Estimating Landmark Locations from Geo-Referenced Photographs

Henrik Kretzschmar Cyrill Stachniss Christian Plagemann Wolfram Burgard

***Abstract*— The problem of estimating the positions of landmarks using a mobile robot equipped with a camera has intensively been studied in the past. In this paper, we consider a variant of this problem in which the robot should estimate the locations of observed landmarks based on a sparse set of geo-referenced images for which no heading information is available. Sources for such kind of data are image portals such as Flickr or Google Image Search. We formulate the problem of estimating the landmark locations as an optimization problem and show that it is possible to accurately localize the landmarks in real world settings.**

I. Introduction

Popular Internet resources such as Flickr or Google Image Search offer a large amount of real world imagery. Many of these images contain geo-references, i.e., the locations where the photographs have been taken in longitude and latitude coordinates as well as manual annotations such as marked image regions and a tag word like “cathedral”. Currently, Flickr offers millions of tagged images, a trend which is likely to continue given the growing popularity of mobile devices, GPS receivers and specialized integrated systems. The question of how this large amount of freely available data can be used to infer quantitative knowledge about the world was our main motivation for this work. On a comparably small spatial scale, systems such as Microsoft’s Photo Tourism [13] process sets of images of the same location to yield a dense 3D model of the local environment, which is capable of producing artificial views and virtual fly-throughs. In this paper, we deal with the problem of localizing a discrete set of distinct landmarks on a larger spatial scale, like a town or a campus environment. Concretely, our task can be formulated as follows. Given a set of geo-referenced photographs of an environment annotated with labels for distinct landmarks, how can we recover the locations of the landmarks in the world? Aside from the noisy geo-reference coordinates and inaccurate label placements, the main difficulty lies in the missing information about the camera headings.

A robot that is able to utilize a so far unused source of information offers new ways for building models of places it has not observed directly. It furthermore allows a robot to also refine or annotate exiting models. Consider, for example, a mobile tour guide robot deployed to a city center or to an archaeological site. Given the localized landmarks and the corresponding imagery, the system could offer a large range of location-dependent information without requiring a human expert to collect and formalize this knowledge.

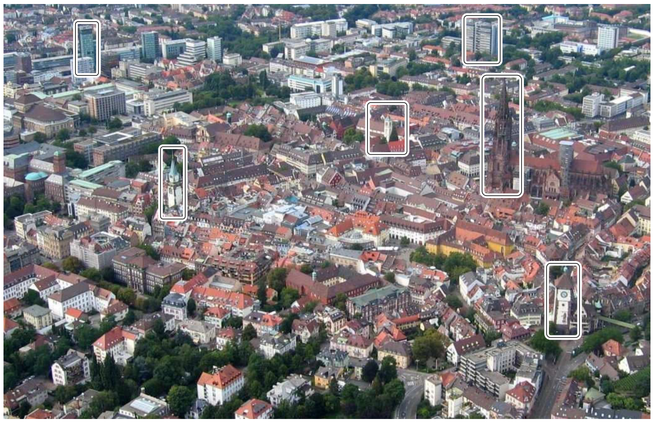

Fig. 1. The goal of the work presented in this paper is to estimate the locations of the set of distinct buildings (here enclosed by rectangles) using geo-referenced photographs taken while walking through the city.

In this paper, we consider the problem of estimating the positions of landmarks given a set of geo-referenced photographs. The longitude and latitude information of the locations from which the photos have been taken are assumed to be known approximatively by means of a standard consumer GPS device. By combining this data with labeled regions in the photos referring to objects, such as buildings, our approach is able to localize these buildings and to determine the direction in which the photo has been taken. In contrast to bearing-only SLAM, our approach does not require an image stream from a camera. We furthermore assume to have no knowledge about the orientation of the camera at any point in time. We address this problem by formulating it as an optimization problem. As we show in the experiments, we are able to accurately localize the labeled buildings based on photos taken in an urban environment.

Figure 1 depicts the downtown area of Freiburg including several distinct landmarks enclosed by rectangles. The goal of this work is to estimate the positions of such landmarks based on photographs taken while walking or moving through the city center.

The remainder of this paper is organized as follows. After discussing related work, we present a mathematical formulation of our problem and derive an objective function that needs to be minimized to solve the estimation problem. In Section IV, we then explain two optimization procedures that are used throughout this work. Finally, we present real world as well as simulation experiments that illustrate the performance of our method.

