
GPS data associated with the photos taken by mobile devices are usually noisy also because GPS signals are weak in the proximity of tall buildings. These difficulties are further recognized by The 2009 and 2010 ACM Multimedia Grand Challenge [5] posed by Nokia where the primary goal is to derive the exact location and direction of a given photo with the aid of reference images. We note that there was no response to this particular challenge at the two past conferences.

In addition, the use of reference images has its own challenges because (1) reference images are not evenly distributed throughout the world, and (2) GPS data associated with the reference images found in the digital photo communities may be inaccurate and inconsistent (due to manual inputs). For example, some photos are associated with the GPS data at the interesting objects, some photos

* Correspondence to: Research and Engineering, ObjectVideo, 11600 Sunrise Valley Drive Suite 210 Reston, VA 20191 USA.
E-mail addresses: mpark@objectvideo.com (M. Park), jluo@cs.rochester.edu (J. Luo), rcollins@cse.psu.edu (R.T. Collins), yanxi@cse.psu.edu (Y. Liu).

[1] http://research.nokia.com/research/projects/mara/index.html.

http://dx.doi.org/10.1016/j.patcog.2014.03.002
0031-3203/© 2014 Elsevier Ltd. All rights reserved.


a
b
Groudtruth and MLE esitmation of the viewing direction on the map: MLE Error: 2.881
c
d
Groudtruth and esitmation of the viewing direction on the map: Error: 001
🟩 Ground Truth
🟥 Estimation

User photo

Estimated viewing direction

User photo

Estimated viewing direction

Fig. 1. The objective is to estimate the camera viewing directions. (a) and (c) Geotagged urban photo. (b) and (d) Geotagged suburban photo (green FOV triangle – ground truth, red FOV triangle – estimate). All figures are best viewed at 200% zoom on screen. Note that the red triangles for the estimated FOV can be covered by the green triangles for the ground truth FOV when the estimates are near perfect. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

are associated with GPS data at the user camera locations, and most tagged geo-locations are noisy. Indeed, estimating both geo-location and viewing direction simultaneously is an extreme problem.

In this study, the deficiency of the current GPS data and the scarcity of reference images are addressed by utilizing Google Street Views (covering major cities) when available and Google Earth satellite views (covering the entire globe) otherwise. Our goals are (1) to estimate the 2D viewing direction given GPS coordinates, and (2) to provide a general framework that can cover the entire world. Fig. 1 illustrates our goals with actual examples (with camera viewing directions estimated by the proposed algorithms) taken in both urban and suburban environments.

2. Related work

Snavely et al. [6,7] developed the Photo Tourism system for browsing large collections of photographs in 3D. Their system takes as input large collections of images from either personal photo collections or photo sharing web sites, and automatically computes each photo's viewpoint and a sparse 3D model of the scene. Their photo explorer interface then enables the viewer to interactively move about the 3D scene by seamlessly transitioning between photographs.

Later, Snavely et al. [8] also proposed a system where the goal is finding paths through the world's photos. When a scene is photographed many times by different people, the viewpoints often cluster along certain paths. These paths are largely specific to the scene being photographed, and traverse interesting regions and viewpoints. This work seeks to discover a range of such paths and turn them into control points for image-based rendering. Their approach again takes as input a large set of community or personal photos, reconstructs camera viewpoints, and automatically computes orbits, panoramas, canonical views, and optimal paths between views. The scene can then be interactively browsed in 3D using these controls or with five degree-of-freedom free-viewpoint control. However, the works introduced so far have not dealt with mapping of data back to actual maps. To address this problem automatically, Kaminsky et al. [9] proposed a method for aligning 3D point clouds with overhead images. They address the problem of automatically aligning structure-from-motion reconstructions to overhead images, such as satellite images, maps and floor plans, generated from an orthographic camera. They compute the optimal alignment using an objective function that matches 3D points to image edges while imposing free space constraints based on the visibility of points in each camera. However, their method is not suitable for estimating the viewing direction of a single arbitrary user photo because the method requires hundreds of images related to the photo.

Lalonde et al. [10,11] analyzed two sources of information available within the visible portion of the sky region: the sun position, and sky appearance. By fitting a model of the predicted sun position to an image sequence, they estimated camera parameters and geo-location including how to extract camera parameters such as the focal length, and the zenith and azimuth angles. Although their solution requires visibility of sun or sky in a user photo and a database, they generated impressive results on such images.

Schindler et al. [12] proposed a method for automatically geo-tagging photographs taken in man-made environments via detection and matching of repeated patterns on building facades. They exploit the highly repetitive nature of urban environments, detecting multiple perspectively distorted periodic 2D patterns in an image and matching them to a 3D database of textured facades by reasoning about the underlying canonical forms of each pattern. Although they show very accurate results on a few image sets, their driving cue for the estimation is repeating patterns and thus the algorithm requires a database for such building facades.

Luo et al. [4] proposed a system called View Focus where they retrieve geo-tagged photos sharing similar viewing directions using community photos. The system depends on bundle adjustment [13] that requires significant overlap in scene content between photos. However, with the exceptions of popular landmark spots for which there is a concentration of community photos, this requirement is often not satisfied in practice.

In contrast, our proposed method [14] is a general framework that can estimate the 2D viewing direction of geotagged photos in more realistic settings. Beyond our previous work [14], we (1) add more reference images to make the estimation of viewing direction more robust, (2) propose robust matching that provides more number of reliable matching, (3) remove approximation and some assumptions made in our previous work [14], and (4) propose an optimization algorithm for estimating viewing direction. Our proposed method only requires one input query image that is geotagged, regardless of the picture-taking environment.

3. The proposed framework

Our proposed method consists of two parts, where PART 1 (Section 4) handles a case when Google Street View is available and PART 2 (Section 5) handles a case when Google Street View is not available (Fig. 2). Geo-location tagged in a user photo U is used to automatically check the availability of the reference images on the Internet. If Google Street View near the geo-location of U is available, we download all of Street View images $S_{all} = \{S_i | 1 \le i \le N\}$ within a certain peripheral boundary (Fig. 3) where N is the total number of Street View images around the geo-location of U and S_i is an ith Street View image in the set S_{all}. The matching Street View image, S_j, that contains the same scene as U is retrieved using RANSAC based homography matching algorithm frequently used by many researchers (Fig. 4a and d). All of the matching Street View images, $S_j \in S_{all}$, are used to estimate the viewing direction of the user photo U.

PART 2 (Section 5) handles a case when Google Street View is not available. In that case, we employ a novel matching algorithm designed for two near orthogonal views, namely, the ground-level view of the camera and the overhead view provided by satellite.

4. PART 1 – when Google Street View exists

When Google Street View exists, the estimation of viewing direction is less ill-posed than otherwise. Advantages of using Google Street View are the following: (1) Google Street View contains *accurate* GPS information, and (2) Google Street View can generate a view in a given viewing direction through its application procedure interface.

4.1. Reference image retrieval

Since Google Street View provides linked nodes where each node indicates its view center and pointers to neighboring nodes, we first

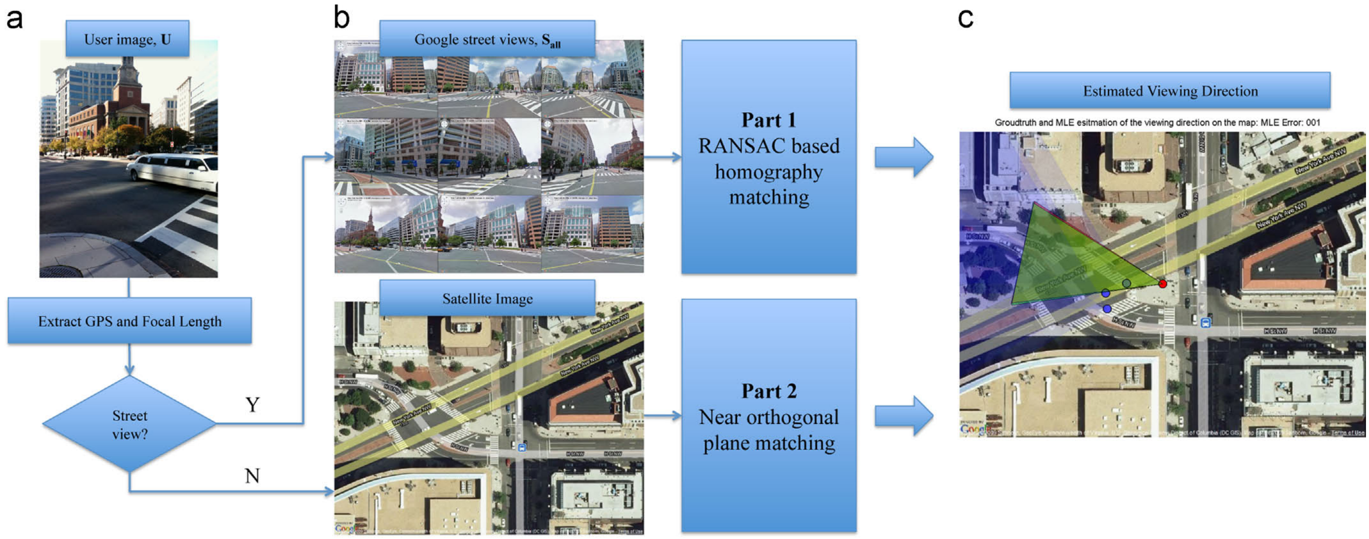

a
User image, U
Extract GPS and Focal Length
Street view?
Y
N
b
Google street views, S_{all}
Satellite Image
Part 1
RANSAC based homography matching
Part 2
Near orthogonal plane matching
c
Estimated Viewing Direction
Groudtruth and MLE esitmation of the viewing direction on the map: MLE Error: 001
New York Ave NW
New York Ave NW

Fig. 2. Overview of the proposed approach: (a) GPS information tagged in a user photo is used to check the availability of reference images from the Internet. (b) If Google Street View is available, surrounding views at that location are downloaded and homography-based matching is performed. If Google Street View is not available, a satellite aerial view at the location is downloaded from Google Earth, and a novel matching between the two near-orthogonal views is performed to estimate the viewing direction. (c) The estimated viewing direction is displayed on the satellite view. Note that satellite views are not used together in PART 1 because they are not helpful in an urban environment (largely showing building roofs and indistinct pavements).

