# GPS-Tag Refinement using Random Walks with an Adaptive Damping Factor

Amir Roshan Zamir Shervin Ardeshir Mubarak Shah
Center for Research in Computer Vision, UCF

## Abstract

*The number of GPS-tagged images available on the web is increasing at a rapid rate. The majority of such location tags are specified by the users, either through manual tagging or localization-chips embedded in the cameras. However, a known issue with user shared images is the unreliability of such GPS-tags. In this paper, we propose a method for addressing this problem. We assume a large dataset of GPS-tagged images which includes an unknown subset with contaminated tags is available. We develop a robust method for identification and refinement of this subset using the rest of the images in the dataset. In the proposed method, we form a large number of triplets of matching images and use them for estimating the location of the query image utilizing structure from motion. Some of the generated estimations may be inaccurate due to the noisy GPS-tags in the dataset. Therefore, we perform Random Walks on the estimations in order to identify the subset with the maximal agreement. Finally, we estimate the GPS-tag of the query utilizing the identified consistent subset using a weighted mean. We propose a new damping factor for Random Walks which conforms to the level of noise in the input, and consequently, robustifies Random Walks. We evaluated the proposed framework on a dataset of over $18k$ user-shared images; the experiments show our method robustly improves the accuracy of GPS-tags under diverse scenarios.*

## 1. Introduction

Due to the emergence of mobile devices with various internal positioning methods, the majority of images taken nowadays can be geo-tagged at the time of collection. However, different tagging systems, e.g. GPS, WiFi positioning system (WPS), Cell positioning system, manual tagging, etc. have a broad range of reliability and accuracy which altogether translate into inaccuracies in the geo-tags of user-shared images. These inaccuracies can become critical for the applications which are based on crowdsourced images, such as 3D reconstruction [1] or image localization [5, 8].

In general, the fact that user-shared images typically have inaccurate and unreliable GPS-tags is well known and is acknowledged in several previous works [20, 5, 12]. In this paper, we propose a method for *GPS-tag refinement*. That is, given a dataset of GPS-tagged images with an unknown subset which includes inaccuracies in the tags, we discover the contaminated subset and adjust its GPS-tags to the correct locations. We achieve this goal utilizing the rest of the images in the dataset (i.e. self-refinement) without using any other source of imagery or data. We accomplish this task by generating a large number of estimations for the location of a particular image in the dataset based on the rest of the images therein. Then, we use a robust method based on Random Walks which identifies the reliable estimations based on their pairwise consistency and use them for computing the refined GPS-tag. *Robustness* is the key trait of the proposed method. We show that our approach achieves good characteristics in this regard, such as high Breakdown Point or descending Influence Function [6] (section 3).

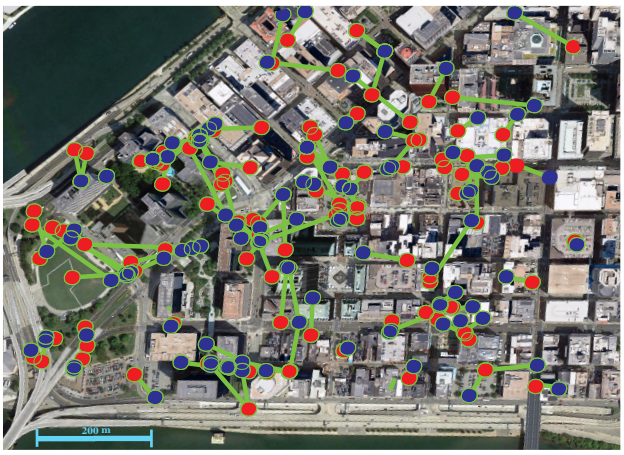


Figure 1. The user-specified GPS-tags (blue) of about 100 images from Pittsburgh along with their correct GPS-locations (red). The green line connects the user-specified location to the ground truth. Significant inaccuracies in the GPS labels can be observed. In this paper, we propose a robust method for refining the GPS-tags.

Besides the inaccurate geo-tags, another main characteristic of user-shared images is their non-uniform distributions with respect to different locations. Many factors, such as the layout of the city or the dynamics of the area could cause the non-uniformity. We argue that the nonuniform distribution could act as a bias in various applications, par-

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.545

ticularly image localization and GPS-tag refinement. We incorporate the geo-distribution of user images a priori in our framework in order to cope with the bias it causes.

Random Walks include a term (damping factor) which is primarily intended for injecting the prior knowledge about the data in the diffusion process. We use this term for incorporating the geo-distribution in the refinement. However, we argue that the conventional *constant* damping factor makes Random Walks prone to noise in the input. Instead, we propose an *adaptive* damping factor which conforms to the estimated level of noise in each input data point and consequently robustifies Random Walks.

Various operations on *textual* tags of images such as labeling[9], ranking [10] and refinement [19] have been extensively explored in the literature. The main differences between our work and the aforementioned ones are refining the GPS-tags, which are numerical and consequently pose a problem with different properties compared to textual tags, and maintaining robustness as a key factor.

Additionally, several methods for automatic image geo-localization have been recently developed [5, 8, 12]. These methods often require a reference dataset (e.g. Street View) with presumably accurate GPS-tags, whereas our approach performs *self-refinement*, has an internal robustness mechanism, and effectively uses the *initial* GPS-tag of the image.