The authors are with the University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany. {kretzsch, stachnis, plagem, burgard}@informatik.uni-freiburg.de

II. Related Work

The problem of estimating the poses of landmarks based on observations has been intensively studied in the past. Several researchers studied the landmark-based simultaneous localization and mapping (SLAM) problem [1], [9], [8], [12], [4], [7]. In the literature, one distinguishes between SLAM approaches that are designed to operate on proximity sensors and those operating on bearing-only (typically vision) sensors. For example, Davison *et al.* [2] presented a vision-based 6 DoF SLAM system that extracts features from a monocular camera and creates a sparse map of high-quality stable features. The locations of the features are tracked by applying an EKF. Lemaire *et al.* [6] focused on the problem of how to initialize landmarks in the context of EKFs and bearing-only observations.

Approaches to bearing-only SLAM, however, consider that the robot constantly perceives the environment with its camera. As a result, a continuous stream of images and thus feature observations is provided so that the landmarks can be tracked over time. Furthermore, the robot is often assumed to roughly know the relative change in its orientation while moving from odometry. These two assumptions are not made in the work presented in this paper.

Recently, related approaches to this paper have been presented. The most prominent one is probably Microsoft's Photo Tourism [13]. This approach considers a set of images of the same place and generates a dense 3D model of the local environment. It is capable of producing artificial views and virtual fly-throughs. In contrast to this, we deal with the problem of localizing a discrete set of distinct landmarks on a larger spatial scale, for example in a town or a campus environment.

The MIT City Scanning Project [14] addresses the problem of building models from city environments with a mobile robot. This approach focuses on the textured 3D reconstruction of buildings in the environment. In the so-called "4D-Cities" project, Dellaert and colleagues [3], [11] address the problem of building spatial-temporal models of cities. They use current and historical photographs to reconstruct city scenes at different points in time. The temporal ordering can be inferred from the images by formulating it as a constraint satisfaction problem. This allows for time travels in cities.

III. Landmark and Camera Pose Estimation

We consider the problem of estimating the camera poses and the locations of observed landmarks given a labeled set of camera images. In the remainder of this paper, we use the following notation.

A. Problem Formulation

Let $P_i = (X_{P_i}, Y_{P_i}, \theta_{P_i})^T$ be the position and orientation of the camera when recording image i. Let $L_j = (X_{L_j}, Y_{L_j})^T$ be the position of landmark j in the Cartesian space. Each observation of landmark j seen in image i recorded at position P_i is a horizontal angle α_{ij} that describes the location of the landmark relative to the optical axis of the camera when recording the image.

In this paper, we assume that the 2D locations (X_{P_i}, Y_{P_i}) at which the camera images have been recorded are approximately known since images are supposed to be geo-reference. This information can be obtained from a low cost consumer GPS device. However, the orientation information θ_{P_i} is unknown. Given the observations α_{ij}, the goal is to estimate the landmark locations L_j as well as the orientations of the cameras θ_i. In addition to that, we improve the estimate of the locations of the cameras as delivered by the GPS device.

We consider the landmarks as uniquely identifiable. The problem of extracting appropriate features to identify them is not the focus of this work. We furthermore assume that the roll angle of the camera during image recording is zero. By means of low cost attitude sensors used together with the cameras, images can easily be corrected by a simple rotation. Other information such as the focal length of the lens while taking the image can be obtained from the EXIF tags stored in the images.

Note that we assume to have no direct information about the orientations of the cameras. As a result, images that only contain a single landmark have no influence on the estimate because the landmark can be located anywhere. This makes our approach different from typical bearing-only SLAM techniques which, in general, assume to have an estimate about the orientation of the camera that is typically computed from an image stream recorded by the camera or by using odometry information.

B. Objective Function

In our approach, we solve the described location estimation problem by means of optimization. To apply an optimization procedure, one needs to define an objective function. In our scenario, the objective function can be defined as the error between the obtained observations and the estimated positions of the landmarks.

