Automatic Geographic Metadata Correction for Sensor-Rich Video Sequences

Yifang Yin
Interactive and Digital Media Institute, National University of Singapore, Singapore
idmyiny@nus.edu.sg

Guanfeng Wang
Grab Research and Development, Singapore
guanfeng.wang@grab.com

Roger Zimmermann
School of Computing, National University of Singapore, Singapore
rogerz@comp.nus.edu.sg

ABSTRACT

Videos recorded with current mobile devices are increasingly geotagged at fine granularity and used in various location-based applications and services. However, raw sensor data collected is often noisy, resulting in subsequent inaccurate geospatial analysis. In this study, we focus on the challenging correction of compass readings and present an automatic approach to reduce these metadata errors. Given the small geo-distance between consecutive video frames, image-based localization does not work due to the high ambiguity in the depth reconstruction of the scene. As an alternative, we collect geographic context from OpenStreetMap and estimate the absolute viewing direction by comparing the image scene to world projections obtained with different external camera parameters. To design a comprehensive model, we further incorporate smooth approximation and feature-based rotation estimation when formulating the error terms. Experimental results show that our proposed pyramid-based method outperforms its competitors and reduces orientation errors by an average of 58.8%. Hence, for downstream applications, improved results can be obtained with these more accurate geo-metadata. To illustrate, we present the performance gain in landmark retrieval and tag suggestion by utilizing the accuracy-enhanced geo-metadata.

CCS Concepts

•Networks → Sensor networks; •Computing methodologies → Scene understanding;

Keywords

Geo-metadata correction, geo-referenced video, feature matching, 3D projection

1. INTRODUCTION

Online video content is continuing to experience rapid growth. Uploading, sharing, and viewing videos on the web have become an everyday activity in people's lives. With the ubiquity of sensor-equipped smartphones and tablets, it is increasingly common for users to take images or record videos together with the geographic properties of the camera (*e.g.*, location and viewing direction). The presence of the geospatial contextual information has opened up new opportunities in video management systems. This is especially the case with fine-grained contextual information where every video frame is tagged. A great number of applications, such as navigation systems [32], travel recommendation [15], and video tagging [23], can benefit from the geo-metadata by utilizing it as an alternative or supplement to the traditional content analysis approaches. However, the use of geographic information is sometimes hampered by the presence of inaccuracies in the raw sensor data. While for GPS this issue has been extensively studied [3, 4], only a few efforts have been made on the correction of orientation data acquired from digital compasses and accelerometers [18, 20]. The difficulty level of this problem is especially high since (a) unlike GPS, a compass sensor does not provide any accuracy bounds, and (b) from our empirical observations compass errors can sometimes be very high (up to 180°). Although the Structure from Motion (SfM) technique can be applied for camera pose determination, robust estimation results usually rely on the significant overlap and the large baseline (geo-distance between camera locations) among the images to perform 3D reconstruction [12, 22, 29]. Moreover, such methods do not make full use of the geographic priors in the metadata while reconstructing the scenes and therefore result in high computational costs. In this study we argue that, with the rapid growth of spatial data available online, web images are no longer the only data source that may be utilized. Buildings and other objects within a scene can be efficiently collected from geographic information services (GIS). Thus, we propose to use the scene context obtained from GIS instead of the 3D models reconstructed from large scale images to geo-register video frames to world maps.

In recent years, spatial data have become increasingly available on the Internet. Online mapping services enable users not only to consume but also contribute geospatial information voluntarily. For instance, OpenStreetMap (OSM) is an open project that provides user-generated maps of the world. In the early stage of its development, issues such as important landmarks missing or height information of buildings unavailable hindered its utilization in geo-based applications. But as the data contribution growth has continued to rise quickly [5], the map data has been greatly enriched. Nowadays, users can even build three dimensional city models from it easily [28]. It is reasonable to assume

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

DOI: http://dx.doi.org/10.1145/2996913.2997015

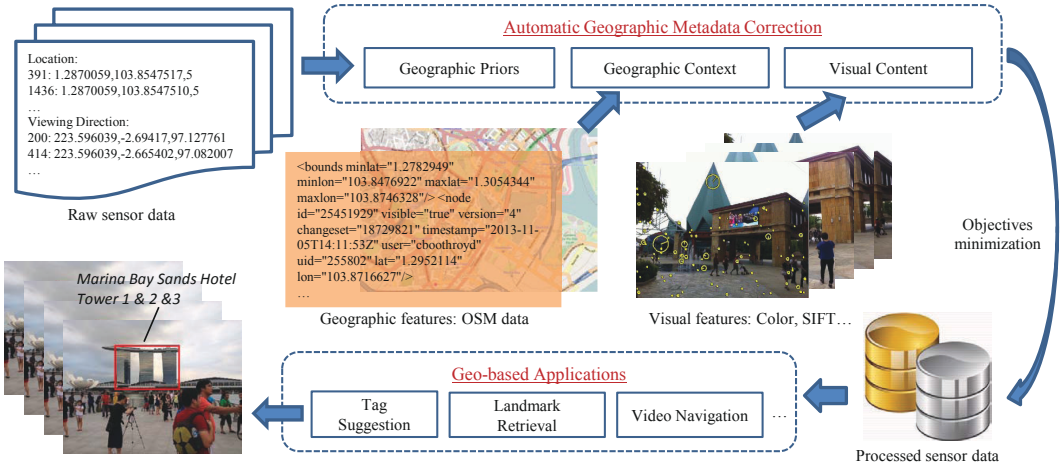


Figure 1: The overall architecture of the proposed automatic geo-metadata correction framework. Raw sensor data is enhanced to provide more accurate geographic information to downstream applications.

that the quality of the spatial data will continue to improve over time. Numerous techniques and solutions can benefit from the valuable information that geo-information services provide about the world.

Figure 1 illustrates the overall architecture of our proposed automatic geo-metadata correction framework. In this study, we mainly focus on the camera orientation correction and formulate the task as an optimization problem by leveraging a set of complementary data sources. First of all, constrained by the geographic priors of the sensor readings, the optimized camera parameters should be near the corresponding input data. Next, we extract local visual descriptors such as SIFT to perform feature matching between consecutive frames for relative rotation estimation. According to Olsson and Enqvist [19], although frames have short baselines that increase the ambiguity in depth determination, it does not have much impact on the rotation estimates. Finally, with the geographic context derived from OSM, we geo-register the frames to the world coordinate system by quantifying the distance between the pixel semantic labels and the 3D projection of the scene. Two distance metrics have been designed and implemented in our system, namely the pixel-based and the pyramid-based measure that encode the spatial information of the semantic labels with different granularity. By minimizing the formulated objectives, we process the raw sensor data to provide more accurate geographic information as an input for downstream applications. Take video annotation as an example, we present a simple approach that can effectively suggest tags to the region of interest in the spatial domain. Here we summarize the contributions of our work as follows:

- We propose to utilize the geographic context derived from OSM for video sensor data correction. Earlier methods only perform feature-based matching with 3D models reconstructed from large scale images.
- We build a comprehensive model to formulate the error terms, which incorporates smooth approximation, rotation estimation, pixel labeling and 3D projection.
- We automatically optimize the geo-metadata while maintaining an excellent balance between accuracy and efficiency compared to existing methods.
- We present the applications of tag suggestion and landmark retrieval with accuracy-enhanced geographic metadata and demonstrate the performance gain.