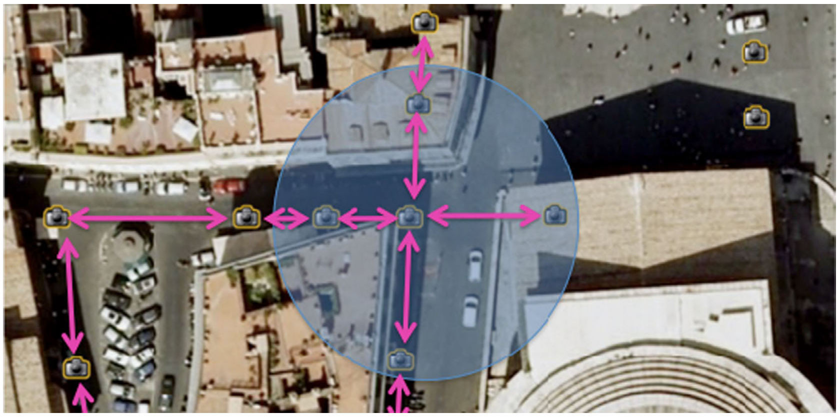


Fig. 3. Google Street View provides a network of linked nodes where each node indicates a view center and there are pointers to its neighboring nodes. We download all Street View images that are within a certain periphery by traversing the linked nodes.

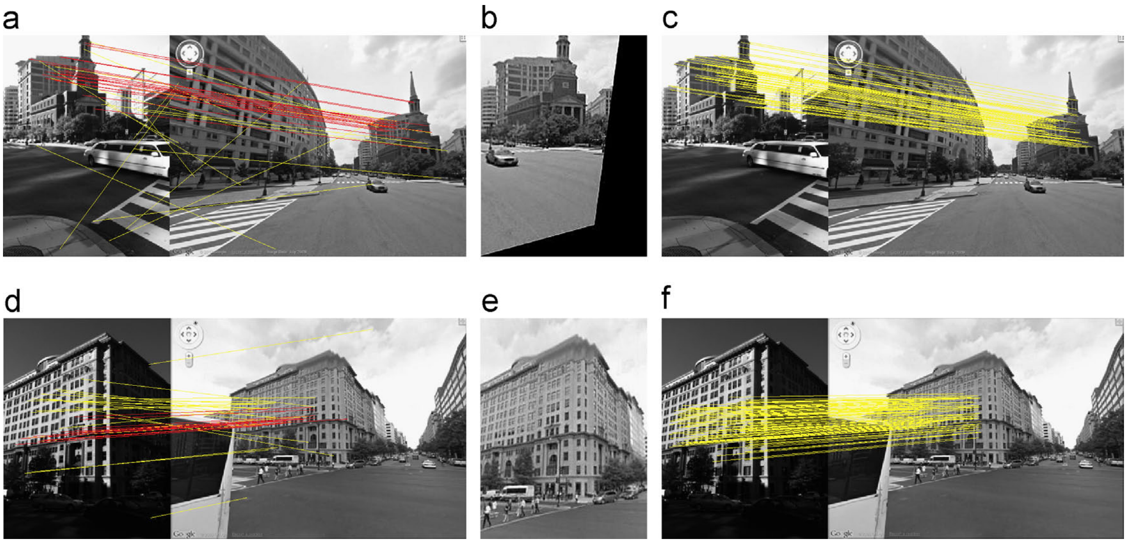


Fig. 4. RANSAC-based matching (a) and (d): the yellow lines indicate initial SIFT matching pairs. The red lines indicate matching pairs after RANSAC-based homography matching. Projective warping (b) and (e): warping from the reference image to the user image. Improved matching (c) and (f): there are more number of reliable matching than (a) and (c). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

identify the Street View node that is closest to the location of a given user photo and download all Street View images that are within a certain peripheral area by traversing the linked nodes (Fig. 3).

To download views at each view center, we generate API calls that simulate viewing angle rotations at every 30° and download the simulated views. Therefore, each view contains information about both the location and viewing direction. These are the references we use to estimate the viewing direction of the user photo.

4.2. Matching

For all of the downloaded reference images, SIFT descriptors are extracted and matched. As shown in Fig. 4, initial SIFT matching can be erroneous (e.g., yellow lines). Since rigid objects such as buildings and traffic signs are everywhere whenever Street Views are available, we can remove false matches using homography constraints between the views. We use RANSAC [15] to compute a homography between candidate views (note that this is a simplified version of the approach by Brown and Lowe [16]). First, we select 4 correspondences randomly from the SIFT matches, $(M_1 : x_1, y_1, u_1, v_1) \sim (M_n : x_n, y_n, u_n, v_n)$ (n is the number of matching points) (yellow lines in Fig. 4a and d), compute the 3×3 projective transform $P_i = [\mathbf{r_1}; \mathbf{r_2}; \mathbf{r_3}]$ where $\mathbf{r_i}$ is a 1×3 row vector, and transform all the remaining points to count the inliers. Inlier set I_S is given by

$$I_S = \{M_i | i = 1, \ldots, n \quad and\ d_i < t\} \tag{1}$$

where the error distance d_i is given by

$$d_i = \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \begin{bmatrix} x_i^* \\ y_i^* \end{bmatrix} \right\|_2, \begin{bmatrix} x_i^* \\ y_i^* \\ 1 \end{bmatrix} = \mathcal{K} \begin{bmatrix} \mathbf{r_1} \\ \mathbf{r_2} \\ \mathbf{r_3} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{2}$$

where t is an empirical threshold to declare inlier points (red lines in Fig. 4a and d) and $\mathcal{K}$ is a normalizing factor that can be computed as

$$\mathcal{K} = 1/(\mathbf{r_3} \times [x_i, y_i, 1]^t) \tag{3}$$

The RANSAC procedure ends when it reaches the maximum number of iterations and proposes the best projective transformation with the most number of inliers. Next, we run least squares using all the inliers to refine the computed mapping and produce projective mapping P_{best}.

In addition, we warp the reference image to a coordinate of the user image U using the P_{best}^{-1} to produce a warped reference image S_{warped}, then we extract a "good features to track" [17] (KLT) in U and track the KLT points from U to S_{warped} to produce more number of matching pairs using simple normalized cross correlation (NCC). To track KLT points, we extract 11 × 11 image patch around each KLT point from U then evaluate the NCC by sliding the extracted image patch on S_{warped}. The matching pair is set by 2D point on the S_{warped} that maximizes NCC score. Finally, aforementioned RANSAC procedure is performed on these tracked points (u'_i, v'_i) again. The final set of matching pair is given as $\{M_i : x_i^{KLT}, y_i^{KLT}, u_i^*, v_i^* | 1 \le k \le n_f\}$ where u_i^* and v_i^* are computed as

$$\begin{bmatrix} u_i^* \\ v_i^* \\ 1 \end{bmatrix} = \mathcal{K}P_{best} \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} \tag{4}$$

and n_f is the final number of matching inliers. Fig. 4 shows the results of the described matching procedure. Finally, we rank the retrieved images with respect to the number of inliers.

4.3. Viewing direction estimation

Once all the relevant images (S_i for $i = 1$–N) are collected, we have N viewing directions associated with S_i and N sets of matching correspondences M between the user photo U and S_i. Given the set of relevant images S_i, the most well-known method for estimating viewing direction is structure from motion. Structure from motion (SfM) refers to the process of finding 3D structure of the common scene seen by multiple images [18]. The process involves in computing camera extrinsic parameters such as camera rotations and relative locations of multiple cameras and viewing direction and camera intrinsic parameters such as principal point and focal length [18]. The standard SfM is discussed below and used as a base algorithm to compare it with our proposed algorithm since the standard SfM can estimate the viewing direction.

4.3.1. Standard structure-from-motion (SfM) algorithms

Given the set of relevant images, we can use standard structure from motion algorithm to estimate the viewing direction of U. However, the stumbling blocks of the standard SfM algorithm in the estimation of a viewing direction using a single user photo and the Google Street views are (1) all of the Street Views are synthesized through panoramic stitching, causing artificial appearance changes and discontinuities at the stitching boundaries, which in turn make structure from motion error prone, (2) even our improved RANSAC-based homography matching algorithm provides only a few point correspondences (i.e., 10–200 matches) due to possible piece-wise warping within a single view, and (3) we have only a few matching images. These violate general requirements of SfM algorithms. To verify this, we use Vincent's SfM toolbox [19]. However, as predicted, any of the standard SfM algorithms in the toolbox could not find correct viewing direction. Therefore, as an alternative method, we propose a new method to overcome erroneous behavior of standard SfM by taking advantages of available information such as locations of the each cameras on Google Map and focal length recorded in the user photo.

4.3.2. Our proposed algorithm

We formulate a problem of estimating viewing direction as a constrained optimization problem where a location of U (C_u), locations of S_i (C_{S_i}), viewing angle of the Street View images (R_s), and the focal length of the user camera (f_u) are known while a focal length of Street View S (f_s), pitch (α), yaw (β), and roll (γ) of U are unknown. We should design an objective function $f(\mathbf{x})$ that is maximized only when the estimated $\mathbf{x} = [\alpha, \beta, \gamma, f_s]^t$ is correct. The problem is formally given as

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x}) \tag{5}$$

Suppose that the good objective function $f(\mathbf{x})$ is given, all of the possible $\mathbf{x}$ can be enumerated and the one that maximizes the $f(\mathbf{x})$ can be selected to estimate the viewing direction. However, enumerating all possible choices of $\mathbf{x}$ on a high-dimensional continuous space is prohibitive. Therefore, inspired by recent efforts on optimization on high-dimensional space, we use smoothing-based optimization (SBO) [20] to estimate the viewing direction. In this section, we will briefly review the SBO, introduce the objective function $f(\mathbf{x})$, then explain our method on estimating the viewing direction using SBO.

Smoothing-based optimization (*SBO*): Consider smoothing a nonnegative function $f(\mathbf{x})$ by convolving with a Gaussian kernel $N(\mathbf{x}; \mathbf{0}, \sigma)$ with zero mean and covariance matrix $\sigma^2\mathbf{I}$, where bold characters denote either an $n \times 1$ vector or $n \times n$ matrix. The value of this smoothed function, evaluated at location $\boldsymbol{\mu}$, is defined as $F(\boldsymbol{\mu}, \sigma) = \int \mathcal{N}(\boldsymbol{u} - \mathbf{x}; 0, \sigma) f(\mathbf{x})\, \mathbf{dx} = \int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma) f(\mathbf{x})\, \mathbf{dx}$ where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma)$ is a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}$. Leordeanu and Hebert [20] define a sequence of mean and standard deviation pairs $(\boldsymbol{\mu}^{(t)}, \sigma^{(t)})$ by the following update equations:

$$\boldsymbol{\mu}^{(t+1)} = \frac{\int_{\mathbf{x}} \mathbf{x} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(t)}, \sigma^{(t)}) f(\mathbf{x})\, \mathbf{dx}}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(t)}, \sigma^{(t)}) f(\mathbf{x})\, \mathbf{dx}} \tag{6}$$

$$\sigma^{(t+1)} = \left[\frac{1}{n} \sum_{i=1}^{n} \frac{\int_{\mathbf{x}} (x_i - \mu_i^{(t)})^2 \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(t)}, \sigma^{(t)}) f(\mathbf{x})\, \mathbf{dx}}{\int_{\mathbf{x}} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}^{(t)}, \sigma^{(t)}) f(\mathbf{x})\, \mathbf{dx}} \right]^{1/2} \tag{7}$$

and prove that the following inequalities hold:

$$F(\boldsymbol{\mu}^{(t+1)}, \sigma^{(t)}) \ge F(\boldsymbol{\mu}^{(t)}, \sigma^{(t)}) \tag{8}$$

$$F(\boldsymbol{\mu}^{(t)}, \sigma^{(t+1)}) \ge F(\boldsymbol{\mu}^{(t)}, \sigma^{(t)}) \tag{9}$$

where μ_i and x_i are the ith entries of vectors $\boldsymbol{\mu}$ and $\mathbf{x}$, respectively.