Let the variables indicated by $\hat{\cdot}$ refer to the estimated quantities. In case the estimated camera and landmark positions are consistent with the observations, we obtain

$$\tan(\hat{\theta}_{P_i} + \alpha_{ij}) \quad = \quad \frac{\hat{Y}_{L_j} - \hat{Y}_{P_i}}{\hat{X}_{L_j} - \hat{X}_{P_i}}. \tag{1}$$

In practice, we want to minimize the position error of the landmarks. Our observations, however, only provide bearing information. Thus, the error to be minimized can be specified by the difference between the estimated landmark location ($\hat{L}_j$) and the line of sight starting from the camera position ($\hat{P}_i$) in the direction of the observed landmark ($\hat{\theta}_{P_i} + \alpha_{ij}$) (see Figure 2 for an illustration).

One can easily compute the arc length E_{ij} between the estimated landmark location ($\hat{L}_j$) and the line of sight resulting from the observation. It can be computed as

$$E_{ij} \quad = \quad e_{ij} r_{ij}, \tag{2}$$

where

$$e_{ij} \quad = \quad \operatorname{atan2}(\hat{Y}_{L_j} - \hat{Y}_{P_i}, \hat{X}_{L_j} - \hat{Y}_{P_i}) - \hat{\theta}_{P_i} - \alpha_{ij} \tag{3}$$

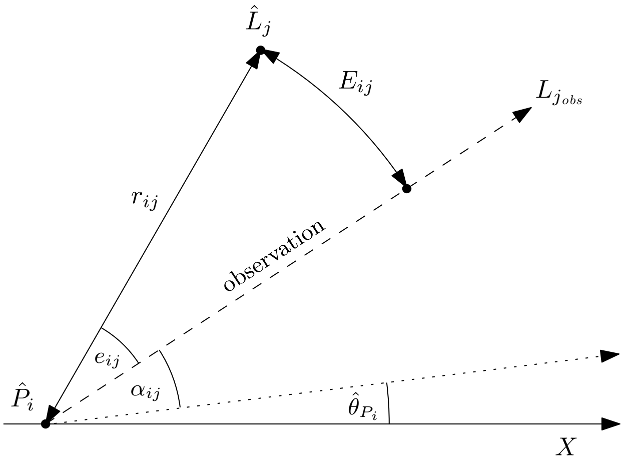

Fig. 2. Illustration on how to compute the estimated error E_{ij}. $\hat{P}_i$ refers to the pose of the camera looking in the direction indicated by the dotted line. The dashed line is the line of sight on which the landmark is located given the observation. $\hat{L}_j$ is the estimated landmark location and e_{ij} is the angular error between the estimated and the observed landmark. r_{ij} is the Euclidean distance between the camera and the estimated landmark.

is the angular error in radians between the estimated and the observed landmark computed from Eq. (1) and

$$r_{ij} = \sqrt{(\hat{X}_{P_i} - \hat{X}_{L_j})^2 + (\hat{Y}_{P_i} - \hat{Y}_{L_j})^2} \qquad (4)$$

is the Euclidean distance between camera i and estimated landmark j. The function $\mathrm{atan2}(\Delta Y, \Delta X)$ refers to $\arctan(\Delta Y / \Delta X)$ but explicitly considers the four quadrants.

In our scenario, we assume that the locations of the cameras are measured with a consumer GPS device. Thus, these positions can be regarded as globally correct with a bounded noise term. As a result, the optimization approach should be allowed to *locally* modify them if this reduces the overall error. Therefore, we add a penalty term $f(x)$ that allows for local corrections only. This function is a differentiable barrier function that goes to infinity as x approaches the maximum assumed GPS error, but is near zero close to the measured location. Such an approach is frequently applied to cope with GPS errors [15]. As a result, the objective function E turns into

$$E \;=\; \sum_{i,j} {E_{ij}}^2 + \sum_i f(||\hat{P}_i - \tilde{P}_i||), \qquad (5)$$

where $\tilde{P}_i$ refers to the locations of the cameras as provided by the GPS observation.