The rest of the paper is organized is follows. We first report the related work in Section 2 and present the system overview in Section 3. Next we introduce the objectives for geo-metadata optimization in Section 4. Applications are presented in Section 5, which benefit from the accuracy-enhanced camera parameters. Finally, we evaluate the effectiveness of our proposed algorithm in Section 6. Section 7 concludes and suggests future work.

2. RELATED WORK

In multimedia, a significant number of techniques benefit from the presence of geographic metadata associated with images and videos [24, 1]. However, such solutions may sometimes face performance issues due to the occurrence of GPS and compass errors [30, 31]. Traditionally, raw GPS trajectories are usually processed by standard smoothing techniques [4] and map matching algorithms [3]. To produce more precise geographic context, the determination of camera viewing direction has attracted much research attention in recent years. Several content-based computer vision techniques have been proposed based on local feature extraction and matching. Luo *et al.* [17, 18] estimated the viewing directions of world's photos by reconstructing the scenes using a normalized 8-point algorithm. Based on the assumption that the camera location extracted from the geographic metadata is correct, they further geo-registered the photos on Google Maps to assist users in exploring places of interests around the world. Park *et al.* [20] proposed to utilize both Google Street View and Google Earth satellite images to determine the camera orientation of a geotagged image. Kroepfl *et al.* [9] presented a method to geo-locate a photo and then estimate the viewing direction by registering the image onto street level panoramas. However, these methods usually require a large image database to perform reliable object matching. Their effectiveness can sometimes be influenced by the limitations of the data sources, *e.g.*,

Table 1: A comparison with the previous work.

Work	Geo Metadata	Visual Features	Auxiliary Images	Geo Context	Factors with major influences
SfM reconstruction [12, 22, 24]		✓	✓		Large scale images required, high computational cost for 3D scene reconstruction
Image-based matching [9, 20]		✓	✓		Limited availability of Street Views or panoramas with precisely geocoded tags
Location-constrained geo-registration [17, 18]	✓	✓			Noise in the geolocation derived from the metadata, especially when the baseline is small
Geocontext-aware sensor data correction, **proposed**	✓	✓		✓	Availability of the spatial data required for the geographic context derivation of the world

Street Views are only applicable for photos taken on or near road networks [27].

It is one of the central problems in photogrammetry to determine the relative position and orientation among a set of images. Horn [7] presented an iterative method to solve the least-squares problem with more than five correspondences. Snavely *et al.* [24] computed sparse 3D model of a scene and determined the relative camera viewpoints of photographs for interactive 3D browsing. However, these approaches have not dealt with the geo-registration of camera poses with respect to world maps. Benefiting from the developed Structure from Motion (SfM) reconstruction approaches, image-based localization using 3D models of urban scenes has been extensively studied in recent years [13, 29]. Sattler *et al.* [22] utilized 3D scenes reconstructed from Flickr images, and showed that direct 2D-to-3D matching offered considerable potential for accurate image localization. Similarly, Li *et al.* [12] estimated camera poses with respect to a large geo-registered 3D point cloud. Aided by advanced matching techniques, system reliability and efficiency have been further improved. However, such methods might sometimes be limited by their feasibility as the 3D reconstruction step usually requires extensive image collections with large baselines and sufficient overlaps.

As illustrated in Table 1, we have compared our method with the related work in terms of feature sources and bottleneck factors. To the best of our knowledge, there are basically no algorithms designed for efficient compass data correction. The existing techniques mostly focus on camera orientation determination where good accuracies rely on the robust feature matching with extensive computational costs. Moreover, it can be easily seen that the proposed method is the first attempt to consider the geographic context derived from OSM for fine-grained video geo-registration.

3. SYSTEM OVERVIEW

As an overview, we first describe the principles that we have followed in the system design. Next we give a formal description of the inputs and introduce the coordinate systems we have adopted in our framework.

3.1 Design Principles

Our objective is to minimize the errors in the geo-metadata recorded by sensors. To achieve this goal, we have formulated design principles by utilizing a set of complementary data sources as follows:

Prior knowledge:
The geographic metadata recorded by GPS, compass and accelerometer. Goal: the optimized locations and orientations should not drift too far away from the input priors.

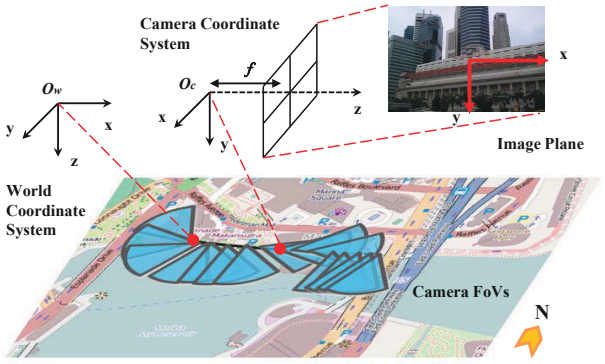


Figure 2: Illustrations of the coordinate systems used in our framework.

Visual content:
The visual clues extracted from frames. Goal: the relative orientation between frames should be consistent with the rotation matrix estimated by keypoint matching.

Geographic context:
The 3D scene built from OpenStreetMap. Goal: video content should be aligned with the 3D scene in respect of the corresponding external camera parameters.

To follow the above criteria, we begin by describing the problem formally.

3.2 Problem Description

Given a sequence of video frames, $S = \{s_1, s_2, ..., s_n\}$, and its associated sensor readings. The video geo-metadata correction problem is formulated as finding the optimal location $L = \{l_1, l_2, ..., l_n\}$ and viewing direction $D = \{d_1, d_2, ..., d_n\}$ sequences that simultaneously satisfy the aforementioned design principles. Note that L and D have the same form with the input priors L^p and D^p derived from the raw sensor data, both of which are formatted as introduced below.