In the Structure from Motion literature, Crandall et al. [2] developed a method for adjusting camera parameters using a graph which spans the whole dataset and by performing a global optimization for all of the images simultaneously. On the contrary, we refine the camera location of one image at a time (our graph includes the location estimations for one image instead of all images) which yields a significant speedup without sacrificing the robustness. Zach et al. [18] proposed a loop constraint for finding incorrect geometric relations between images which results in a better reconstructed 3D model. However, estimating the global GPS-tags of the cameras (even given a perfectly reconstructed local model) when the original reference GPS-tags include outliers is a question which is not in the scope of their work as well as many other Structure from Motion and bundle adjustment methods. Havlena et al. [4] used image triplets, instead of pairs, in 3D reconstruction with the intention of having more reliable initial atomic reconstructions, whereas we employ triplets to remove the scale ambiguity and estimate as many independent estimations as possible for the GPS location of the camera.

The main contributions of this paper can be summarized as: 1) a novel framework for robust refinement of GPS-tags using Random Walks. 2) a new adaptive damping factor for Random Walks. 3) a large scale study of the statistical properties of noise in the GPS-tags of crowdsourced images.

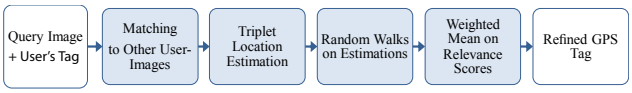


Figure 2. The block diagram of the proposed method.

## 2. Robust Tag Refinement

The block diagram of the proposed method is shown in figure 2. Given a large dataset of images with contaminated geo-tags, first we perform content-based matching between the query image, which is one of the dataset images, and the rest of the dataset and retrieve a number of matches. Then, a large number of image triplets comprised of the query and each feasible pair of retrieved matches are formed. We perform structure from motion (SfM) on each triplet which yields one estimation for the location of the query. Since a considerable percentages of these estimations are inaccurate, we perform Random Walks on a graph defined on the estimations to discover the accurate subset. The final estimation of the query's location is obtained by finding the mean of the estimations weighted by the scores acquired from the Random Walks. The details of each step are provided in the following sections.

### 2.1. Generating Estimations using Triplets

We match the query image, $\mathcal{I}$, against the rest of the images in the dataset and retrieve $\mu$ matches $\{m_1, m_2, \ldots, m_\mu\}$. We use bag of SIFT words with a vocabulary size of $50k$ for matching [11].

Next, $\binom{\mu}{2}$ image triplets composed of the query image and each feasible pair of the retrieved matches are formed. We estimate the relative location of the query image with respect to the two matched images by finding the trifocal tensor and performing SfM [17, 16]. For the triplet $\{\mathcal{I}, m_i, m_j\}$, this operation yields $\{\boldsymbol{l}_\mathcal{I}, \boldsymbol{l}_i, \boldsymbol{l}_j\}$ which are the camera locations of $\mathcal{I}$, $m_i$ and $m_j$ in the coordinate system returned by SfM (which is usually centered at one of the camera locations), respectively. Note that the locations $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, and $\boldsymbol{l}_j$ are typically three dimensional. However, any arbitrary three points fall on a plane, and therefore, their coordinates can be two dimensional. Hence, assuming the images were taken on a roughly flat region, we can reduce the dimensionality of $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, $\boldsymbol{l}_j$ to two (e.g. using PCA).

We want to have an estimation of the GPS-tag of $\mathcal{I}$ using the triplet. Therefore, the relative locations $\boldsymbol{l}_\mathcal{I}$, $\boldsymbol{l}_i$, and $\boldsymbol{l}_j$ should be transformed from the SfM coordinates system to the global GPS coordinate system[1]. These two Cartesian coordinate systems are related by a similarity transforma-

[1]Note that GPS locations are usually specified by Latitude and Longitude values which are in spherical coordinate system. However, they can be easily converted to a Cartesian system called East, North, Up (ENU). Therefore, for the sake of simplicity and without loss of generality, we assume all of the GPS coordinates in this paper are in this Cartesian system. We use the two dimensional version (**East-North**) of this system.

tion consisting of rotation, translation and scaling:

$$\begin{bmatrix} \boldsymbol{g} \\ 1 \end{bmatrix} = (\boldsymbol{RST}) \begin{bmatrix} \boldsymbol{l} \\ 1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{l}$ is a point in the SfM coordinate systems and $\boldsymbol{g}$ is its corresponding point in the global GPS coordinate system; $\begin{bmatrix} \boldsymbol{g} \\ 1 \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{l} \\ 1 \end{bmatrix}$ are homogeneous coordinates of $\boldsymbol{g}$ and $\boldsymbol{l}$, respectively. $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ denote the $3 \times 3$ rotation, scaling and translation matrices. At least two pairs of $\boldsymbol{g} \leftrightarrow \boldsymbol{l}$ correspondences are needed in order to calculate the $\boldsymbol{RST}$ transformation of equation 1. Since the two matches $m_i$ and $m_j$ are GPS-tagged, we use their GPS-tags and $\boldsymbol{l_i}$ and $\boldsymbol{l_j}$ to compute $\boldsymbol{RST}$ of the triplet. This transformation is then used for finding the location of $\mathcal{I}$ in the GPS coordinate system: $\begin{bmatrix} \boldsymbol{g_\mathcal{I}} \\ 1 \end{bmatrix} = (\boldsymbol{RST}) \begin{bmatrix} \boldsymbol{l_\mathcal{I}} \\ 1 \end{bmatrix}$. Since we have $\binom{\mu}{2}$ feasible triplets, we will have $\binom{\mu}{2}$ different estimations for the GPS-location of the query using the described method.