C. Gradient for Optimization

Most optimization techniques either directly rely on gradient information or can be sped up significantly by incorporating knowledge about the gradient of the objective function. In our model, the gradient of the error function E as stated in Eq. (5) is given by

$$\nabla E \;=\; \sum_{i,j} \nabla {E_{ij}}^2 + \sum_i \nabla f(||\hat{P}_i - \tilde{P}_i||)\ . \qquad (6)$$

The gradient consists of the partial derivatives with respect to the individual variables we want to optimize, namely $\hat{\theta}_{P_i}$, $\hat{X}_{L_j}$, $\hat{Y}_{L_j}$, $\hat{X}_{P_i}$, and $\hat{Y}_{P_i}$. By applying a series of mathematical derivations, we obtain

$$\frac{\partial E}{\partial \hat{\theta}_{P_i}} \;=\; -2 \sum_j e_{ij} {r_{ij}}^2 \qquad (7)$$

$$\frac{\partial E}{\partial \hat{X}_{L_j}} \;=\; 2 \sum_i {e_{ij}}^2 (\hat{X}_{L_j} - \hat{X}_{P_i}) - e_{ij} (\hat{Y}_{L_j} - \hat{Y}_{P_i}) \qquad (8)$$

$$\frac{\partial E}{\partial \hat{Y}_{L_j}} \;=\; 2 \sum_i e_{ij} (\hat{X}_{L_j} - \hat{X}_{P_i}) + {e_{ij}}^2 (\hat{Y}_{L_j} - \hat{Y}_{P_i}) \qquad (9)$$

$$\frac{\partial E}{\partial \hat{X}_{P_i}} \;=\; 2 \sum_j e_{ij} (\hat{Y}_{L_j} - \hat{Y}_{P_i}) + {e_{ij}}^2 (\hat{X}_{P_i} - \hat{X}_{L_j}) + \frac{\partial}{\partial \hat{X}_{P_i}} f(||\hat{P}_i - \tilde{P}_i||) \qquad (10)$$

$$\frac{\partial E}{\partial \hat{Y}_{P_i}} \;=\; 2 \sum_j {e_{ij}}^2 (\hat{Y}_{P_i} - \hat{Y}_{L_j}) - e_{ij} (\hat{X}_{L_j} - \hat{X}_{P_i}) + \frac{\partial}{\partial \hat{Y}_{P_i}} f(||\hat{P}_i - \tilde{P}_i||). \qquad (11)$$

Note that the penalty term $f(||\hat{P}_i - \tilde{P}_i||)$ only affects the partial derivatives with respect to the estimated camera locations $\hat{X}_{P_i}$ and $\hat{Y}_{P_i}$.

We have specified the objective function for our problem as well as its partial derivatives. After randomly sampling an initial guess, we can now apply gradient-based optimization techniques to compute a solution.

IV. Optimization

Optimization refers to the task of systematically choosing the values of variables to minimize or maximize an objective function E. For our problem, the objective function is given in Eq. (5). This section briefly introduces gradient descent and RPROP from a general point of view. Both methods are applied in this work.

A. Gradient Descent

Gradient descent is a frequently used iterative optimization technique. Starting from an initial parameter setting x_0, it alternates between (a) computing the gradient ∇E of the objective function E w.r.t. its parameters and (b) changing the parameter vector in the direction opposing the gradient. More formally, we set

$$x_{n+1} \;=\; x_n - \varepsilon \cdot \nabla E\ , \qquad (12)$$

where ε is a scale factor that specifies the change in the variables according to the gradient. This update-rule is iterated until convergence or until a maximum number of iterations has been carried out.

Standard gradient descent is easy to implement, provided that the gradient of the objective function E is known. The scale factor ε, however, is hard to choose in practice and there is no general rule on how to determine it. If ε is chosen too small, the resulting small steps cause convergence to be slow. Too big values for ε, however, can lead to oscillation or even divergence. In short, standard gradient descent converges rather slowly and has no convergence guarantee in the general case.

B. RPROP

Resilient backpropagation (RPROP) [10] was originally proposed as a learning algorithm for artificial neural networks. The goal was to overcome the weaknesses of standard gradient descent outlined above. In contrast to standard gradient descent, RPROP neglects the absolute value of the derivative. Instead, it considers the changes of signs of the individual partial derivatives. The update rule of RPROP consists of two steps. First, the so-called update value Δ_n^k for each dimension k is computed in each iteration n as

$$\Delta_n^k = \begin{cases} \eta^+ \Delta_{n-1}^k & , \text{ if } \quad \frac{\partial E}{\partial x^k}^{n-1} \frac{\partial E}{\partial x^k}^{n} > 0 \\ \eta^- \Delta_{n-1}^k & , \text{ if } \quad \frac{\partial E}{\partial x^k}^{n-1} \frac{\partial E}{\partial x^k}^{n} < 0 \\ \Delta_{n-1}^k & , \text{ else,} \end{cases} \quad (13)$$

where $0 < \eta^- < 1 < \eta^+$ and $\frac{\partial E}{\partial x^k}^{n}$ is an abbreviation for $\frac{\partial E}{\partial x^k}(x_n^k)$. Based on this update value Δ_n^k, the update rule can be specified as

$$x_{n+1} = x_n + \begin{cases} -\Delta_n^k & , \text{ if } \quad \frac{\partial E}{\partial x^k}^{n} > 0 \\ +\Delta_n^k & , \text{ if } \quad \frac{\partial E}{\partial x^k}^{n} < 0 \\ 0 & , \text{ else.} \end{cases} \quad (14)$$