In our framework, we use three coordinate systems to describe the location of a point as shown in Figure 2. The image coordinate system is defined to be located at the centre of the image with x and y axes pointing to right and down, respectively. The origin of the camera coordinate system is located f units before the image plane along the z axis where f is the focal length. The world coordinate system is placed at the geo-coordinates of the first input frame s_1 with x axis pointing to the east and y axis pointing to the south. Subsequently, we interpret the raw sensor readings associated with a frame s_i into the location l_i^p and the viewing direction d_i^p with respect to the world coordinate system. The camera location prior $L^p = \{l_1^p, l_2^p, ..., l_n^p\}$ is

given by $l_i^p = [x_i^p, y_i^p, z_i^p]^\top$ where x_i^p and y_i^p are the UTM coordinates converted from latitude and longitude tuples and z_i^p is related to altitude setting to 1.5 m above ground by default. The camera orientation prior $D^p = \{d_1^p, d_2^p, ..., d_n^p\}$ is presented by $d_i^p = [\alpha_i^p, \beta_i^p, \gamma_i^p]^\top$ which are the angles of yaw (also known as heading), pitch and roll that describe the rotations of the coordinate system around z, y, and x axis, respectively. For example, a positive yaw rotates the camera to the right, the angle of which always equals to the compass reading.

To follow the design principles, we quantify the error terms in the energy function through various distance metrics and discuss the optimization strategy in Section 4.

4. VIDEO GEOREGISTRATION

We start with the introduction of the camera model that we adopt in the framework. To describe the relations between different coordinate systems, we introduce how to compute the external camera parameters based on the raw sensor data and present the formulas for coordinate transformations between different systems. With the above preliminary knowledge, we describe the formulated objectives for error minimization in the raw geo-metadata.

4.1 Camera Model

Without loss of generality, we assume the intrinsic parameter matrix of a camera to be $K = diag([f, f, 1])$. The focal length f is either known for calibrated cameras or can be effectively estimated by content-based approaches [6, 2]. For a 3D point p in the world coordinate system, its corresponding image projection q can be computed based on a rotation matrix R and a translation vector T using the pinhole camera model:

$$\lambda \begin{bmatrix} q \\ 1 \end{bmatrix} = K\left(Rp + T\right) \tag{1}$$

where λ denotes the depth factor. The rotation R and translation T can be derived from L and D, which are the location and viewing direction sequences that need to be optimized. In linear algebra, a rotation matrix is a matrix that is used to perform a rotation in Euclidean space. Using the right hand rule, the three basic rotation matrices that rotate a vector around x, y, or z axis by an angle of θ are given by

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Recall that for the i-th input frame s_i, the camera viewing direction $d_i = [\alpha_i, \beta_i, \gamma_i]^\top$ is given by yaw, pitch, and roll, which are the Tait-Bryan angles representing intrinsic rotations about $z - y' - x''$. Subsequently, the rotation of the camera coordinate system with respect to the world coordinate system can be obtained from the above three elemental intrinsic rotations using matrix multiplication: $R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i)$. According to this change in the coordinate system (also known as passive transformation), the rotation matrix R_i is computed as

$$R_i = \left(R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i)\right)^\top \tag{2}$$

Comparatively, the calculation of translation T_i is quite straightforward, which is simply $T_i = l_1 - l_i$.

4.2 Energy Definition

Given a video sequence associated with geographic metadata, we are interested in finding the optimal locations L and viewing directions D that minimize the following energy function:

$$E = \mu_1 E_{approx} + \mu_2 E_{rotation} + \mu_3 E_{direction} \tag{3}$$

where E_{approx} keeps the outputs from drifting away from the priors too much. $E_{rotation}$ and $E_{direction}$ control the errors of relative rotation and absolute viewing direction, respectively. Parameters μ_1, μ_2 and μ_3 are balancing factors that control the weights assigned to different objectives.

4.2.1 Smooth Approximation

We formulate the approximation requirement as $E_{approx} = E_{approx}^{loc} + E_{approx}^{direc}$. The smoothing cubic spline algorithm [21] is adopted to process the locations $E_{approx}^{loc} = L(t,x) + L(t,y)$ and function $L(\cdot)$ is given by

$$L(t,x) = \rho \sum_{i=1}^{n} \left(\frac{x_i^p - S_x(t_i)}{\sigma_i} \right)^2 + (1-\rho) \int_{t_1}^{t_n} (S_x''(t))^2 dt \tag{4}$$

where t is a sequence of timestamps and $S_x(t)$ is a set of cubic polynomials to fit the observations t and x. The parameters σ_i can be used to change the weight of each point in the error term. We set it to the accuracy measure associated with GPS that indicates the degree of closeness between the GPS reading and the true location. For the approximation of camera viewing direction, we try to minimize the distance between the target D and the input prior D^p described by the sum of L^2 norms, which is

$$E_{approx}^{direc} = \sum_{i=1}^{n} \|d_i - d_i^p\|_2 \tag{5}$$

4.2.2 Relative Rotation

Next we discuss how to estimate the error of relative rotations, $E_{rotation}$. For a 3D point p in the world coordinate system, let q^{s_i} and $q^{s_{i+1}}$ denote its projections on two consecutive frames s_i and s_{i+1}, respectively. If the frames are sampled at a relatively high frequency (*e.g.*, 5 fps), it is reasonable for us to assume that frames s_i and s_{i+1} are taken at the same location. Therefore, according to Eq. 1 we have

$$\lambda_{i+1} \begin{bmatrix} q^{s_{i+1}} \\ 1 \end{bmatrix} = K R_{i+1} R_i^{-1} K^{-1} \cdot \lambda_i \begin{bmatrix} q^{s_i} \\ 1 \end{bmatrix} \tag{6}$$

Given a set of matched keypoints q^{s_i} and $q^{s_{i+1}}$ by feature matching, we are able to rewrite Eq. 6 into a set of linear equations of the form $A_i e_i = 0$, where e_i is a vector consisting of the entries of matrix $KR_{i+1}R_i^{-1}K^{-1}$. Recall that $R_i = \left(R_x(\gamma_i)R_y(\beta_i)R_z(\alpha_i)\right)^\top$, so vector e_i can be written in the form of the camera focal length f and the target viewing direction d_i. Therefore, we seek to optimize the sequence of camera orientations D by minimizing the sum of $\|A_i e_i\|_2$ over the input frames

$$E_{rotation} = \sum_{i=1}^{n-1} \|A_i e_i\|_2 \tag{7}$$

For the keypoint detection and matching, we use SIFT as the visual feature [14]. It provides a local descriptor for each

keypoint including its location, scale and orientation. Thereafter, we match the keypoints between consecutive frames by first querying for the nearest neighbors, followed by using a minimal solver in conjunction with RANSAC to filter out possible outliers. The set of geometrically consistent matches that have been found as described above is used to construct matrices A_i in Eq. 7.

4.2.3 Absolute Viewing Direction

To quantify the error of the absolute viewing direction of a camera is less straightforward and requires additional information of the scene where the video was taken. Recently, image-based localization techniques [12, 22] have been proposed that match photos to pre-built 3D models of the world. Although promising performance gains have been reported, the construction of 3D scenes usually relies on large amounts of high quality input images. Here we argue that photos are no longer the only data source that can be utilized. To facilitate solving problems in computer vision, efforts have been made on building 3D world by extending OSM [28]. Aided by the pre-built 3D world, the scene captured in an image can be well estimated based on the camera parameters derived from the geo-metadata.