We assumed the query image and its matches were taken on a roughly flat surface and reduced the dimensionality of the locations to two. An alternative way would be to keep the coordinates three dimensional and use quadruplets instead of triplets (since one more correspondence would be needed to compute $\boldsymbol{RST}$ in 3D). However, that would be undesirable since a quadruplet is more likely to be affected by noisy GPS-tags (as it has one more image), and thus, the overall percentage of the accurate estimations would drop.

## 2.2. Robustification Using Random Walks

The estimation of the GPS-location of $\mathcal{I}$ which a triplet yields is accurate only if both of the parent reference images have accurate GPS-tags. Since we assume an unknown subset of the images in the dataset have inaccurate GPS-tags, a considerable number of the estimations are inaccurate. However, unlike the inaccurate estimations, the accurate ones are expected to show a high consistency with each other. Therefore, we use Random Walks for discovering the reliable subset of estimations and assigning a score to each. Intuitively, Random Walks diffuse the score of one node to the neighboring ones if they have a high consistency. This can be imagined by assuming a person is to walk from one node of a graph to another and count the number of times each node is visited; the probability of the next node to travel to is determined by a predefined consistency between the nodes. If the number of visits to each node is interpreted as a score, after a large number of walks, the nodes which are more consistent to one another will have a higher final score as they are visited more often.

We define the graph $G = (\boldsymbol{N}, E)$ where $\boldsymbol{N}$ and $E$ represent the set of node and edges. Each node represents one estimation, i.e. $\boldsymbol{N} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_\lambda\}$, and there is an edge between each pair of nodes, $E = \{(\boldsymbol{g_i}, \boldsymbol{g_j}), i \neq j\}$. We include the original GPS-tag of $\mathcal{I}$, as an estimation for its correct GPS-location, in $\boldsymbol{N}$ as well. Therefore, the total number of nodes is the number of estimations plus one.[2] The probability of transition from node $i$ to $j$ is defined as:


Figure 3. Left: the GPS-tags of images in a collection of user-shared images. Right: the corresponding geo-density map $d$.

$$p(i,j) = \frac{e^{-\sigma \|\boldsymbol{g}_i - \boldsymbol{g}_j\|_2}}{\sum_{k=1}^{\lambda} e^{-\sigma \|\boldsymbol{g}_i - \boldsymbol{g}_k\|_2}}, \tag{2}$$

where $\|.\|_2$ denotes the $l_2$ norm. Equation 2 specifies the transition probability between two nodes according to their GPS-distance. It captures the common sense that the closer the nodes, the more consistent they are, and the higher the transition probability is. We set the insensitive parameter $\sigma$ to $0.05$ to reduce the transition probability between the nodes which are inconsistent by more than $60$ meters to less than $5\%$. The denominator normalizes the summation of the transition probabilities departing from each node to one.

### 2.2.1 Incorporating the Geo-density of images

As discussed in section 1, the user-shared images typically show a severely non-uniform geo-distribution; this characteristic can act as a bias and result in a reduction in the accuracy of tag-refinement. To better understand this, consider the case where there exists a popular and unpopular photography spots in the vicinity of each other. When performing image matching between the query and the dataset, more images from the popular spot are likely to be retrieved as more images from that location exist in the dataset. Consequently, there will be more triplet estimations coming from that spot and the final estimation of Random Walks will be leaning towards the location suggested by the images of that spot. To reduce the impact of this phenomena, we incorporate the density of the dataset in our Random Walks formulation. We define the initial score of the $n^{th}$ node in $\boldsymbol{N}$ as:

$$v(n) = \frac{\frac{1}{d_i d_j}}{\sum_a \sum_b \frac{1}{d_a d_b}}, \tag{3}$$

where $d_i$ and $d_j$ are the geo-densities of the two reference images which generated the $n^{th}$ triplet estimation. We define the geo-density, $d$, of an image as the number of other reference images within the radius $r$ of it. The geo-locations

[2] minus the number of triplets for which SfM failed to estimate $\boldsymbol{l}_\mathcal{I}$.

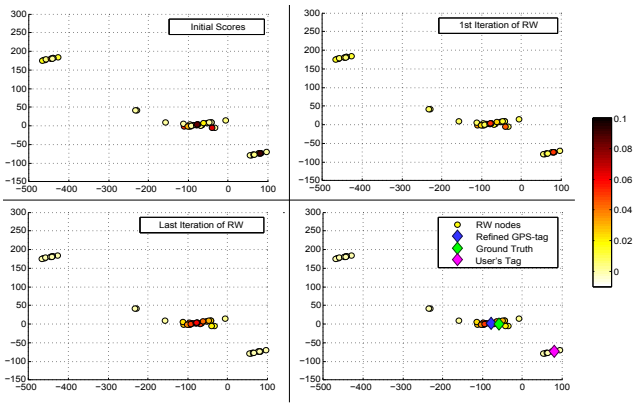


Figure 4. The process of Random Walks shown for a sample query in the East-North coordinate system. The initial scores based on geo-densities along with the relevance scores after the first and the last iterations, as well as the final estimation are illustrated.

of a collection of user images from Washington D.C. are illustrated in figure 3-left; the corresponding density map ($d$) is shown on the right. The denominator of equation 3 is intended to fulfill the Markov chain requirement of $\|\boldsymbol{v}\|_1 = 1$.