As shown in [20], scale-space function F has the same global optimum as the original function f, achieved when $\sigma = 0$. Furthermore, F has fewer local optima than f when $\sigma > 0$, due to the smoothing properties of scale-space. Starting with a sufficiently large σ, iteration of Eqs. (6) and (7) performs gradient ascent in scale space, until the procedure converges to a pair $(\boldsymbol{\mu}^*, \sigma^*)$ with σ^* close to zero. The value $\boldsymbol{\mu}^*$ at the final iteration will be, if not the global optimum of f, at least a significant local optimum.

Viewing direction estimation using SBO: Now we use SBO to estimate the best viewing direction. We need to find an unknown variable $\mathbf{x} = [\alpha, \beta, \gamma, f_s]^t$ that maximizes our objective function $f(\mathbf{x})$ where f_s is a focal length of S_i and α, β, and γ are pitch, yaw, and roll angles of U, respectively. The objective function $f(\mathbf{x})$ should be designed in a way that it produces high value on correct estimation of $\mathbf{x}$ and low value on incorrect estimation of $\mathbf{x}$. Fortunately, there are two cues that can measure accuracy of estimation. The first one is Sampson distance of fundamental matrix and the second one is re-projection error of 3D points back to 2D image space.

Sampson distance of fundamental matrix refers to first-order geometric error in the estimation of fundamental matrix F. If the two image views contain the same scene structure, point correspondences between the two views define a fundamental matrix F. The point correspondences M_i define a fundamental matrix F. In addition, knowledge of user camera internal parameter K_u, the user image location C_u, 3D rotation matrix of user camera viewing direction R_u, Street View camera internal parameter K_s, 3D rotation matrix of Street View camera viewing direction R_s, and the Street View image location C_s can define the fundamental matrix F where

K_u, C_u, R_u, K_s, R_s, and C_s are given as

$$K_u = \begin{bmatrix} f_u & 0 & cx_u \\ 0 & f_u & cy_u \\ 0 & 0 & 1 \end{bmatrix}, \quad C_u = \begin{bmatrix} x_u \\ y_u \\ 0 \end{bmatrix}, \tag{10}$$

$$R_u = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \times \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \tag{11}$$

$$K_s = \begin{bmatrix} f_s & 0 & -cx_s \\ 0 & f_s & -cy_s \\ 0 & 0 & 1 \end{bmatrix}, \quad R_s = \begin{bmatrix} \cos\beta_s & 0 & \sin\beta_s \\ 0 & 1 & 0 \\ -\sin\beta_s & 0 & \cos\beta_s \end{bmatrix},$$

$$C_s = \begin{bmatrix} x_s \\ y_s \\ 0 \end{bmatrix}, \tag{12}$$

respectively, where f_u is known focal length of U recorded in the camera EXIF, (cx_u, cy_u) is a known center of the user image U, β_s is a known viewing direction associated to the Street View image S, (cx_s, cy_s) is a known center of the Street View image S, C_u is the known user image location recorded in EXIF, C_s is the known Street View image location given by Google Street View, f_s is a focal length of S, and α, β, and γ are pitch, yaw, and roll angles of U, respectively.

Therefore the correct estimation of $\mathbf{x} = [\alpha, \beta, \gamma, f_s]^t$ should minimize the Sampson distance error and we define the objective function $f(\mathbf{x})$ using Sampson distance as

$$f_{Sampson}(\mathbf{x}) = \left(\sum_i^n \frac{(\mathbf{x}_i^{*}tF\mathbf{x}_i)^2}{(F\mathbf{x}_i)_1^2 + (F\mathbf{x}_i)_2^2 + (F^t\mathbf{x}_i^{*})_1^2 + (F^t\mathbf{x}_i^{*})_2^2} \right)^{-1} \tag{13}$$

where $(F\mathbf{x}_i)_j$ represents the square of the jth entry of the 3×1 vector $F\mathbf{x}_i$, $\mathbf{x}_i^{*} = [u_i^{*}, v_i^{*}, 1]^t$, $\mathbf{x}_i = [x_i^{KLT}, y_i^{KLT}, 1]^t$, and the fundamental matrix F is given as

$$F = K_s^{-t} E K_u^{-1}, \quad E = [-(R_s C_u - R_s C_s)]_\times R_s R_u^t \tag{14}$$

where $[]_\times$ is 3×3 skew-symmetric matrix. If $\mathbf{a} = [a_1, a_2, a_3]^t$ then $[a]_\times$ is given as

$$[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{15}$$

The second cue is the re-projection error of 3D points back to image space of U and S. If the estimation of $\mathbf{x}$ is correct then we should be able to compute 3D points from M_i, R_u, R_s, C_u, C_s, K_u, and K_s then re-projection of the 3D points (X_i, Y_i, Z_i) back to image space U and S should be the same as M_i. Therefore we define the objective function $f(\mathbf{x})$ as

$$f_{Reproj}(\mathbf{x}) = \left(\sum_i^n \left\| \begin{bmatrix} x_i^{proj_{to}U} \\ y_i^{proj_{to}U} \end{bmatrix} - \begin{bmatrix} x_i^{KLT} \\ y_i^{KLT} \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} x_i^{proj_{to}S} \\ y_i^{proj_{to}S} \end{bmatrix} - \begin{bmatrix} u_i^{*} \\ v_i^{*} \end{bmatrix} \right\|_2 \right)^{-1} \tag{16}$$

where $[x_i^{proj_{to}U}, y_i^{proj_{to}U}]^t$ and $[x_i^{proj_{to}S}, y_i^{proj_{to}S}]^t$ are computed as

$$\begin{bmatrix} x_i^{proj_{to}U} \\ y_i^{proj_{to}U} \\ 1 \end{bmatrix} = \mathcal{K} \left(K_u R_u \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - R_u C_u \right) \tag{17}$$

$$\begin{bmatrix} x_i^{proj_{to}S} \\ y_i^{proj_{to}S} \\ 1 \end{bmatrix} = \mathcal{K} \left(K_s R_s \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - R_s C_S \right) \tag{18}$$

where X_i, Y_i, and Z_i are recovered 3D point. The function $f_{Reproj}(\mathbf{x})$ can be evaluated using reprojection error of the 3D point (X_i, Y_i, Z_i) to the image coordinate of U and S for given $\mathbf{x} = [\alpha, \beta, \gamma, f_s]^t$. This is possible since the 3D point (X_i, Y_i, Z_i) can be triangulated when $M_i = \{x_i^{KLT}, y_i^{KLT}, u_i^{*}, v_i^{*} | 1 \le i \le n_f\}$, K_u, K_s, R_s, R_u, C_u, and C_s are given [18].

Now, the unknown variable $\mathbf{x} = [\alpha, \beta, \gamma, f_s]^t$ that maximizes $f(\mathbf{x})$ is estimated using SBO starting from initial guess. How to make good initial guess will be introduced later. Algorithm 4.1 summarizes our proposed method to estimate the viewing direction of U using SBO. We first set a current estimate of $\mathbf{x}$, $\boldsymbol{\mu}^{(t)}$ to initial guess $\mathbf{x}^{(0)}$ and set variance of estimation $\mathbf{x}$, $\boldsymbol{\sigma}^{(t)}$ to $\boldsymbol{\sigma}^{(0)}$ where $t=1$ (line 1 of Algorithm 4.1), 200 samples are drawn from the Gaussian distribution (line 2 of Algorithm 4.1), weight of each sample is evaluated (line 3 of Algorithm 4.1), new weighted mean and variance for each sample and weight are computed using $\mathbf{x}$ and $f(\mathbf{x})$, respectively, given by Eq. (13) or (16) (line 4 of Algorithm 4.1), then new weighted mean and variance are updated (line 5 of Algorithm 4.1), finally this procedure repeats until there is not much change in weighted mean and variance. The converged mean contains the estimated viewing direction of the user image U. In Section 6, we compare accuracy of estimation using $f_{Sampson}(\mathbf{x})$ and $f_{Reproj}(\mathbf{x})$.

Algorithm 4.1. ViewEstimation $(K_u, C_u, C_s, R_s, \mathbf{x}^{(0)}, \boldsymbol{\sigma}^{(0)})$.

```
$t \leftarrow 1$
$\boldsymbol{\mu}^{(t)} \leftarrow \mathbf{x}^{(0)}, \boldsymbol{\sigma}^{(t)} \leftarrow \boldsymbol{\sigma}^{(0)}, \ e_\mu \leftarrow \infty, \ e_\sigma \leftarrow \infty$    (1)
while $e_\mu > 10^{-5} || e_\sigma > 10^{-5}$
do
    comment : Sampling $N$ samples from
              Gaussian distribution
              using mean of $\boldsymbol{\mu}^{(t)}$ and variance of $\boldsymbol{\sigma}^{(t)}$
    $\mathbf{S} \leftarrow Gaussian(\boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}, nS)$    (2)
    comment : Compute weight for each sample $\mathbf{x_i}$
    for each $\mathbf{x_i} \in \mathbf{S}$
        do $w_i \leftarrow f(\mathbf{x_i})$    (3)
    comment : Compute new weighted
              mean and variance
    $\boldsymbol{\mu}' \leftarrow avg(\mathbf{x_i}, w_i), \boldsymbol{\sigma}' \leftarrow var(\mathbf{x_i}, w_i)$    (4)
    comment : Compute movement
    $e_\mu \leftarrow ||\boldsymbol{\mu}' - \boldsymbol{\mu}^{(t)}||_2, e_\sigma \leftarrow ||\boldsymbol{\sigma}' - \boldsymbol{\sigma}^{(t)}||_2$
    $t \leftarrow t+1, \mu^{(t)} = \boldsymbol{\mu}', \sigma^{(t)} = \boldsymbol{\sigma}'$    (5)
return $\boldsymbol{\mu}^{(t)}$
```

Robust initial guess: We also aim to make robust initial guess of $\mathbf{x}$ as follows. To estimate an initial rough viewing direction, we examine each FOV (field of view) at every Street View center. We seek to find overlapping regions seen by all Street Views. Each region is given a relevance weight proportional to the number of inliers found in Section 4.2, as can be seen in Fig. 5a. Then we use Parzen window estimation to find the highest mode of the 2D location of interesting region and obtain an initial estimate of user viewing direction $\beta^{(0)}$ as a ray coming from the center of user location to the highest mode, as can be seen in Fig. 5b. We set $\mathbf{x}^{(0)} = [0, \beta^{(0)}, 0, \frac{40^\circ}{180^\circ}\pi]^t$ and we set $\boldsymbol{\sigma}^{(0)} = [0.005, 0.05, 0.005, 0.01]^t$. We set the variance of the first element and the third element small since there is prior knowledge that pitch and roll angles are likely to be close to 0.