Alternatively, we can also try to understand an image scene based on the content by semantic pixel labeling, *e.g.*, the SuperParsing method [25]. As shown in Figure 3, it annotates every pixel with a semantic label (*e.g.*, building, water, road, and *etc.*), which provides a good outline of the semantic classes and their distributions in the image. On the other hand, as we mentioned before, a scene can be labeled based on 3D projection techniques. OSM uses tags, such as building, road, *etc.*, to indicate the category of an object. Therefore, the semantic labels can be derived from the 2D projections of the world on the image plane. We illustrate this idea in Figure 3 by giving two examples, namely the Marina Bay Sands hotel and the Marina Bay Reservoir. If the input of camera location and orientation is close to the ground truth, the 3D projection results should be well aligned with the semantics derived from the content. This observation provides us a simple but effective solution to estimate the absolute viewing direction of a camera.

For a frame s_i, let $Label_c(s_i)$ and $Label_p(s_i)$ denote the semantic labels derived from the image content and the world projection, respectively. Considering the orientation of a camera is a continuous variable that has the form of $d_i = [\alpha_i, \beta_i, \gamma_i]^\top$, it may not be feasible to compute the distance between $Label_c(s_i)$ and $Label_p(s_i)$ every time we change the camera parameters for optimization. Therefore, we alternatively choose to sample a set of virtual scenes $S^r = \{s_1^r, s_2^r, ..., s_m^r\}$ with fixed camera parameters as references, based on which the absolute viewing direction error of frame s_i, denoted by $E^{s_i}_{direction}$, can be estimated as a weighted sum using the following equation

$$E^{s_i}_{direction} = \sum_{j=1}^{m} w_{ij} \cdot Dist\left(Label_c(s_i), Label_p(s_j^r)\right) \quad (8)$$

where w_{ij} denotes the weight of the j-th reference scene s_j^r with respect to frame s_i. Without loss of generality, the reference scenes S^r can be selected by sampling uniformly in each of the six dimensions of the camera pose. w_{ij} should be defined based on the similarity of the camera parameters between s_i and s_j^r, as scenes that are taken within a small area pointing to similar directions can be considered as good

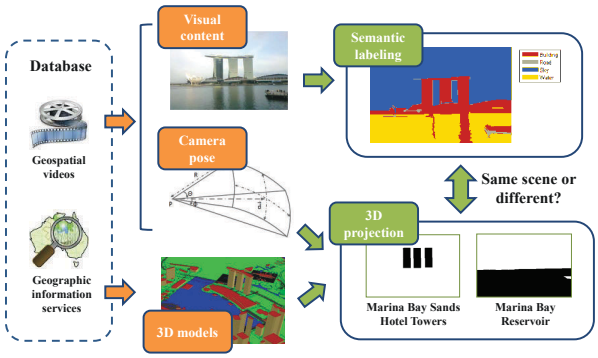


Figure 3: Scene understanding by semantic pixel labeling and 3D projection based on camera pose and OSM data.

representatives for each other. The details about how to decide w_{ij} will be discussed in Section 4.3, as it is related to the selection of S^r and the optimization strategy.

The next task for us is to compute the difference between $Label_c(s_i)$ and $Label_p(s_j^r)$. We select a list of concepts including building, water, road, sky and pedestrian to annotate pixels. Both $Label_c(s_i)$ and $Label_p(s_j^r)$ are matrices whose entries are integer numbers that serve as the index of the pixel labels. Thus, they have the same size as the input frame s_i, denoted by $height(s_i) \times width(s_i)$. In our framework, two distance measures are analyzed. The first one is pixel-based. We count the number of pixels that are labeled with the same concept in both $Label_c(s_i)$ and $Label_p(s_j^r)$ and normalize the value as follows

$$Dist\left(Label_c(s_i), Label_p(s_j^r)\right) = 1 - \frac{numofzeros(Label_c(s_i) - Label_p(s_j^r))}{height(s_i) \cdot width(s_i)} \quad (9)$$

where function $numofzeros(M)$ returns the number of zero entries in matrix M. This measure estimates the pixel-wise distance between two label matrices, but the results can be sometimes susceptible to small changes in camera pose. Inspired by the spatial pyramid matching designed for scene recognition based on local features [11], we also implement a pyramid-based distance measure by partitioning the label matrix into increasingly fine cells and computing histograms of concepts for each cell. More specifically, we construct a spatial pyramid that has a total of L_{pyr} levels. At level $l_{pyr} = 1, 2, ..., L_{pyr}$, the label matrix is partitioned into $2^{l_{pyr}-1}$ sub-regions. For each sub-region, a histogram of concepts is generated by counting the number of times that each label appears. Let $hist^{l_{pyr}}(M)$ be the vector formed by concatenating the histograms generated on level l_{pyr} for a label matrix M. Intuitively, we would like to penalize the features of larger cells because they preserve decreasing spatial information. Therefore, we assign weights $\frac{1}{2^{L_{pyr}-l_{pyr}}}$ to histograms $hist^{l_{pyr}}(M)$ and concatenate the weighted histograms into a feature vector which is $hist(M) = [\frac{hist^1(M)^\top}{2^{L_{pyr}-1}}, \frac{hist^2(M)^\top}{2^{L_{pyr}-2}}, ..., \frac{hist^{l_{pyr}}(M)^\top}{2^{L_{pyr}-l_{pyr}}}, ...]^\top$. Subsequently, the distance between two label matrices can be measured based on this pyramid-based feature as

$$Dist\left(Label_c(s_i), Label_p(s_j^r)\right) = 1 - \frac{hist(Label_c(s_i))^\top hist(Label_p(s_j^r))}{\|hist(Label_c(s_i))\|_2 \cdot \|hist(Label_p(s_j^r))\|_2} \quad (10)$$

We compare the performance of the above two distance measures and the analysis results are discussed later in Section 6. Finally, the absolute viewing direction error $E_{direction}$ in the energy function (see Eq. 3) is estimated by the sum, $E_{direction} = \sum_{i=1}^{n} E_{direction}^{s_i}$.

4.3 Energy Minimization

Inference in our model can be conducted by adopting an efficient two-stage optimization strategy [8]. First, we optimize the location L by minimizing the energy term E_{approx}^{loc}. Next we optimize the viewing direction D by keeping the previously estimated location L fixed.