According to equation 3, the higher the densities of the parent images, the lower the initial value of the corresponding estimation. That is because a high density value implies many triplet estimations originated from the corresponding spot will be included in $\boldsymbol{N}$. Therefore, their overall impact needs to be reduced to restrain them from dominating the rest of the estimations. To better understand why equation (3) helps in realizing this goal, consider the simplified case where there are two spots $f$ and $q$ with $d_f$ and $d_q$ images in their vicinity. The number of triplets formed by taking one image from spot $f$ and another from spot $q$ is $d_f d_q$. Therefore, by defining the initial value of an estimation as the inverse of this number, we constrain the total estimations originated from different spots to have equal values irrespective of the number of references images in their neighborhood.

In our experiments, we set the value of $r$ and the initial score (before normalization) of the estimation corresponding to the initial GPS-tag to 5 meters and 1, respectively.

### 2.2.2 Adaptive Damping Factor

Having the node-to-node transition probabilities and the initial scores, Random Walks updates the relevance score of one node at each iteration based on the probability of transition from other nodes to it. Equation 4 is the formula of the basic Random Walks which performs this operation:

$$x_{(k+1)}(j) = \sum_{i=1}^{\lambda} \overbrace{\alpha}^{①} x_k(i)p(i,j) + \overbrace{(1-\alpha)}^{②} v(j), \quad (4)$$

where $x_k(i)$ is the relevance score of the $i^{th}$ node at the $k^{th}$ iteration. The argument of summation (left term) is the part which computes the probability of transition from other nodes to a particular one, and the right one is a damping term. The damping term was added to Random Walks to enable leveraging the prior knowledge about the relevance of nodes and to ensure *irreducibility* of the transition probabilities matrix which is a convergence condition for Random Walks [13, 7]. $\alpha$ is a mixture constant usually set to a value between 0.8 and 1. The summation of the terms ① and ② in equation 4 has to be 1 since the summation of the relevance scores at any iteration must be 1: $\sum_{i=1}^{\lambda} x_k(i) = 1$.

A careful look at equation 4 reveals an important characteristic of the basic Random Walk: the updated relevance scores always include $(1-\alpha)$ of the initial scores. That means $(1-\alpha)$ of the initial score of a node appears in the final relevance score regardless of its consistency with the other nodes. This is undesirable particularly when the nodes could include outliers with inaccuracies, as it essentially means a fixed portion of the input noise will always appear in the output. We propose a damping factor which adaptively changes according to the consistency of each node to the others. We accomplish this by making the damping term of a node a function of its relevance score at each iteration:

$$x_{(k+1)}(j) = \frac{1}{\eta}\Big( \sum_{i=1}^{\lambda} \overbrace{\Big(1-(1-\alpha)x_k(j)\Big)}^{①} x_k(i)p(i,j) + \overbrace{(1-\alpha)x_k(j)}^{②} v(j)\Big). \quad (5)$$

Equation 5 is equivalent to equation 4 with the difference that the damping term (②) is proportional to the relevance score of the node; therefore, the amount of contribution from the initial score of the node depends on its so-far consistency with the other nodes. Hence, an arbitrary noise in the input can be handled as the input error does not directly propagate in the output. In the context of our problem, we will show (in section 3.3) that Random Walks with the adaptive damping factor can handle GPS-location estimations ($\boldsymbol{g}_i$) with arbitrarily large errors while the basic Random Walks fails to do so.

Similar to equation 4, the term ① in equation 5 is equal to (1-②). The normalization constant $\eta$ given below makes the summation of relevance scores at all iterations 1:

$$\eta = \sum_{j=1}^{\lambda}\Big( \sum_{i=1}^{\lambda} \Big(1-(1-\alpha)x_k(j)\Big)x_k(i)p(i,j) + (1-\alpha)x_k(j)v(j)\Big). \quad (6)$$

The matrix form of Random Walks with the adaptive damping factor (i.e. equation 5) can be derived as:

$$\boldsymbol{x}_{(k+1)} = \frac{1}{\eta}(\boldsymbol{x}_k \boldsymbol{\Gamma} \boldsymbol{P} + \boldsymbol{v}(\boldsymbol{I} - \boldsymbol{\Gamma})), \quad (7)$$

where

$$\boldsymbol{\Gamma} = diag(1 - (1 - \alpha)\boldsymbol{x}_k). \quad (8)$$

$\boldsymbol{x}_{(k)}$ and $\boldsymbol{v}$ are $1 \times \lambda$ dimensional vectors of the relevance scores at the iteration $k$ and their initial scores respectively. $\boldsymbol{P}$ is a $\lambda \times \lambda$ matrix which has the pairwise transition probabilities as defined in equation 2. $diag(.)$ is an operator which generates a diagonal matrix where the elements on the main diagonal are the elements of the argument vector and the rest of the elements are set to zero. Also, the simpler matrix form of the normalization constant, $\eta$, can be written as $\eta = \|\boldsymbol{x}_k\boldsymbol{\Gamma}\boldsymbol{P} + \boldsymbol{v}(\boldsymbol{I} - \boldsymbol{\Gamma})\|_1$.