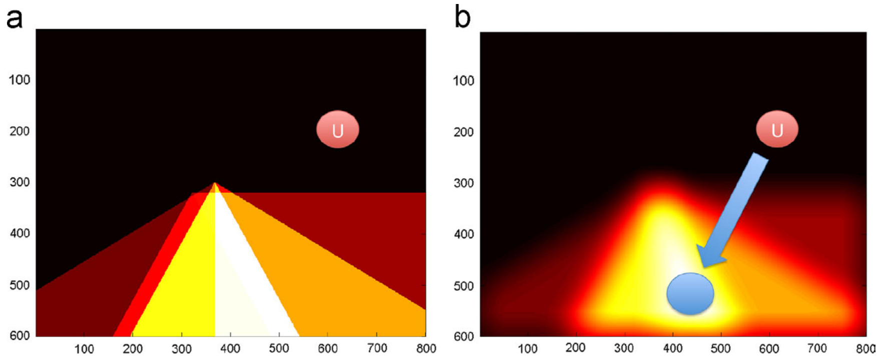


Fig. 5. Initial estimate of yaw angle. The red circles in (a) and (b) indicate the user location and blue circle in (b) indicates 2D location of interesting object. (a) FOV at every Street View center. Each region covered by FOV is given a relevance weight proportional to the number of inliers found in matching. (b) Parzen window estimation of interesting area seen by Street View. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

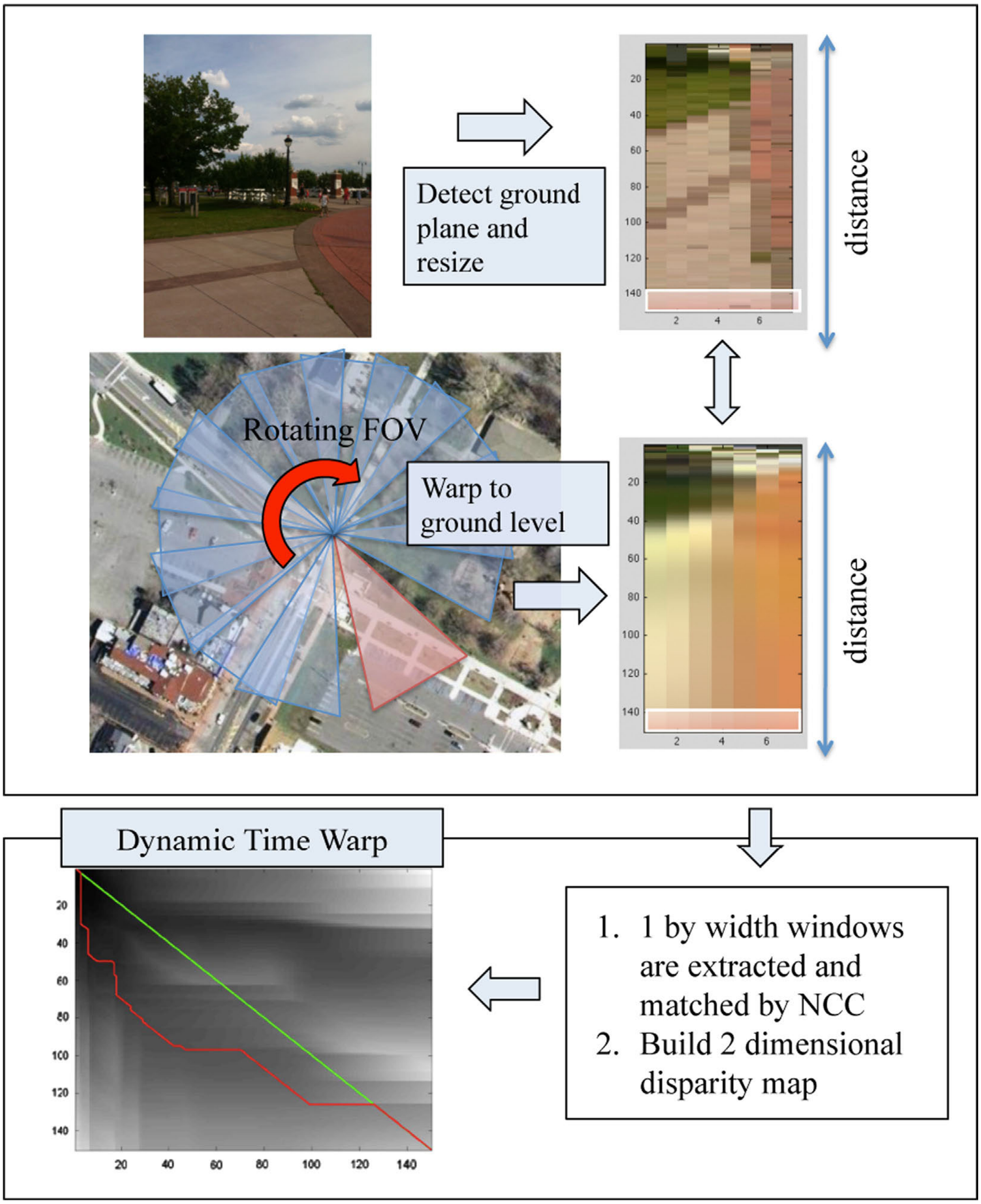

Detect ground plane and resize
distance
Rotating FOV
Warp to ground level
distance
Dynamic Time Warp
1. 1 by width windows are extracted and matched by NCC
2. Build 2 dimensional disparity map

Fig. 6. When only a satellite view exists: (a) user photo, (b) detected ground plane from the user photo using horizon detection, (c) extraction of the ground plane at a specific user photo location, viewing direction, and FOV, (d) simulated ground level view using the result of (c), (e) dynamic time warping and disparity score for (b) and (d).

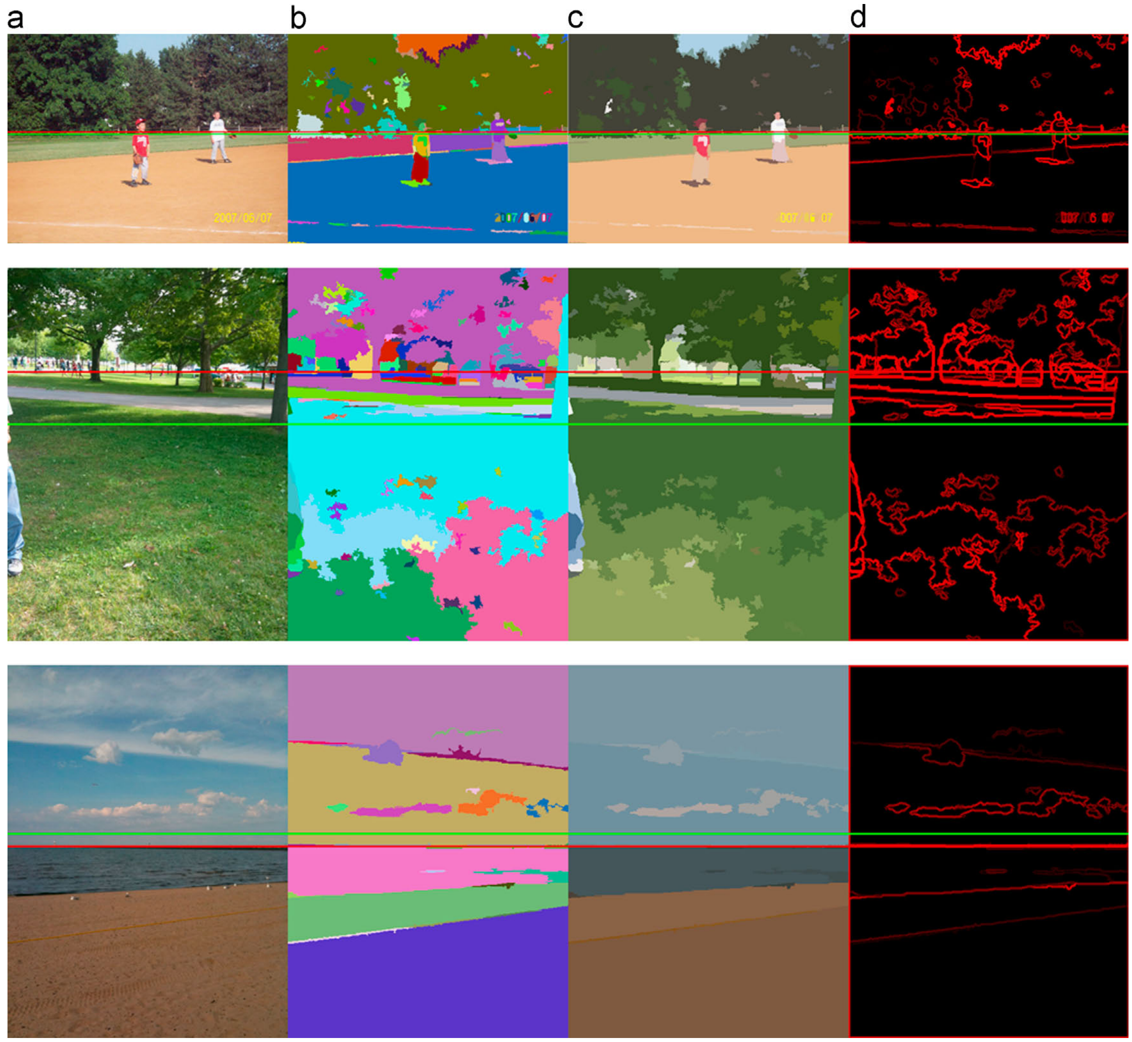


Fig. 7. Horizon detection: (a) input image, (b) segmented image, (c) the segmented image colored with an average color within the segmented region, (d) edge magnitude on (c). Red horizontal line is a maximum likelihood solution and green is a minimum mean squared solution. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

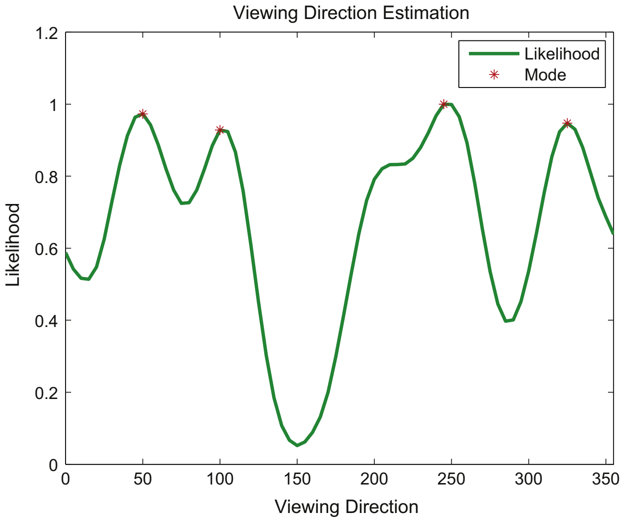


Fig. 8. Likelihood curve for PART 2 DTW: there are multiple modes (indicated by red stars) with comparable likelihood score. The modes are extracted using mean-shift clustering algorithm [22]. We take all the modes of the likelihood function as the estimation of viewing direction. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Table 1
Average and standard deviation of estimation error using PART 1 algorithm.