According to Eq. 4, we smooth the GPS trajectories with cubic splines. As it is a traditional method, here we focus on discussing the optimization of the viewing direction D while keeping the location L fixed. In order to simplify the calculation of w_{ij} in Eq. 8, we sample the virtual scenes S^r at the optimized locations in L instead of a uniform sampling in the 3D space. As discussed before, w_{ij} should be formulated based on the similarity between the camera poses of input frame s_i and reference scene s_j^r. Given the above sampling strategy of S^r, only the virtual scenes that are located at l_i will be considered while computing $E_{direction}^{s_i}$. In other words, let l_j^r and d_j^r denote the location and orientation associated with scene s_j^r. The weight before normalization is $\tilde{w_{ij}} = 0$ if $l_j^r \neq l_i$. Otherwise, we define the orientation difference between d_i and d_j^r, $Dist(d_i, d_j^r)$, to be the degrees by which the unit vector along the z axis $[0, 0, 1]^\top$ rotates from one camera coordinate system to the other. Thereafter, we convert distance to similarity using equation $\tilde{w_{ij}} = 180 - Dist(d_i, d_j^r)$, and normalize the weights by the softmax function,

$$w_{ij} = softmax_j(\tilde{w_{ij}}) = \frac{\exp \tilde{w_{ij}}}{\sum_j \exp \tilde{w_{ij}}} \tag{11}$$

The softmax function reduces the influence of reference scenes whose camera pose greatly differs from the input frame, and limits the weights to have a sum of one. After the normalization, we use the simplex search algorithm [10] to optimize the camera viewing directions D with the initial point setting to the geographic priors D^p derived from the geographic metadata.

5. ENHANCED VIDEO APPLICATIONS

For sensor-rich videos, advanced geo-based methods have been proposed to facilitate fast video search and browsing on the Internet [23]. Compared with compute-intensive content-based techniques, geo-based methods have significantly improved system efficiency by alternatively processing the sensor metadata instead. With more accurate geo-metadata, considerable performance gain can subsequently be obtained in downstream applications. We present two examples here, namely landmark retrieval and tag suggestion.

5.1 Landmark Retrieval

With the geographic metadata associated with a frame, a landmark's visibility can be determined efficiently given the height and the footprint of the target building [30]. As aforementioned, such geometry information can be easily collected from online mapping services. In terms of the camera's geometry, its viewable scene is usually characterized by the following five parameters: (1) camera position and viewing direction that are extracted from sensor metadata, (2) horizontal and vertical viewable angles, and the far visible distance that are estimated from camera optics. Thereafter, for a queried landmark, its visibility can be easily computed by geometry calculations and occlusion checks. As has been pointed out [30], the effectiveness of such geo-based methods highly depend on the accuracy of the geo-metadata. By simply reducing the noise in the raw sensor data, we will see that significant improvements can be obtained without any adaptation of the original method.

ALGORITHM 1: Tag suggestion processor.

Input: the region of interest to suggest tags ROI; the set of nearby geographic objects $GeoObjs$; and the accuracy-enhanced camera location L and viewing direction D.
Output: the predicted list of tags $Tags$.

```
Candidates = projection(GeoObjs, L, D)
for i = 1 to length(Candidates) do
    C = Candidates(i).region
    distloc = ||centre(C) − centre(ROI)||
    distsize = height(C) / height(ROI)
    if distsize < 1 then distsize = 1 / distsize;
    distshape = (height(C)·width(ROI)) / (width(C)·height(ROI))
    if distshape < 1 then distshape = 1 / distshape;
    dist(i) = distloc · distsize · distshape
end
[mindist, idx] = min(dist)
Tags = Candidates(idx).tags
```

5.2 Tag Suggestion

Imagine that for interactive videos, users can draw rectangles to indicate their interests while watching a video. The system should be able to immediately suggest a set of tags describing the objects in the bounding box. This functionality can be easily implemented by ranking the projections of nearby geographic objects with respect to the input bounding box, using the distance measure described in Algorithm 1.

Let ROI denote the region of interest specified by a user or detected by automatic algorithms [33]. We first compute a list of candidates by projecting the 3D models of $GeoObjs$ onto the image plane with $projection(GeoObjs, L, D)$. Note that each candidate has two attributes, the projection region and the corresponding object names (tags). As the input bounding box may cover multiple objects, we also check the projections of possible groups formed by geographic objects located close to each other. Next, for every candidate, we compute the Euclidean distance between the centres of C and ROI and penalize the bounding boxes that differ in shape or size compared to ROI. The weighting factor for size difference is defined to be a division between the heights with the numerator being whichever is bigger. The shape of a bounding box is parameterized by the ratio of the height to the width, and the difference is quantified in the same way as for the size. Finally, we search for the candidate with the minimum distance value and return the corresponding list of tags to the users.

6. EVALUATION

We implemented our proposed algorithm and evaluated its effectiveness. We proceed in three steps. The first part introduces the dataset we collected and used for the experiments. The second and third parts evaluate the performance of the proposed model in geographic metadata correction and downstream applications, respectively.

6.1 Experimental Setup

We evaluated our proposed algorithm on the publicly available geo-referenced video dataset [16] from the GeoVid[1] website. Users can record and share videos using the GeoVid smartphone applications, or explore the world by watching videos via a web browser. Moreover, the GeoVid project also provides APIs[2] for users to obtain public videos together with their corresponding geographic metadata.

To evaluate our approach, we manually annotated the ground truth of camera poses and landmark visibility based on map services (*e.g.*, Google Maps and Google Street View). We randomly selected ten sensor-rich videos taken in Singapore to carry out the experiments in Section 6.2. The description of the dataset is illustrated in Table 2. The average video duration of this dataset is 28 seconds. We sampled frames every three seconds to let users perform the ground truth annotation and interpolated the camera parameters between the sampled frames for later comparisons. Additionally, we labeled the visibility of four landmarks on 8,430 frames sampled at 1 fps from 224 videos to perform the retrieval evaluations in Section 6.2. Please note that we utilized a larger test set of 224 videos in Section 6.2 due to the following reasons: (1) it is easier to annotate the visibility of a landmark in a frame than to determine the ground-truth camera viewing direction; and (2) the improved results in the downstream applications also indicate the effectiveness of our proposed method and emphasize the importance of geo-metadata correction.

Table 2: Georeferenced video dataset description.

Video duration	Shortest	Longest	Average
	20 sec	62 sec	28 sec
No. of videos	10 videos with 83 ground truth labels		

The dataset might still be small mostly because of the effort needed to obtain the ground truth annotations, but its size is comparable to other camera orientation determination papers [20, 18]. Moreover, to the best of our knowledge, this work is among the early efforts to solve the problem of automatic geo-metadata correction for video sequences.

6.2 Geographic Metadata Correction

We processed the raw geo-metadata and present the error reduction results. The GPS accuracy of our test dataset is good, as all the accuracy measures associated with GPS (σ_i in Eq. 4) are less than or equal to five. Since this work focuses on the correction of the orientation data, at the current stage we simply processed locations by the traditional smoothing technique with cubic splines. Here we report the smoothing result on a more challenging dataset (the accuracy value $max(\sigma_i) > 50$) [26] in Table 3. The parameter ρ in Eq. 4 was set to 0.6. We show the precision before and after processing at different geographic margins of error.

Table 3: Precision comparison of raw and processed GPS data.