Notice the similarity between equation 7 and the matrix form of basic Random Walks: $\boldsymbol{x}_{k+1} = \alpha\boldsymbol{x}_k\boldsymbol{P} + (1 - \alpha)\boldsymbol{v}$. The main difference is that the damping factor matrix $\boldsymbol{\Gamma}$ is adaptively changing at each iteration instead of being fixed.

The relevance scores are iteratively computed until they converge to the final values $\boldsymbol{x}_\pi$, commonly termed as "stationary probability". Therefore, the vector $\boldsymbol{x}_\pi$ includes the final relevance scores of all of the GPS-location estimations.

#### 2.2.3 Final Tag Estimation using the Relevance Scores

The estimations which are severely affected by noise are expected to have $\approx 0$ final relevance scores, and the other estimations gain scores based on their agreement with the other nodes. Thus, we compute the refined GPS-location of the query, $\mathcal{I}$, utilizing a weighted mean using the scores $\boldsymbol{x}_\pi$:

$$\hat{\boldsymbol{g}} = \sum_{i=1}^{\lambda} \boldsymbol{g}_i x_\pi(i), \quad (9)$$

where $\hat{\boldsymbol{g}}$ is the refined GPS-location. The process of Random Walks is illustrated in figure 4 where the initial scores (based on the geo-densities) and the relevance scores after the first and the last iterations are demonstrated. The refined GPS-location along with the initial location and the ground truth are shown as well. Notice that the estimations which are far from the correct location are successfully identified by the Random Walks as they have a low relevance score.

## 3. Experimental Results

We performed our evaluations on a mixed dataset of 18,075 GPS-tagged user-shared images from the cities of Pittsburgh, PA; Palo Alto, CA and Washington, DC. The images were downloaded from Panoramio, Flickr and Picasa and were all captured and GPS-tagged by users.

### 3.1. Statistical Properties of Error in User Tags

Existence of inaccuracies in the user-shared GPS-tags has been acknowledged in several papers [20, 12, 5], and a few previous publications reported statistical properties of error of geo-tags acquired from GPS devices or by assuming the output of their methods to be the ground truth [3, 14]. In order to provide a formal *large scale* statistical study of the amount of noise in user-shared tags, we manually verified the accuracy of the GPS-tags of 8,127 images captured in Pittsburgh. We found that, depending on the resource website, typically about $10.2\%$ to $30\%$ of the user shared images have inaccurate tags (Panoramio showed the least error). By "inaccurate", we mean an image whose GPS-tag has an error more than 30 meters which is the nominal accuracy of the commercial GPS devices. Figure 5-left shows the distribution of the error of the inaccurate GPS-tags. It shows a near Gaussian distribution with the mean and standard deviation of 425.6 and 228.0 meters. We focus on the errors less than $1km$ in figure 5-left, as the larger values seem to significantly correlate with the layout of the city and consequently fail to generalize.

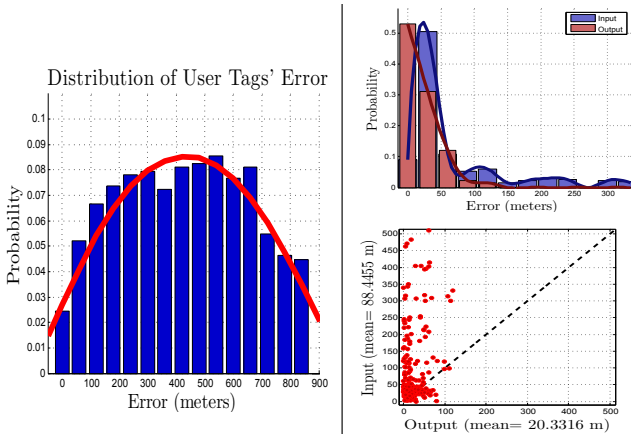


Figure 5. Left: the distribution of the error in the user-specified GPS-tags of 8127 images with inaccurate tags. It shows a near-Gaussian distribution with the mean and standard deviation of 425.6 and 228.0 meters, respectively. Right: the results of tag refinement when no additional contamination is added.

### 3.2. Tag Refinement Results

As the test set, we selected a subset of 500 images from the dataset and accurately annotated their ground truth location (with an error $< 10$ meters). We refined the GPS-tags of the test set images using the rest of the images in the dataset and compared the refined location against the ground truth to find the refinement error. The query images which returned less than 5 matches from the rest of the dataset and the ones for which SfM failed to generate at least 9 estimations were removed from the test as they typically correspond to either isolated images or panoramic/edited ones. Figure 5-right shows the refinement results when no additional contamination was added to the dataset; the input error is the inaccuracy of the user tags. The distributions of error in the input and output are shown in the top which show the considerable improvement made by our method. The error scatter plot in which each point represents one query image is illustrated in the bottom.

In another experiment, in order to investigate the per-

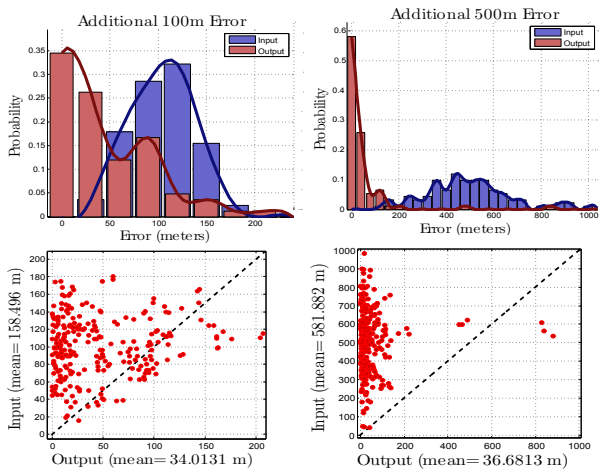


Figure 6. The refinement results for various contaminations with the mean values of 100 and 500 meters. The distributions and scatter plots are shown on the top and bottom rows, respectively.