Dataset	Reprojection	Sampson
DC/Baltimore	13.37° ± 9.81	13.94° ± 9.54
NYC/NJ	13.69° ± 11.97	15.06° ± 11.94
SC, PA	5.05° ± 5.33	10.51° ± 8.13
All 68 images	11.49° ± 10.21	13.40° ± 10.00

5. PART 2 – when only a satellite view exists

When Google Street View is not available for an area, we download a satellite image from Google Earth according to the GPS coordinates extracted from the geotagged user photo. Since the user photo is usually a ground-level view and the satellite view is top-down from above, computing a match between them is extremely challenging because two views are near orthogonal and furthermore the appearance of common objects can vary significantly due to the different imaging conditions. That said, the ground plane and fixture objects on the ground are visible from both the aerial view and ground view (Fig. 6b and d). This is the basis for matching the two near orthogonal views in order to determine the camera viewing direction.

5.1. Alignment of a user photo and a satellite view

The goal of this section is to align the two planes in a way that the effect of alignment error is minimal, provided that there are structures visible from both views (albeit from perpendicular view points). Since we can extract the FOV of the user camera, we can simulate a ground-level view in a certain viewing direction by rotating the FOV on the co-located satellite image, extracting image patch covered by the FOV,

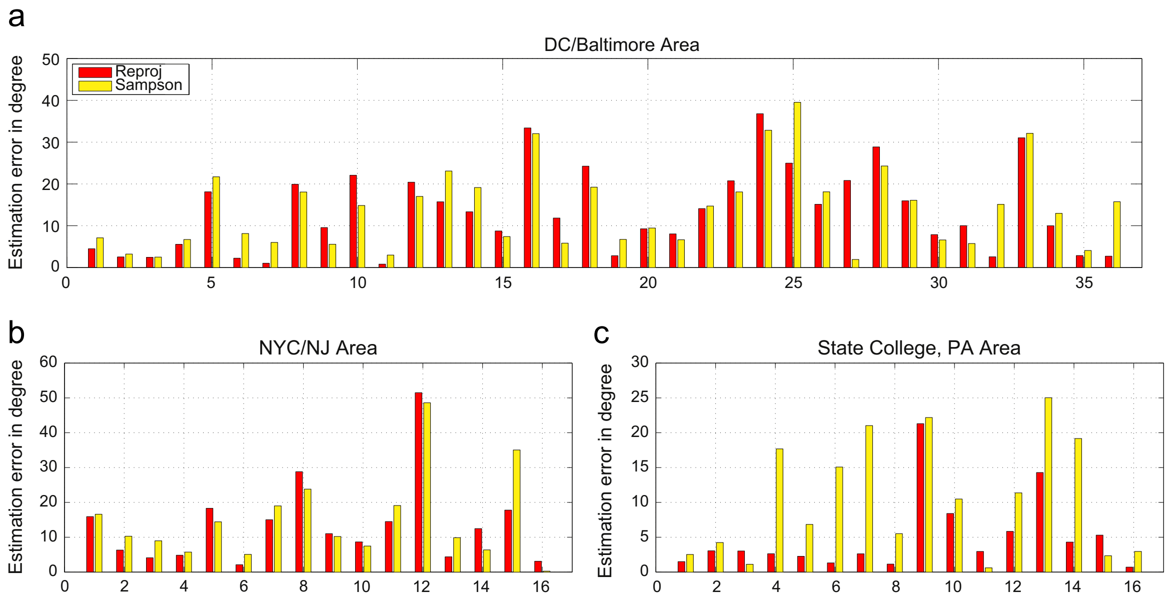


Fig. 9. Estimation error for each user photo using PART 1. Estimation of viewing direction for each user photo in big cities is less accurate than in a small college town due to inaccurate GPS information. The results show that the estimation using $f_{Reproj}(\mathbf{x})$ is more accurate than using $f_{Sampson}(\mathbf{x})$.

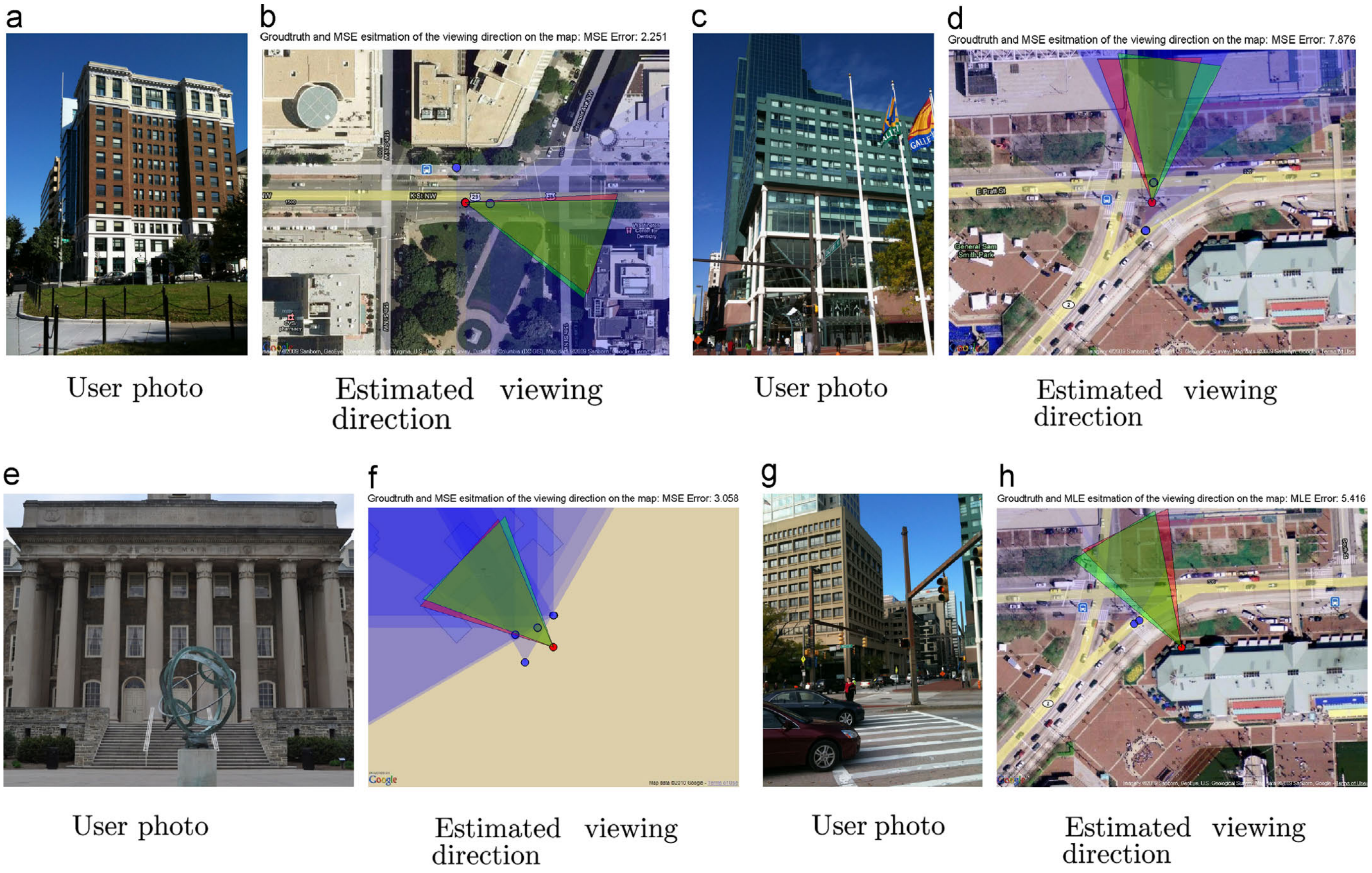

a
User photo
b
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 2.251
Estimated viewing direction
c
User photo
d
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 7.876
Estimated viewing direction
e
User photo
f
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 3.058
Estimated viewing direction
g
User photo
h
Groudtruth and MLE esitmation of the viewing direction on the map: MLE Error: 5.416
Estimated viewing direction

Fig. 10. Example results using Street Views. Left: user photos. Right: estimated viewing directions (red triangles) compared with ground truth (green triangles). Viewing directions are overlaid on normal maps (default 2D tiles of Google Maps) when Google satellite images are not available (Fig. 10f) at the finest zoom level as in other examples (note that Google Street View is available to enable the Part 1 algorithm for all the examples here including Fig. 10f). Note that the red triangles for the estimated FOV can be covered by the green triangles for the ground truth FOV when the estimates are near perfect. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

and warping to the ground-level view (Fig. 6c and d). Then we detect the horizon of the user image and pick only the ground plane region for the matching (Fig. 6b) because the ground plane is most likely to be seen by both the two near orthogonal views.

5.1.1. Horizon detection

We first segment a user photo image using [21]. Then we set RGB value of each segmented region by an mean RGB value computed by averaging over original RGB value of the user photo within each segmented region (see Fig. 7c). Then we compute edge magnitude and sum all the edge magnitude along the x-axis. This yields a vector with length equal to the height of the image. Since we do not expect the image to be perfectly normal to the ground plane, we use a box filter and convolve the computed vector with the filter. For a possible large tilt change, we can increase the size of the box filter so that we can also detect a rotated horizon. Formally, the solution is given as follows:

$$I_r(y) = \sum_{x=1}^{width} I_m(y,x) \tag{19}$$

where $I_m(y,x)$ is the edge magnitude at pixel (x, y) on a segmented image and $I_r(y)$ is an edge response at vertical axis y

$$I_r(y) = I_r(y) \Bigg/ \sum_{y=1}^{height} I_r(y). \tag{20}$$

$$y_{ML} = \arg\max_{y} (I_r * BOX)(y)$$

$$y_{MMSE} = \sum_{y=1}^{height} y \times (I_r * BOX)(y) \tag{21}$$

where BOX is a box filter and $*$ is a convolution operator. Since everything is a linear computation, detection takes less than a second. Some sample results are shown in Fig. 7.