Radius	10 m	20 m	30 m	40 m	50 m
Raw Data	68.2%	91.0%	92.0%	92.9%	93.4%
Processed	70.9%	92.0%	93.4%	94.5%	95.2%

As can be seen, there was an improvement on location accuracy within all error margins. On average, the smoothing splines were able to reduce the error per frame by 27.32%. Further improvements can be obtained by applying more advanced techniques, such as mapping GPS traces to road maps [3]. Those approaches can be integrated into our framework easily as we adopt a two-stage optimization strategy by processing camera location and camera viewing direction separately in different modules.

Next, we compared the camera orientation errors, and report the results in Figure 4. For the content-based semantic pixel labeling, we adopted the SuperParsing method proposed by Tighe and Lazebnik [25]. Only the images that are geographically close to the test videos are used for training, in order to ensure the accuracy of this supervised image parsing technique. Recall that the viewing direction of a camera has the form of $d = [\alpha, \beta, \gamma]^\top$. As most of the users hold the camera perpendicular to the ground while taking a video, the variations in pitch and roll are usually very small (*i.e.*, $\beta \approx 0^\circ$ and $\gamma \approx 90^\circ$). Therefore, we focus on evaluating the correction of yaw, α, and define the error to be the absolute angle difference between the measured and the true values in degrees. In other words, let α_i^t and α_i^e denote the true and the estimated camera heading for frame f_i. The error δ_i is computed as $\delta_i = \min\left(\|\alpha_i^e - \alpha_i^t\|, 360 - \|\alpha_i^e - \alpha_i^t\|\right)$. For an input video, the orientation error is computed as the average of its frames, *i.e.*, $E = \frac{1}{n}\sum_{i=1}^{n}\delta_i$.

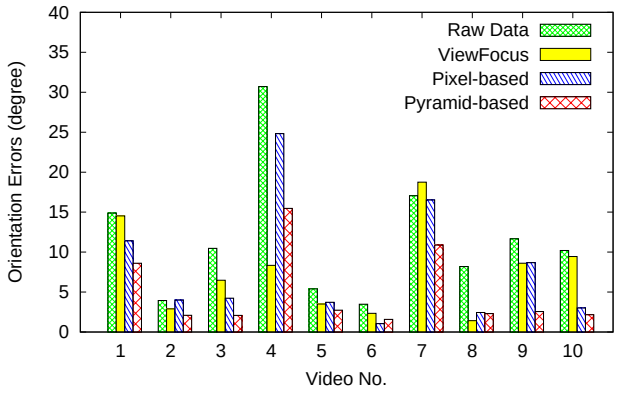
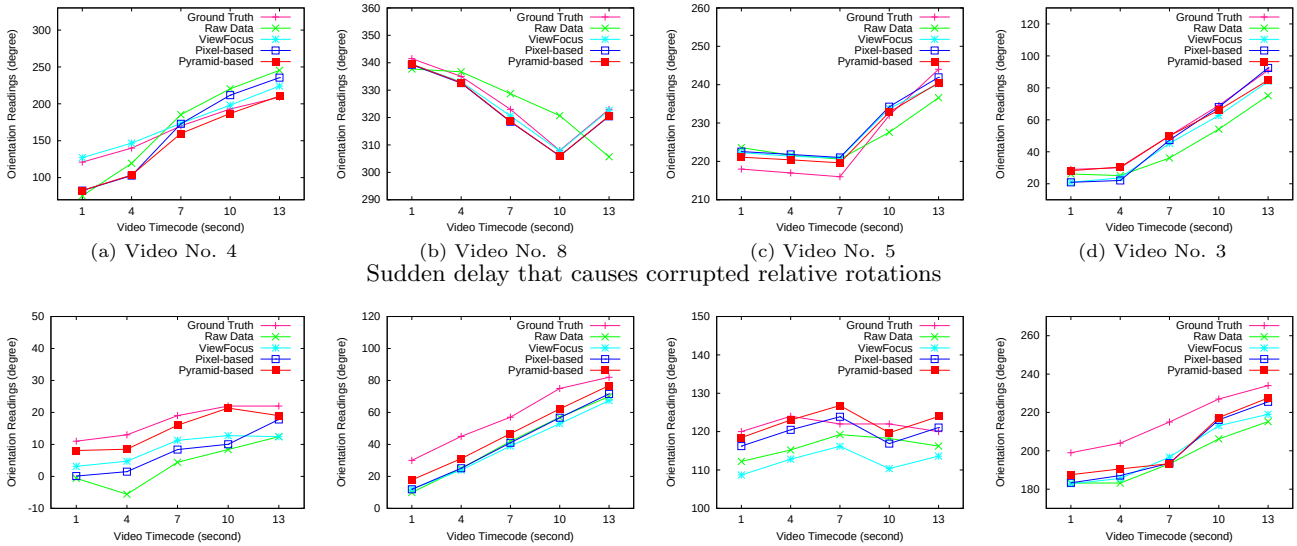


Figure 4: Raw and processed camera orientation error comparison for individual videos.

Based on the measurement above, we compared our proposed method with ViewFocus, which is the most related to our work that determines the camera direction with the existence of geo-metadata [17, 18]. Considering the baseline between frames is usually small, we further optimized the result of ViewFocus by conjunctively minimizing the distance to both the estimated external camera parameters and

[1]http://geovid.org/
[2]http://api.geovid.org



(a) Video No. 4 (b) Video No. 8 (c) Video No. 5 (d) Video No. 3

Sudden delay that causes corrupted relative rotations

(e) Video No. 9 (f) Video No. 7 (g) Video No. 10 (h) Video No. 1

Orientation shift where the relative rotations stay approximately accurate

Figure 5: Effectiveness analysis of sensor data correction algorithms with or without geographic context and its connections to the error patterns in the camera orientation readings.

the raw geographic priors (both location and orientation). For the image-based methods discussed in the related work (see table 1), it is difficult to perform a fair comparison due to the lack of third-party auxiliary images. Moreover, such techniques are not always applicable, as the appearance of at least one geo-object in the content is required to perform robust feature matching and 3D reconstruction. As shown in Figure 4, raw data represents the geographic priors derived from the input sensor readings. Pixel-based (see Eq. 9) and Pyramid-based (see Eq. 10) indicate the distance measure we used to quantify the difference between two label matrices. The reference scenes S^r were sampled at the optimized locations of the input frames with viewing directions sampled uniformly every 10 degrees. The balancing coefficients in Eq. 3 were set to $\mu_1 = 1$, $\mu_2 = 0.02$, and $\mu_3 = 1000$.