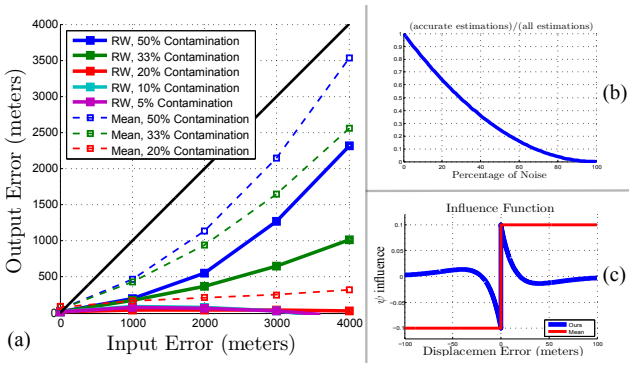


Figure 7. (a): The performance of the proposed refinement method for additional contaminations with various mean and percentage values. (b): the ratio of the accurate estimations over the total number of estimations with respect to the percentage of noise. (c): the Influence Function of our method and the baseline (mean).

formance of our method under various scenarios, we added random Gaussian noise with the mean values of 100, 500, 1000, 2000, 3000, and 4000 meters to 5, 10, 20, 33 and 50 percents of the 18075 images in our dataset; the standard deviation was set to 0.5 of the mean to replicate the user tags' error (see section 3.1). Note that these errors are on top of the already existing noise in the user specified tags in our dataset. Therefore, the additional contamination determines the lower bound of noise since the exact amount of error in the dataset is unknown as the ground truth location of all of the 18075 images are not known. We also made sure that in this experiment, the query images were among the ones with contaminated GPS-tags to ensure the evaluation is fair and challenging enough. Figure 6 shows the results of this experiment for the additional contamination percentage of 20% with the means values of 100 and 500 meters. As apparent in both of the distributions and scatter plots, our method significantly reduces the amount of error.

Figure 7 (a) shows the mean of the output error for various amounts of contaminations in the input tags. Two observations can be made in the figure 7 (a): first, for the contamination percentages less than 30%, our method almost completely eliminates the error regardless of the mean of the contamination in the input. That is why the error curves for the contamination percentages of 5, 10 and 20 are almost flat. This shows the high empirical *Breakdown Point* [6] (defined as the resistance of a method against the proportion of inaccurate observations in the data in robust statistics) of our estimator. However, when the percentage of error increases to beyond 33% and 50%, the output error becomes noticeable, yet it is considerably less than the error in the input. This observation is consistent with the bases of our method as the ratio of the number of estimations not affected by noise over all of the estimations is $\binom{n-nq}{2}/\binom{n}{2}$ where $q$ and $n$ are the percentage of noisy tags and total number of images in the dataset, respectively. This ratio is shown in figure 7 (b); as apparent in the plot, when the percentage of contamination goes beyond 30% and 50%, the percentage of estimations affected by noise increases to over 50% and 75%, respectively, and thus, it becomes excessively difficult to discover the inliers. Also, figure 7 (b) justifies why we used images *triplets* for generating the estimations and not quadruplets or quintuplets; the ratio of 7 (b) would drop with a sharper slope if more images were used for generating an estimation which is undesirable.

Additionally, the *Influence Function*, which is a measure of the dependence of an estimator on the displacement error of one observation [6], of our method has the favorable *descending* shape (see figure 7 (c)). That means a sample with an arbitrarily large error have a small impact on our final estimation whereas it has an unbounded effect on the results of non-robust methods such as mean.

The dashed curves in 7 (a) illustrates the results of using the average of the triplet estimations as the refined GPS location (i.e. bypassing Random Walks and using uniform mean instead). Unlike the Random Walks results, the output error curves of all contamination percentages are always monotonically increasing, which shows the input error is propagated to the output. Similalrly, table 1 compares the refinement results when Random Walks were replaced by alternative mode seeking methods. Unlike Random Walks, incorporation of the geo-density is not straightforward when Mean Shift or RANSAC is employed. Moreover, employing Means Shift or RANSAC requires specifying a kernel function (we used Gaussian) or a distribution for the inlier nodes (we used a uniform disk), respectively; on the contrary, Random Walks only need a *pairwise* transition function. Also, Random Walks is superior to inference techniques, such as Loopy Belief Propagation, that have convergence issues in fully connected graphs [15].

In general, inaccurate estimations by SfM (which usually causes finer errors), or too few or no uncontaminated triplet

| | | Input Error (m) | | | |
|---|---|---|---|---|---|
| | | 100m | 300m | 3000m | 5000m |
| Method | Mean | 79.4 | 95.9 | 244.3 | 459.3 |
| | RANSAC, BW=20m | 38.2 | 48.3 | 93.5 | 153.8 |
| | RANSAC, BW=150m | 73.2 | 77.7 | 82.9 | 83.1 |
| | Mean Shift, BW=20m | 34.1 | 45.2 | 81.2 | 130.7 |
| | Mean Shift, BW=150m | 61.4 | 63.7 | 67.2 | 67.4 |
| | Ours (Random Walk) | **34.0** | **34.8** | **38.4** | **43.6** |

Table 1. Comparison of tag refinement results of various methods. The percentage of additional contamination is 20%.

estimations (rare but leads to a large error) are the two main reasons behind the cases which our method failed to refine.