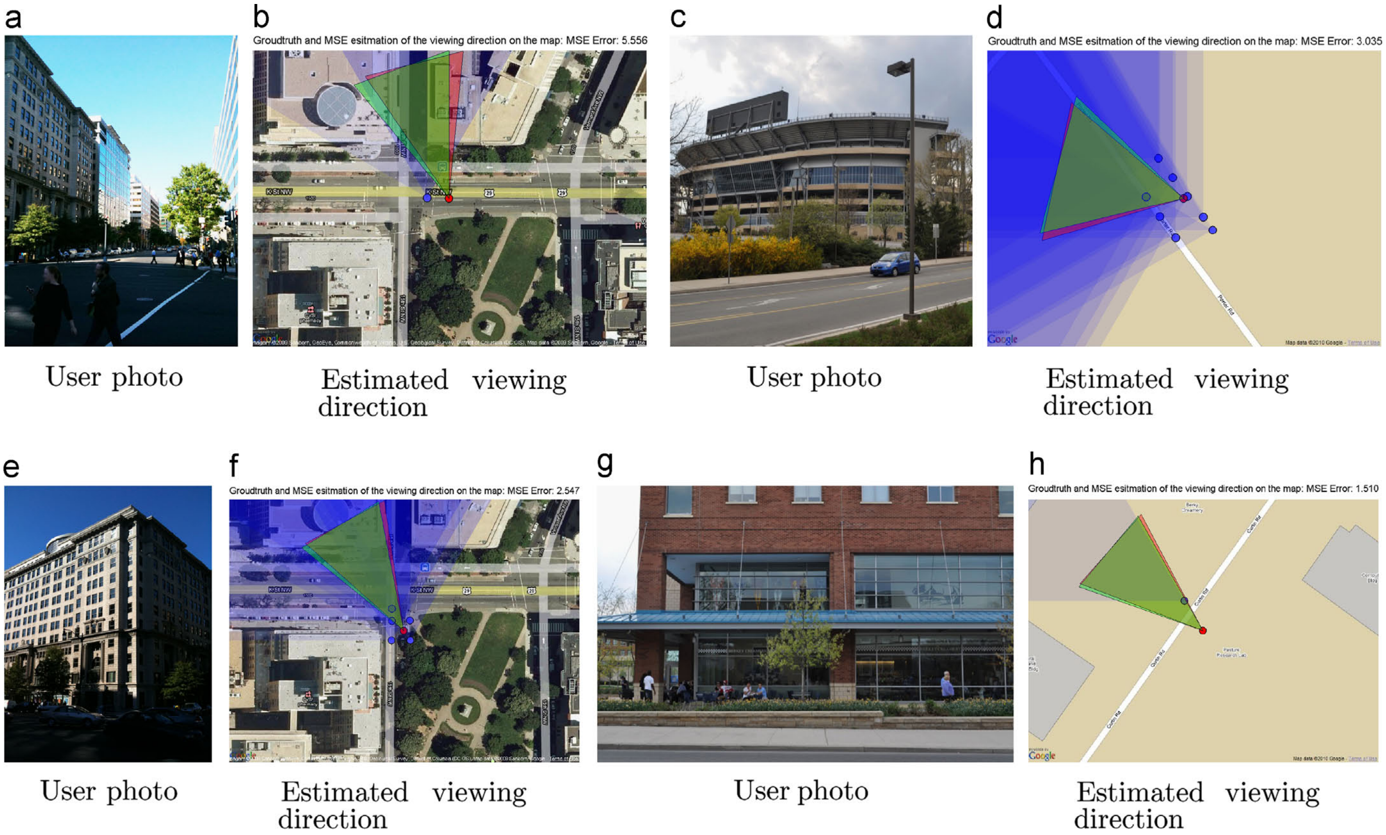
5.1.2. Alignment

We resize both the user image and the satellite image into small patches, which we call $Code_U$ and $Code_S$, to normalize the horizontal axis and vertical axis (Fig. 6b and d). Since we use the same FOV when simulating a ground-level view from the satellite image, this normalization makes the horizontal axes of the $Code_U$ and $Code_S$ approximately correspond to each other. However, the y-axes that relate to distances from a camera center may not correspond to each other because we do not know the tilt angle of the camera (Fig. 6a and b). This will be taken care of in the next section.

5.2. Intensity-based matching through dynamic time warping

If we regard the vertical axis as a time axis, there is a conceptual similarity between our matching problem and time series analysis where two signals have different speed and acceleration (e.g., speech). The similarity score of the two $3 \times w$ matrices (m_i, m_j) extracted from both $Code_U$ and $Code_S$ at distance (i,j) where w is the width of the $Code_U$ and $Code_S$ is used to evaluate similarity between two time series at a given time (i, j) (see Fig. 6b and d).

Having converted our matching problem to time-series analysis, we can use normalized cross correlation (NCC) to generate a 2D disparity map between the codes and use dynamic programming to find the minimum shortest path, as can be seen in Fig. 6. Although we can use any types of appearance similarity scores and features such as the earth-mover's distance [23] and color


a
b
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 5.556
c
d
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 3.035
e
f
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 2.547
g
h
Groudtruth and MSE esitmation of the viewing direction on the map: MSE Error: 1.510

User photo

Estimated viewing direction

User photo

Estimated viewing direction

User photo

Estimated viewing direction

User photo

Estimated viewing direction

Fig. 11. Example results using Street Views. Left: user photos. Right: estimated viewing directions (red triangles) compared with ground truth (green triangles). Viewing directions are overlaid on normal maps when Google satellite images are not available at the finest zoom level as in other examples. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Table 2
Average and standard deviation of estimation error using PART 2 algorithm.

Dataset	NCC	DTW
Ontario Beach (31 images)	33.00° ± 27.85	20.94° ± 15.63

histogram for dynamic time warping (DTW), NCC and texture help overcome the differences in terms of optics, weather, lighting, sun position, shading, shadow variations, and other factors originated from two extremely different imaging conditions (by a camera on a satellite vs. a consumer-level camera on the ground). However, the mentioned variations may be an issue in matching. This is so

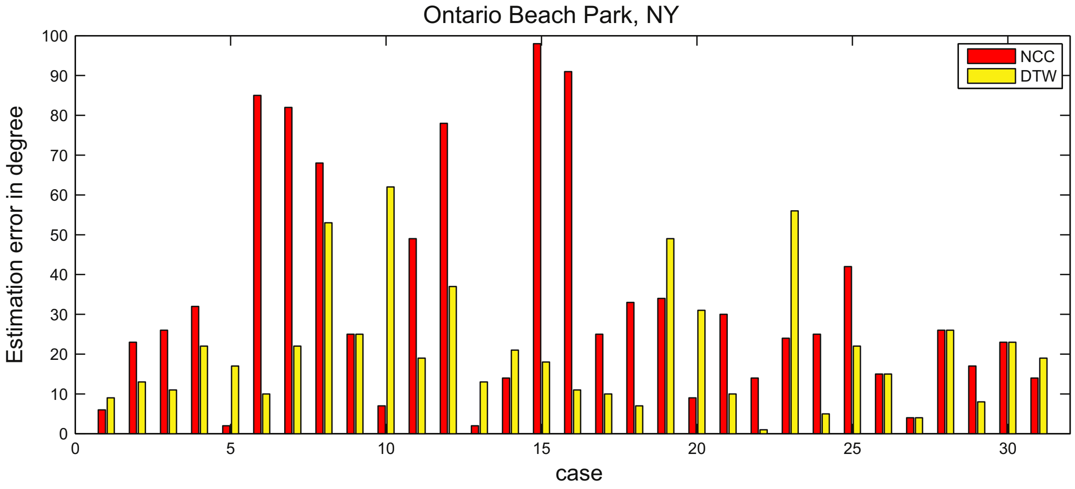


Fig. 12. Error of viewing direction estimation for each user image using PART 2. The results show that the proposed DTW matching is better than NCC.