The average error reduction obtained by ViewFocus was 31.4%. Without considering the geo-context derived from OSM, it was only able to work well on certain videos (*e.g.*, video 4), while being less effective for the rest of the cases. Actually the correction effectiveness of ViewFocus is related to the error patterns of the geo-metadata. We will discuss this in the next paragraph by showing some examples. Among the three approaches, the pyramid-based method is the most effective and outperforms its competitors in eight out of the ten cases. It obtained an average error reduction of 58.8%, where the best and the worst cases were an 80.1% and 36.2% error decrease, respectively. Compared with the pixel-based distance measure, the pyramid-based approach achieved an average of 27.6% improvement over the former. This is mostly because the value of the pixel-based measure is susceptible to the changes in camera pose. Even a small shift in camera orientation may have a big impact on the result of the pixel-based distance measure. This might cause some issues as we sampled the reference scenes S^r with a relatively coarse granularity. It is possible to further improve the effectiveness by adopting a more fine-grained sampling approach, but this will also increase the computational complexity. Comparatively, the pyramid-based measure achieved better results as it is less sensitive to camera changes while encoding part of the spatial information of the semantic labels into the distance calculation.

To better understand real world effects, we further examined the raw, the processed and the ground truth camera orientation sequences in our test dataset. For the eight videos where the average orientation error of the raw geo-metadata was larger than five degrees (videos 2 and 6 were excluded), we plotted the compass readings in the beginning 13 seconds of each video in Figure 5. The graphs were sorted ascendantly according to $E_{ViewFocus} - E_{Pyramid}$, which is the difference between the orientation errors obtained by the pyramid-based and the ViewFocus approach. In other words, we show the plots with increasing effectiveness of the former method *w.r.t.* the latter from Figure 5(a) to 5(h). As can be seen, interestingly the videos in the first and the second row exhibited different inaccuracy patterns of the raw geo-metadata. While ViewFocus worked well on cases where the raw compass readings were distributed around the truth values and the inaccuracy mostly came from the relative rotation errors (*e.g.*, Figures 5(a)), it became highly ineffective to handle the camera orientation shift without considering the geographic context of the world. As shown in the second row, the orientation shift resulted in the incorrectness of the absolute orientation values while the relative rotations stayed approximately accurate. By applying our proposed optimization strategy, this kind of error can be effectively reduced by the third energy term, $E_{direction}$, in Eq. 3, which matches the image scene to the projections of the world. Moreover, the second energy term $E_{rotation}$ limits the error of the relative rotation between consecutive frames. This part is similar to ViewFocus, which is capable of correcting the corrupted compass readings caused by sudden delays or outliers. To summarize, our proposed model is more general, which handles all error patterns effectively.

Table 4: Comparison of landmark retrieval effectiveness with raw and corrected geo-metadata.

(a) Singapore Flyer

	Precision	Recall	F-measure
Raw Data	0.6497	0.7099	0.6785
Processed	0.7448	0.8827	0.8081

(b) Esplanade

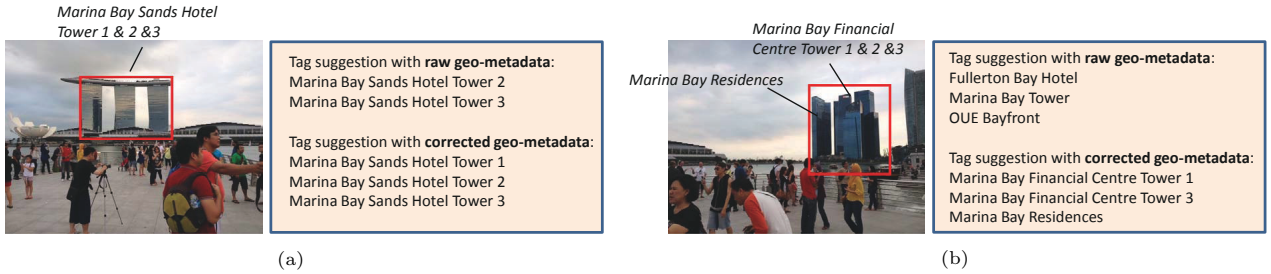
	Precision	Recall	F-measure
Raw Data	0.7740	0.9752	0.8630
Processed	0.8324	0.9536	0.8889

(c) Merlion

	Precision	Recall	F-measure
Raw Data	0.9693	0.6819	0.8006
Processed	0.9903	0.6873	0.8115

(d) The Float @ Marina Bay

	Precision	Recall	F-measure
Raw Data	0.7773	0.7558	0.7664
Processed	0.7769	0.8664	0.8192



(a) (b)

Figure 6: Illustrations of tag suggestion with raw and corrected geo-metadata. The *ground truth annotations* are marked in italic. The input is the object-of-interest automatically detected by Edge Boxes with the highest objectness score.

6.3 Landmark Retrieval and Tag Suggestion

Geographic contextual information has long been utilized as a supplement to content analysis. As an example, we clean the geo-metadata by our proposed method and discuss the results on landmark retrieval (see Section 5.1) and video tagging (see Section 5.2). We show that enhanced results can be obtained by simple techniques when the errors in the metadata have been reduced and controlled within reasonable margins.

We selected four landmarks, namely the Singapore Flyer, the Esplanade, the Merlion, and the Float @ Marina Bay to perform a landmark retrieval [30] experiment on 8430 frames. The precision, recall and F-measure obtained with raw and corrected geo-metadata are reported in Table 4. Note that the F-measure is a good indicator for the retrieval effectiveness and the geo-metadata quality since it considers both precision and recall. As can be seen, considerable improvements have been achieved in all cases. The errors in the raw sensor data significantly hindered the retrieval of the Singapore Flyer, as the F-measure was only reported to be 0.6785. In this case, the optimized geo-metadata obtained the greatest performance gain of 14.6%, 24.3% and 19.1% in terms of precision, recall and F-measure. Comparatively, the frames that capture the rest of the three landmarks were associated with less noisy geo-metadata. Under such circumstances, we were still able to improve the F-measure by 1.34% $\sim$ 6.89% with the accuracy-enhanced geo-metadata.

Next, we show two examples of tag suggestion to region-of-interest in Figure 6. Given the bounding box that indicates a user's interest, we suggested tags by predicting the geographic objects covered by the input region. As illustrated, the precision of the suggested tags was improved after we performed the geo-metadata correction. For relatively isolated buildings, such as the one shown in Figure 6(a), the annotation results tend to be more robust to the noise in the geo-metadata. However, in areas where the building density is high, a small error in camera viewing direction can sometimes cause serious performance issues. For example in Figure 6(b), the tags suggested using the raw geographic metadata are actually the names of some nearby buildings other than the correct ones. Problems like this can be solved, at least to some extent, by pre-processing the geographic metadata with our proposed optimization technique.

To summarize, the above comparisons indicate the importance of conducting geo-metadata correction before utilization. The proposed optimization method is orthogonal to downstream video applications and can be applied as the first step in geo-based video management systems.

7. CONCLUSION AND FUTURE WORK

We formulated the sensor data correction as an optimization problem. To improve the efficiency and the feasibility of the framework, we built 3D scenes based on OSM data. Next, we projected the 3D models onto the image plane and compared it to the image scene analyzed by pixel labeling. This technique provided us with an efficient way to quantify the absolute viewing direction error of a camera. By analyzing the real-world data, we draw a number of interesting observations that we summarize as follows:

(i) The geo-metadata errors can be roughly divided into two categories by checking if there are serious corruptions in terms of the relative rotation. Content-based approaches can effectively reduce rotation errors between consecutive frames, but without the context of the scene it becomes highly difficult to correct orientation shift where the relative rotations are approximately accurate.