### 3.3. Evaluation of the Adaptive Damping Factor

Figure 8 shows the evaluation of the proposed adaptive damping factor compared to the conventional damping. On the left, the distribution and scatter plot of the error in the input and output for $\alpha = 0.8$ and the mean contamination of 3,000 meters in 20% of the images is shown.

The curves on the right illustrate the mean error in the output of Random Walks with the constant (i.e. equation 4) and adaptive damping factors (i.e. equations 5, 7) for various values of $\alpha$. The value of $\alpha$ determines the contribution of the initial scores in the final relevance scores; the green curve signifies the error of the constant damping factor increases with increasing $\alpha$ while the error of adaptive damping factor remains nearly flat. That shows adaptive damping successfully prevents the noise in the input from being directly propagated in the output.

**Incorporating the Geo-Density:** Table 2 provides the performance of utilizing the geo-density (equation 3) as the initial score compared to using uniform initial scores. As apparent in the table, for almost all values of $\alpha$ and input contaminations, the geo-density yields better output error compared to the uniform scores (except for the case of 300 meters where the performance of both methods are $\leq 1$ meter different.); the improvement made by density handling is more noticeable in large errors. The red numbers show the best performance for each value of contamination. The $\alpha$ values between 0.80 and 0.90 typically yield the best results where lower values work better for lower errors and vice versa; that's because in lower contaminations, the initial scores are more accurate and consequently increasing their influence boosts the performance. Since we can make no prior assumptions about the mean of error, in all of our experiments, we set $\alpha$ to 0.90 which is found to work satisfactorily for both small and large errors. Bear in mind that our query set is a subsample of our large dataset; the improvement made by density handling would be even more significant if the test set showed a distribution relatively different from the rest of the dataset.

### 3.4. Empirical Convergence and Efficiency

On an 8 core 2.4 GHz machine running MATLAB, our framework, excluding performing SfM on triplets, runs in

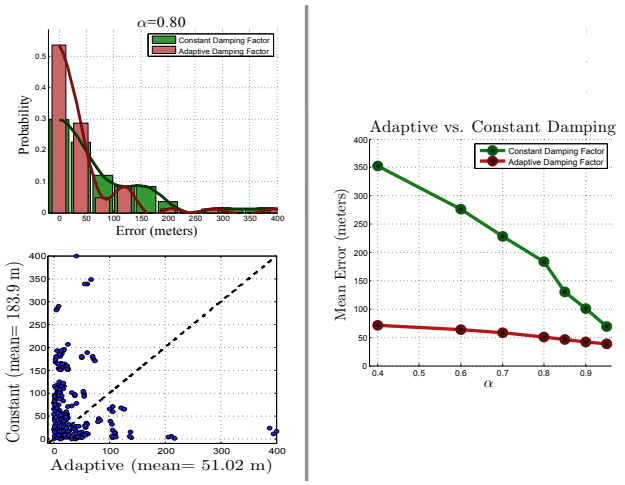


Figure 8. Evaluation of the adaptive damping factor. The distributions and scatter plot comparing constant and adaptive damping are shown on the left. The curves on the right compare the adaptive and constant damping factor for different values of $\alpha$.

0.04 seconds per image. The main reasons behind the high efficiency of our method is the fast convergence characteristic of Random Walks and forming the graph of estimations in a local manner (i.e. separate for each query).

In order to empirically analyze the convergence of the proposed formulation for Random Walks with adaptive damping, we randomly generated a set of 1 million graphs with 100 to $10k$ nodes and various distributions (Gaussian, GM, uniform and their mixtures) for unary and binary terms. Random Walks converged in 100% of the instances in the average time of 0.0070 seconds with the standard deviation of 0.0067. The mean number of required iterations and its standard deviation were 18.43 and 1.65, respectively.

### 3.5. Refinement using Image Geo-tags (no SfM)

So far, we generated the estimations, $\boldsymbol{g}_.$, using SfM while one could use the GPS-tags of the images matched to $\mathcal{I}$ as the estimations for its location. However, that would imply we assume the dataset is dense enough to the point that there exist similar images in the dataset with camera locations very close to the one of $\mathcal{I}$. Otherwise, performing the tag refinement using the matched images' GPS-tags would achieve a limited success whereas SfM wouldn't have the requirement of having images with near-identical camera locations to $\mathcal{I}$. In order to empirically investigate this, we performed an experiment to compare employing SfM vs. using the GPS-tags of the matched images as the estimations, $\boldsymbol{g}_.$, in our framework. The scatter plot in figure 9-left illustrates the results for the contamination with the mean value and percentage of 3,000 meters and 20%. As expected, using SfM improves the overall accuracy (by 9.2m).