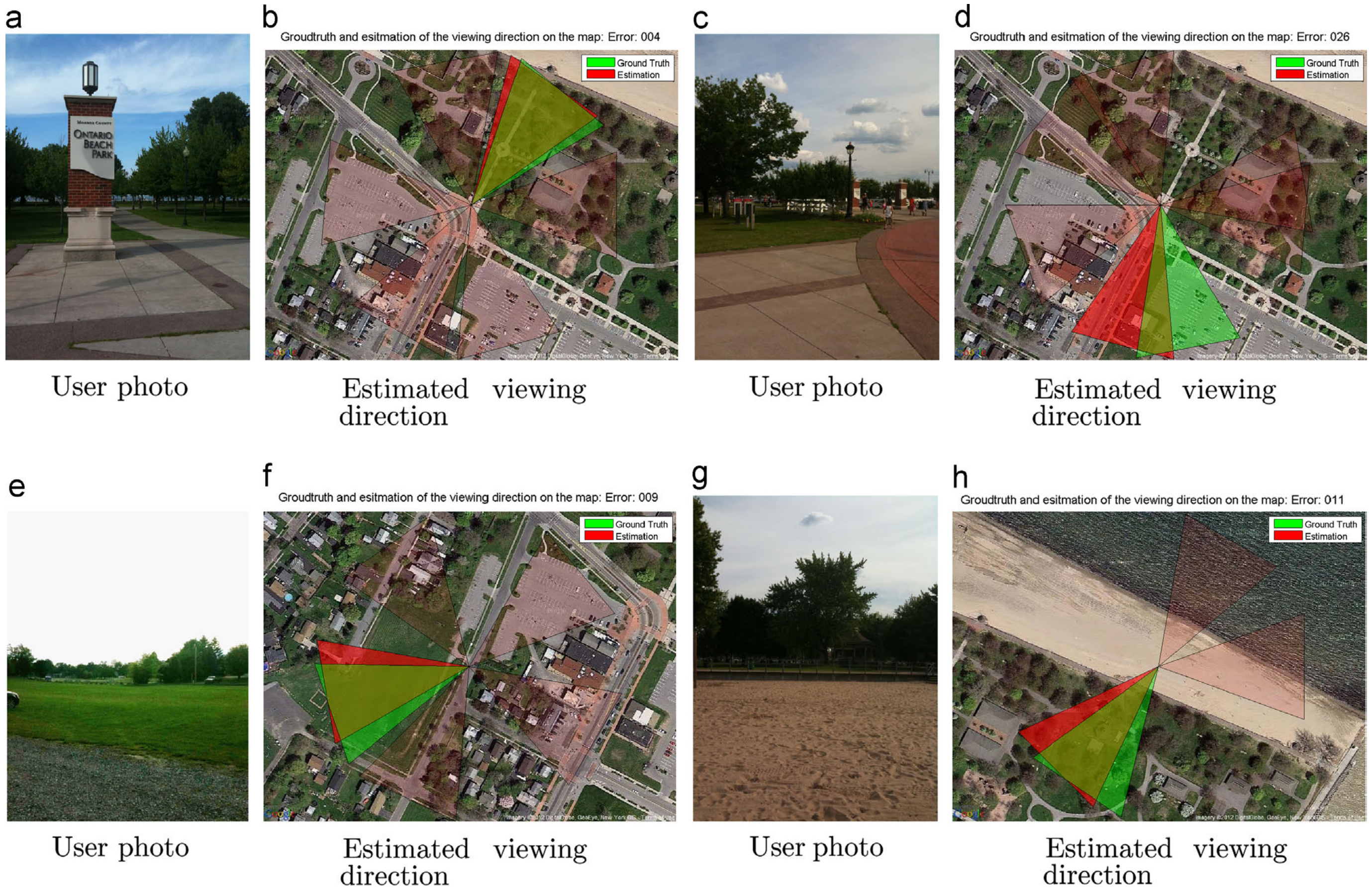

a
b
Groudtruth and esitmation of the viewing direction on the map: Error: 004
Ground Truth
Estimation
c
d
Groudtruth and esitmation of the viewing direction on the map: Error: 026
Ground Truth
Estimation
User photo
Estimated viewing direction
User photo
Estimated viewing direction
e
f
Groudtruth and esitmation of the viewing direction on the map: Error: 009
Ground Truth
Estimation
g
h
Groudtruth and esitmation of the viewing direction on the map: Error: 011
Ground Truth
Estimation
User photo
Estimated viewing direction
User photo
Estimated viewing direction

Fig. 13. Example results using satellite views. Left: user photos. Right: estimated viewing directions (red triangles) compared with the ground truth (green triangles). The transparent red triangles are candidate viewing direction by computing modes L_i. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

because there is only one single reference image for a given user photo and if the user photo is dominated by such variations there is no way to reliably match the user photo to the single reference image. Therefore, instead of estimating one single viewing direction as in PART 1, PART 2 aims to provide a few viewing direction candidates. Fig. 8 shows likelihood L_i with respect to viewing direction i where the likelihood L_i is given as

$$L_i = \left(-DTW_i + \max_{i=1}^{n}(-DTW_i) \right) \Big/ \sum_{i=1}^{n} (-DTW_i) \tag{22}$$

where DTW_i is the minimum cost of dynamic time warping for a given viewing direction i. As can be seen in Fig. 8, there are multiple competing modes (red stars) in the likelihood. Since PART 2 is ill-posed, we choose all of the locations of modes on L_i to be candidates viewing directions and measure the minimum error between the groundtruth and the estimations. In Section 6, we also compare results by the DTW and simple normalized cross correlation to show that DTW has better performance than simple direct NCC matching of $Code_U$ and $Code_S$.

6. Experimental results

Ground-truth data set: Obtaining accurate ground truth is required to measure performance of the method. However, the best way for accurate ground truth generation is to use traditional surveying methods that require intensive labors and expertises [5]. Instead, we have used iPhone 3GS® to collect ground truth data since a manual verification of GPS and viewing direction on the spot is possible using Google Map application right on the iPhone 3GS®. In an attempt to increase the accuracy as well as the number of ground truth images, we have tried Nikon D5000® with Solmeta Geotagger Pro®. However, the state-of-the-art GPS module with 3D compass also requires a manual verification for viewing direction, which prevented efficient data gathering.

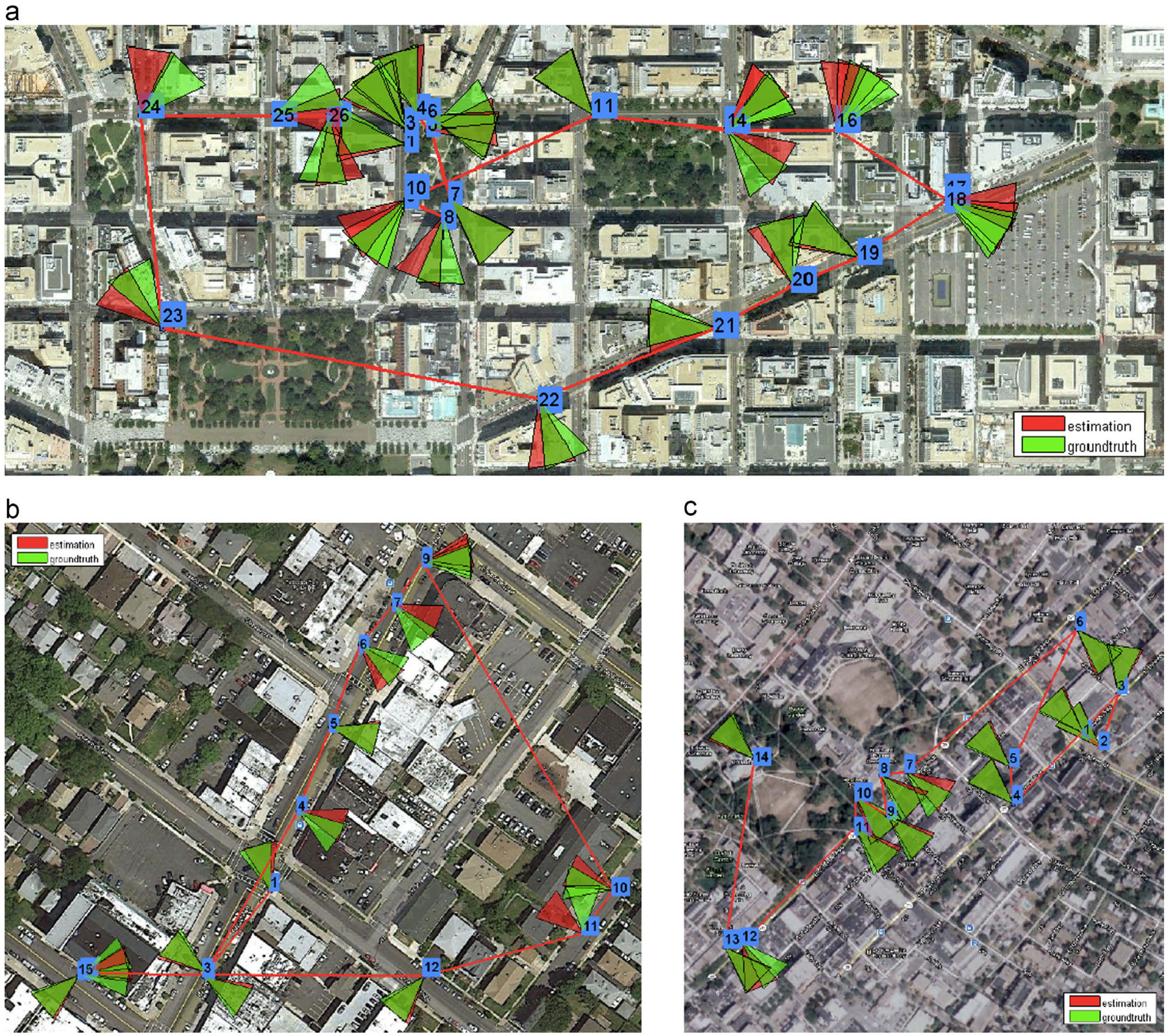


Fig. 14. PART1 – the green triangles show the ground truth viewing directions and the red triangles show estimated viewing directions. The number at the center of the camera shows the sequence of moving trajectory of a person collecting the data. Note that the examples are in cities where street view is available. (a) D.C Downtown, (b) NJ and (c) State College, PA. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

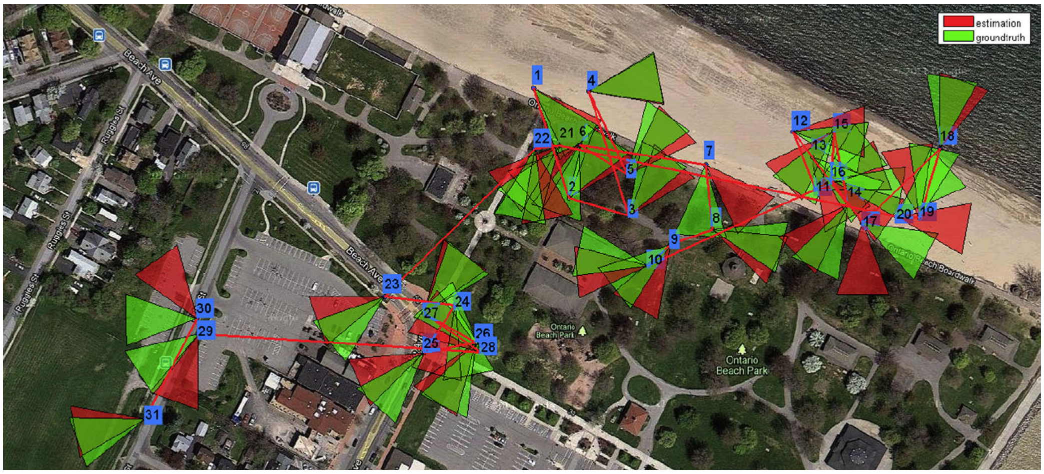


Fig. 15. PART 2 – the green triangles show the ground truth viewing directions and the red triangles show the estimated viewing directions. The number at the center of the camera shows the sequence of moving trajectory of a person collecting the data. Note that the example is in a beach park where no street view is available. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

As a result, we built a dataset of 99 images using iPhone 3GS® and Nikon D5000 with ground truth viewing directions in Washington D.C. (36 images using iPhone 3GS®), New York City (16 images using iPhone 3GS®), Ontario Beach, NY (31 images using iPhone 3GS®), and State College, PA (16 images using D5000) areas for our experiments (we will make the dataset public). The dataset is small given the effort needed to obtain and verify the ground truth, but it is larger than the one used in [12]). More importantly, it covers significantly more diverse cities of various sizes (two major metropolitan cities, a mid-size city, and a college town).

PART 1 *results*: As can be seen in Table 1, our experiments show an average error of $11.49^\circ \pm 10.21$ using $f_{Reproj}(\mathbf{x})$ and an average error of $13.40^\circ \pm 10$ using $f_{Sampson}(\mathbf{x})$. The speed of optimization takes about 1 s and 8 s using $f_{Sampson}(\mathbf{x})$ and $f_{Reproj}(\mathbf{x})$, respectively. The estimation error of viewing direction of each user photo can be seen in Fig. 9. Figs. 10 and 11 show examples in the urban environments using the PART 1 algorithm. Note that our algorithms can handle cases with foreground objects (Figs. 1a and 10e and g) as long as they do not overwhelm the scene. Finally, Fig. 14 shows the trajectories of a person collecting data in various cities and the corresponding viewing direction estimates.