(ii) Most of the existing image-based methods are only applicable to photos that clearly capture at least one object in

order to perform robust keypoint matching and reconstruction. Comparatively, we geo-register cameras by conjunctively considering the distribution of geo-objects and the rotation consistency in the temporal domain. Good estimation can be obtained as long as the landscape, where the video was taken, is fairly diverse towards different directions.

(iii) One factor that may have an impact on our approach is the detail level of the spatial data available from mapping services, *e.g.*, the label matrix generated by 3D projection can be imprecise due to missing buildings. Fortunately, with the rapid growing collection of map data, it is reasonable to expect that the proposed method will be able to geo-register video sequences with increasing accuracies in the future.

At the current stage, the geo-based 3D projection and the content-based semantic pixel labeling are regarded as two separate modules in our framework. As part of the future work, we are interested in developing a joint camera geo-registration and image scene understanding algorithm to further improve the results in both of the subtasks.

8. ACKNOWLEDGMENTS

This research has been supported in part by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office through the Centre of Social Media Innovations for Communities (COSMIC), by the National Natural Science Foundation of China under Grant no. 61472266 and by the National University of Singapore (Suzhou) Research Institute, 377 Lin Quan Street, Suzhou Industrial Park, Jiang Su, People's Republic of China, 215123.

9. REFERENCES

[1] S. A. Ay, R. Zimmermann, and S. H. Kim. Viewable Scene Modeling for Geospatial Video Search. In *ACM Multimedia*, pages 309–318, 2008.

[2] S. Bennett, J. Lasenby, A. Kokaram, S. Inguva, and N. Birkbeck. Reconstruction of the Pose of Uncalibrated Cameras via User-Generated Videos. In *ACM ICDSC*, pages 3:1–3:8, 2014.

[3] L. Cao and J. Krumm. From GPS Traces to a Routable Road Map. In *ACM SIGSPATIAL GIS*, pages 3–12, 2009.

[4] F. Chazal, D. Chen, L. Guibas, X. Jiang, and C. Sommer. Data-driven Trajectory Smoothing. In *ACM SIGSPATIAL GIS*, pages 251–260, 2011.

[5] M. Haklay and P. Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[6] R. I. Hartley. Self-Calibration of Stationary Cameras. *Int. J. Comput. Vision*, 22(1):5–23, 1997.

[7] B. K. P. Horn. Relative Orientation Revisited. *Journal of the Optical Society of America A*, 8:1630–1638, 1991.

[8] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyper-lapse videos. *ACM Transactions on Graphics*, 33(4):78:1–78:10, 2014.

[9] M. Kroepfl, Y. Wexler, and E. Ofek. Efficiently Locating Photographs in Many Panoramas. In *ACM SIGSPATIAL GIS*, pages 119–128, 2010.

[10] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[12] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. Worldwide Pose Estimation Using 3D Point Clouds. In *ECCV*, pages 15–29, 2012.

[13] H. Liu, T. Mei, J. Luo, H. Li, and S. Li. Finding Perfect Rendezvous on the Go: Accurate Mobile Visual Localization and Its Applications to Routing. In *ACM Multimedia*, pages 9–18, 2012.

[14] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2Trip: Generating Travel Routes from Geo-tagged Photos for Trip Planning. In *ACM Multimedia*, pages 143–152, 2010.

[16] Y. Lu, H. To, A. Alfarrarjeh, S. H. Kim, Y. Yin, R. Zimmermann, and C. Shahabi. GeoUGV: User-generated mobile video dataset with fine granularity spatial metadata. In *ACM MMSys*, pages 43:1–43:6, 2016.

[17] Z. Luo, H. Li, J. Tang, R. Hong, and T.-S. Chua. ViewFocus: Explore Places of Interests on Google Maps Using Photos with View Direction Filtering. In *ACM Multimedia*, pages 963–964, 2009.

[18] Z. Luo, H. Li, J. Tang, R. Hong, and T.-S. Chua. Estimating Poses of World's Photos with Geographic Metadata. In *Advances in Multimedia Modeling*, volume 5916, pages 695–700. 2010.

[19] C. Olsson and O. Enqvist. Stable Structure from Motion for Unordered Image Collections. In *Image Analysis*, volume 6688, pages 524–535. 2011.

[20] M. Park, J. Luo, R. T. Collins, and Y. Liu. Beyond GPS: Determining the Camera Viewing Direction of a Geotagged Image. In *ACM Multimedia*, pages 631–634, 2010.

[21] D. Pollock. Smoothing with Cubic Splines. Department of Economics, Queen Mary and Westfield College, 1993.

[22] T. Sattler, B. Leibe, and L. Kobbelt. Fast Image-based Localization using Direct 2D-to-3D Matching. In *ICCV*, pages 667–674, 2011.

[23] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic Tag Generation and Ranking for Sensor-rich Outdoor Videos. In *ACM Multimedia*, pages 93–102, 2011.

[24] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH*, pages 835–846, 2006.

[25] J. Tighe and S. Lazebnik. Superparsing: Scalable Nonparametric Image Parsing with Superpixels. In *ECCV*, pages 352–365, 2010.

[26] G. Wang, B. Seo, and R. Zimmermann. Automatic Positioning Data Correction for Sensor-annotated Mobile Videos. In *ACM SIGSPATIAL GIS*, pages 470–473, 2012.

[27] G. Wang, Y. Yin, B. Seo, R. Zimmermann, and Z. Shen. Orientation Data Correction with Georeferenced Mobile Videos. In *ACM SIGSPATIAL GIS*, pages 400–403, 2013.

[28] S. Wang, S. Fidler, and R. Urtasun. Holistic 3D Scene Understanding From a Single Geo-Tagged Image. In *CVPR*, 2015.

[29] X. Xu, T. Mei, W. Zeng, N. Yu, and J. Luo. AMIGO: Accurate Mobile Image Geotagging. In *ICIMCS*, pages 11–14, 2012.

[30] Y. Yin, B. Seo, and R. Zimmermann. Content vs. Context: Visual and Geographic Information Use in Video Landmark Retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(3):39:1–39:21, 2015.

[31] Y. Yin, Y. Yu, and R. Zimmermann. On generating content-oriented geo features for sensor-rich outdoor video search. *IEEE Transactions on Multimedia*, 17(10):1760–1772, 2015.

[32] B. Zhang, Q. Li, H. Chao, B. Chen, E. Ofek, and Y.-Q. Xu. Annotating and Navigating Tourist Videos. In *ACM SIGSPATIAL GIS*, pages 260–269, 2010.

[33] C. L. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, volume 8693, pages 391–405. 2014.