However, bypassing the SfM has some advantages such as substantially lowering the time complexity or increasing the number of estimations due to the typical high failure rate

| | | Input Error (m) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 100m | | 300m | | 3000m | | 5000m | |
| | | Den. | Uni. | Den. | Uni. | Den. | Uni. | Den. | Uni. |
| $\alpha$ | .95 | **35.1** | 35.2 | 35.8 | **35.6** | **38.6** | 38.8 | **39.4** | 43.1 |
| | .90 | **34.0** | 35.2 | **34.8** | 35.2 | **38.4** | 42.1 | **43.6** | 52.1 |
| | .80 | **33.8** | 34.0 | 36.0 | **35.2** | **51.0** | 52.1 | **51.6** | 73.5 |
| | .60 | **34.7** | 34.9 | 37.8 | **36.8** | **63.45** | 65.1 | **67.4** | 101 |
| | .40 | **36.4** | 36.5 | 38.8 | **37.8** | **67.5** | 72.4 | **81.4** | 118 |

Table 2. Evaluation of the density handling method for various values of $\alpha$ and contamination means. Den. and Uni. represent setting the initial scores based on the geo-density or uniform scoring. The bold numbers show the best performance for a particular value of $\alpha$ and contamination means. The red ones show the best overall performance for a particular contamination mean.

of SfM, which could make this approach desirable in certain scenarios, e.g. when a fine error in the results is acceptable.

### 3.6. Tag refinement vs. Localization

We used the initial GPS-tag of the query image in our framework in order to refine the tag. However, our approach could be viewed as an image geo-localization method if the initial geo-tag was not leveraged. The scatter plot of figure 9-right shows the results of an experiment on the overall impact of the initial GPS-tag in the final estimated GPS-tag (i.e. tag-refinement vs. localization mode). The mean and percentage of contamination are 3,000 meters and 20% respectively. We made sure the initial GPS-tags are *not* contaminated in this experiment as we are investigating their impact. As one would expect, utilizing the initial GPS-tag leads to better results as it is an additional cue to the right location of the query; this additional estimation could become essential particularly for the images for which few matches were retrieved from the dataset or few estimations were generated using SfM.

However, the mean of the output error in localization mode is only 82.2 meters while 20% of the dataset images have the mean contamination of 3,000 meters. This signifies our method preserves its robustness trait in the localization mode as well and can be used for geo-localization purpose *when the reference dataset includes unknown inaccuracies*. This is especially important as the majority of existing image localization methods [12, 5, 8] do not have a particular mechanism for dealing with noisy tags in their reference data (i.e. the noise in input will directly affect the output).

## 4. Conclusion

In this paper, we proposed, to the best of our knowledge, the first method for refinement of the GPS-tags of crowdsourced images. Given a large dataset of GPS-tagged images with an unknown subset with inaccurate tags, we discovered the contaminated subset and adjusted the GPS-tags therein to the correct locations. This was done by performing image matching, generating location estimations using SfM on triplets of matching images, performing Random Walks to identify the subset with the maximal agreement, and finally using a weighed average of the consistent estimations. We proposed an adaptive damping factor for Random Walks and incorporated the geo-density of images to minimize the bias it induces in the results. The experiments evaluated various aspects of the method and showed it constantly performs robustly across different scenarios.

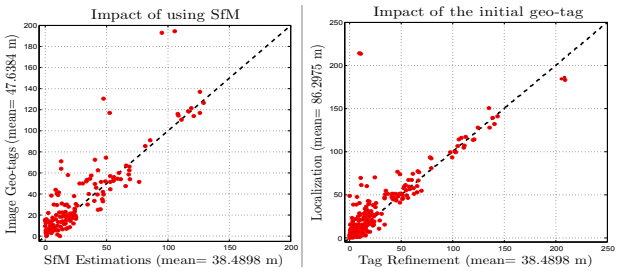


Figure 9. Left: The scatter plot showing the effect of using SfM for generating the location estimations as compared to directly using the GSP-tags of the matched images as the estimation. Right: The impact of the initial GPS-tag in the overall results (i.e. localization vs. tag-refinement mode).

## References


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

[2] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011. 2

[3] D. Chen et al. City-scale landmark identification on mobile devices. In *CVPR*, 2011. 5

[4] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *CVPR*, 2009. 2

[5] J. Hays and A. Efros. IM2GPS: estimating geographic information from a single image. In *CVPR*, 2008. 1, 2, 5, 8

[6] P. J. Huber. *Robust statistics*. Springer, 2011. 1, 6

[7] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. 4

[8] T.-Y. Lin, S. Belongie, and J. Hays. Cross-view image geolocalization. In *CVPR*, 2013. 1, 2, 8

[9] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Image retagging. In *ACM Multimedia*, 2010. 2

[10] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *ACM Multimedia*, 2009. 2

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2

[12] A. R. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 1, 2, 5, 8

[13] F. Spitzer. *The Theory of Stochastic Processes*. Springer, 2001. 4

[14] C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and scalable large scale image reconstruction. In *CVPR*, 2010. 5

[15] Y. Weiss. Correctness of local probability propagation in graphical models with loop. In *Neural Computation*, 2000. 6

[16] C. Wu. Visualsfm: A visual structure from motion system. *http://ccwu.me/vsfm/*, 2011. 2

[17] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, 2011. 2

[18] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, 2010. 2

[19] G. Zhu, S. Yan, and Y. Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *ACM Multimedia*, 2010. 2

[20] D. Zielstra and H. H. Hochmair. Positional accuracy analysis of flickr and panoramio images for selected world regions. *Journal of Spatial Science*, 2013. 1, 5