PART 2 *results*: As can be seen in Table 2, our experiments show an average error of $20.94^\circ \pm 15.36$ using DTW and an average error of $33.00^\circ \pm 27.85$ using NCC. The estimation error of viewing direction for each user photo can be seen in Fig. 12. Fig. 13 shows examples in the suburban or park environments using the PART 2 algorithm. Note that due to matching ambiguity originated from lack of information, several viewing direction candidates are identified by PART 2 algorithm instead of suggesting only one estimate as PART 1. Other candidate viewing direction estimates are shown in light red color in Fig. 13. The number of modes in L_i is 2–6 and the range of expected error of randomly chosen viewing direction is 60–26° when the number of drawing is 2–6. Finally, Fig. 15 shows the trajectories of a person collecting data in Ontario Beach Park and the corresponding viewing direction estimates.

7. Discussion

We discuss some of the possible and actual failure modes of the proposed method and suggest future direction of the work. Generally, sun position, shading, shadow, and capture date difference can affect the matching of individual pairs. However, sift feature and RANSAC-based matching approach used in PART 1 tolerate some of the mentioned variations. Moreover, since there are many relevant Street View images for a given user photo, it is less likely that PART 1 cannot find any of proper matching Street View images.

In addition, we found that the detection of horizon in PART 2 is quite robust against such variations as well. E.g., although middle row in Fig. 7 contains several shadows around horizon and grass area, it did not affect detection of horizon.

However, the variations may be an issue in matching method proposed in PART 2. This is so because there is only one single reference image for a given user photo and if the user photo is dominated by such variations, there is no way to reliably match the user photo to the single reference image. The problem becomes even more challenging if the ground plane is not visible in the user photo, or the structures on the ground plane are either distinctive or confusing since it is not possible to estimate the viewing direction with the use of satellite images. As discussed in Section 5.2, the PART 2 problem is in general far more ill-posed than PART 1 and perhaps multiple co-located web photos can be helpful. We plan to use multiple co-located web photos to improve the performance of the PART 2 algorithm. Finally, we notice that the GPS device used for collecting the ground truth was inaccurate at the center of big cities such as D.C. and New York City (in the middle of the concrete jungle with maximum signal interference). This suggests a future research direction where we want to estimate both the viewing direction and (more accurate) GPS coordinates. We will pursue further in these directions to address these problems.

8. Conclusions

We propose a general framework to estimate the camera viewing direction of a single geotagged photo in any environment and have demonstrated its promises. The main contributions are the exploitation of Google Street View and Google Earth satellite images as references, and the solutions designed to overcome various technical challenges inherent within each ill-posed scenario. Our methods perform the best when the recorded GPS coordinates are accurate. In the future work, we hope to evaluate the proposed algorithms on a larger scale and further diversified dataset and refine potentially noisy GPS coordinates while estimating the associated viewing directions within the same framework. We also hope to exploit web photos to help resolve the cases where no street view is available.

Conflict of interest

None declared.

References

[1] J. Luo, J. Yu, D. Joshi, W. Hao, Event recognition: viewing the world with a third eye, in: ACM Multimedia, 2008, pp. 1071–1080.

[2] J. Luo, W. Hao, D. McIntyre, D. Joshi, J. Yu, Recognizing picture-taking environment from satellite images: a feasibility study, in: International Conference on Pattern Recognition, 2008, pp. 1–4.

[3] L. Cao, J. Yu, J. Luo, T.S. Huang, Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression, in: ACM Multimedia, 2009, pp. 125–134, ISBN 978-1-60558-608-3.

[4] Z. Luo, H. Li, J. Tang, R. Hong, T.-S. Chua, ViewFocus: explore places of interests on Google maps using photos with view direction filtering, in: ACM Multimedia, 2009, pp. 963–964, ISBN 978-1-60558-608-3.

[5] Nokia challenge: where was this photo taken, and how?, 2009/2010, ⟨http://comminfo.rutgers.edu/conferences/mmchallenge/2010/02/10/nokia-challenge/⟩.

[6] N. Snavely, S.M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3D, ACM Trans. Graph. 25 (3) (2006) 835–846.

[7] N. Snavely, R. Garg, S.M. Seitz, R. Szeliski, Modeling the world from internet photo collections, Int. J. Comput. Vis. 80 (2) (2008) 189–210, ISSN 0920-5691.

[8] N. Snavely, R. Garg, S.M. Seitz, R. Szeliski, Finding paths through the world's photos, ACM Trans. Graph. 27 (3) (2008) 15:1–15:11, ISSN 0730-0301.

[9] R.S. Kaminsky, N. Snavely, S.M. Seitz, R. Szeliski, Alignment of 3D point clouds to overhead images, in: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009, pp. 63–70.

[10] J.-F. Lalonde, S.G. Narasimhan, A.A. Efros, What does the sky tell us about the camera?, in: European Conference on Computer Vision, 2008, pp. 354–367, ISBN 978-3-540-88692-1.

[11] J.-F. Lalonde, S.G. Narasimhan, A.A. Efros, What do the sun and the sky tell us about the camera? Int. J. Comput. Vis. 88 (1) (2010) 24–51, ISSN 0920-5691.

[12] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, F. Dellaert, Detecting and matching repeated patterns for automatic geo-tagging in urban environments, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[13] M.A. Lourakis, A. Argyros, SBA: a software package for generic sparse bundle adjustment, ACM Trans. Math. Softw. 36 (1) (2009) 1–30.

[14] M. Park, J. Luo, R. Collins, Y. Liu, Beyond GPS: determining the camera viewing direction of a geotagged image, in: ACM Multimedia, International Conference, 2010.

[15] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 06/1981, 24, pp. 381–395. 〈http://dx.doi.org/10.1145/358669.358692〉.

[16] M. Brown, D. Lowe, Automatic panoramic image stitching using invariant features, Int. J. Comput. Vis. 74 (1) (2007) 59–73, ISSN 0920-5691.

[17] J. Shi, C. Tomasi, Good Features to Track, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.

[18] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, New York, NY, USA, 2003.

[19] V. Rabaud, Vincent's Structure from Motion Toolbox, 2006, ⟨http://vision.ucsd.edu/~vrabaud/toolbox/⟩.

[20] M. Leordeanu, M. Hebert, Smoothing-based optimization, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), 2008, pp. 1–8.

[21] P. Felzenszwalb, D. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181.

[22] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

[23] Y. Rubner, C. Tomasi, L.J. Guibas, A metric for distributions with applications to image databases, in: International Conference on Computer Vision, 1998, pp. 59–66.

Minwoo Park is a Research Scientist with ObjectVideo, Reston, VA. He received the B.Eng. degree in electrical engineering from Korea University, Seoul, in 2004, the M.Sc. degree in electrical engineering from The Pennsylvania State University in 2007, and the Ph.D. degree in computer science and engineering from The Pennsylvania State University in 2010. Prior to joining to ObjectVideo, he was a Research Scientist with the Kodak Research Laboratories. His research area is computer vision, with current emphasis on understanding the theory and application of a probabilistic graphical model on computer vision problems. His particular interests are in automatic understanding of 3D from an image, perceptual grouping, event recognition, and an efficient inference algorithm. Dr. Park has authored over 20 technical papers and 9 pending/issued US patents. Dr. Park has been actively involved in numerous technical conferences, including serving as the general chair of IEEE WNYIPW 2011 and Industry Program organizer of IEEE ISM 2012. He routinely serves as a reviewer for IEEE, ACM, Springer, and Elsevier conferences and journals in the area of computer vision. He is a member of the IEEE and the IEEE Signal Processing Society.

Jiebo Luo joined the University of Rochester in Fall 2011 after over fifteen years at Kodak Research Laboratories, where he was a Senior Principal Scientist leading research and advanced development. He has been involved in numerous technical conferences, including serving as the program co-chair of ACM Multimedia 2010 and IEEE CVPR 2012. He is the Editor-in-Chief of the Journal of Multimedia, and has served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, Pattern Recognition, Machine Vision and Applications, and Journal of Electronic Imaging. He has authored over 200 technical papers and 70 US patents. He is a Fellow of the SPIE, IEEE, and IAPR, IEEE Advancing Technology for Humanity. His research spans image processing, computer vision, machine learning, data mining, medical imaging, and ubiquitous computing. He has been an advocate for contextual inference in semantic understanding of visual data, and continues to push the frontiers in this area by incorporating geo-location context and social context. A recent research thrust focuses on exploiting social media for machine learning, data mining, and human–computer interaction, for example, mining the wisdom of crowds for social, political, and economic prediction and forecasting. He has published extensively in these fields with over 200 papers and 70 US patents.

Robert T. Collins received the Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1993 for work on recovering scene models using stochastic projective geometry. He is an associate professor in the Computer Science and Engineering Department at The Pennsylvania State University. His research interests include video scene understanding, automated surveillance, human activity modeling, and real-time tracking. He was a coeditor of the August 2000 special issue of the IEEE Transactions on Pattern Analysis and Machine Intelligence on the topic of video surveillance. He has served as an area chair for CVPR '99, CVPR '09, and ICCV '09 and is currently an associate editor for the International Journal of Computer Vision. He routinely serves as a reviewer for the major conferences and journals in computer vision, IEEE workshops on tracking, video surveillance, and activity recognition, and US National Science Foundation (NSF) review panels in the area of computer vision. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Yanxi Liu received the B.S. degree in physics/electrical engineering from Beijing, China, the Ph.D. degree in computer science for group theory applications in robotics from the University of Massachusetts, Amherst, and postdoctoral training at LIFIA/IMAG, Grenoble, France. She also spent one year at the US National Science Foundation (NSF) Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) with an NSF Research Education Fellowship Award, and a sabbatical-semester in the radiology department of UPMC. She serves as the co-director of the Lab for Perception, Action, and Cognition (LPAC) and is a tenured faculty member of the Computer Science Engineering and Electrical Engineering Departments of The Pennsylvania State University. Before joining PSU, she was an associate research professor in the Robotics Institute of Carnegie Mellon University. Her research interests span a wide range of applications, including computer vision, computer graphics, robotics, human perception, and computer-aided diagnosis in medicine, with a theme on computational symmetry/regularity and discriminative subspace learning from large, multimedia datasets. She co-chaired the International Workshop on “Computer Vision in Biomedical Image Applications” (ICCV 2005), and is chairing the first US NSF funded international competition on "Symmetry Detection from Real World Images" (CVPR 2011). She is a senior member of the IEEE and the IEEE Computer Society.