According to Eqs. (13) and (14), if the sign of a partial derivative changes with respect to the previous iteration, the step size is decreased by a constant factor η^-. If the sign does not change, the step size is increased by the factor η^+. The former is done in order to prevent the algorithm from jumping over local minima, whereas the latter is done to accelerate convergence in shallow regions.

Despite its comparably fast convergence, RPROP is rather easy to implement. Compared to gradient descent, it does not depend on a fixed scale factor ε which is hard to determine. RPROP adapts its scale factors automatically and thus leads to more robust and flexible optimization, which does not require manual parameter tuning. As we will show in the experimental section, RPROP is a suitable technique for solving our estimation problem and clearly outperforms gradient descent.

Note that other optimization approaches such as Levenberg-Marquardt or scaled conjugate gradient can be used as alternatives to gradient descent or RPROP. However, as we illustrate in the experimental evaluation, the conceptually simpler and easy to implement RPROP already leads to highly satisfactory results.

V. Experiments

We performed a series of simulation and real world experiments to test our method. Simulated experiments allow us to compare the solution of our algorithm versus the ground truth, whereas the real world experiments show that our technique is able to solve the addressed problem in realistic settings.

A. Real World Experiments

For data acquisition, we used a standard digital photo camera and a consumer GPS logger (XAiOX iTrackU SiRF III). We walked through the city of Freiburg and took a series of photos from different locations. We then manually labeled a set of buildings in the images to obtain the correspondences between the images. Note that the data association problem is not addressed in the paper. To get an estimate of the quality of our approach in real applications, we compared the estimated landmark locations to the ones obtained from satellite images. We furthermore mounted a compass to the camera in order to compare the estimated orientation of the camera to the angle indicated by the compass (analog, accurate up to $\sim 3\,\text{deg}$).

To determine the horizontal angle in which objects are observed, the camera needs to be calibrated. Such a calibration is a mapping between pixel coordinates and bearing angles and accounts for lens distortion and other camera specific parameters. To achieve the necessary calibration, one could either use appropriate databases for consumer cameras or accurately calibrate it using chessboard patterns [5].

We collected two datasets, one in Freiburg downtown and one on the campus of the computer science department of the University of Freiburg. In the city center, the landmarks were located in an area of approximately 1.5 km by 1 km (distance camera-landmarks: 180 m to 2.5 km) and on the campus in an area of approximately 320 m by 300 m (distance camera-landmarks: 10 m to 300 m). Whereas the left image in Figure 3 shows the Freiburg downtown area, the right image depicts the campus area including the estimated landmark locations as well as the true positions. Additionally, Figure 4 illustrates the absolute error for the individual landmarks based on the campus experiment. The true landmark locations were measured manually using high-resolution satellite images. To further analyze the robustness of our methods, we carried out 300 optimization runs in both real world experiments and randomly initialized the landmark locations and camera heading angles. In all runs, our approach converged towards the same solution which illustrates its robustness.

We also compared the estimated camera orientations to the ones measured with a compass. It turned out that all estimated orientations differ from the ones measured with the compass by less than 3 deg, which is approximately the measurement accuracy of our analog compass. In the future, a digital compass could be used to automate this task.

B. Simulated Experiments

Simulated experiments allow us to analyze our method in a controlled environment. We examine the evolution of the real error during the process of optimization. We also provide a comparison of the performance of our approach using RPROP and the same approach applying standard gradient descent.

As explained above, we assume that the locations where the images were taken are roughly known from a GPS device. Like all measurements, GPS observations are distorted by noise. Hence, in our simulation experiments, we simulate the noise by sampling from a Gaussian with a standard deviation $\sigma = 10\,\text{m}$. Furthermore, we assume a horizontal opening angle of the cameras of 65 deg.

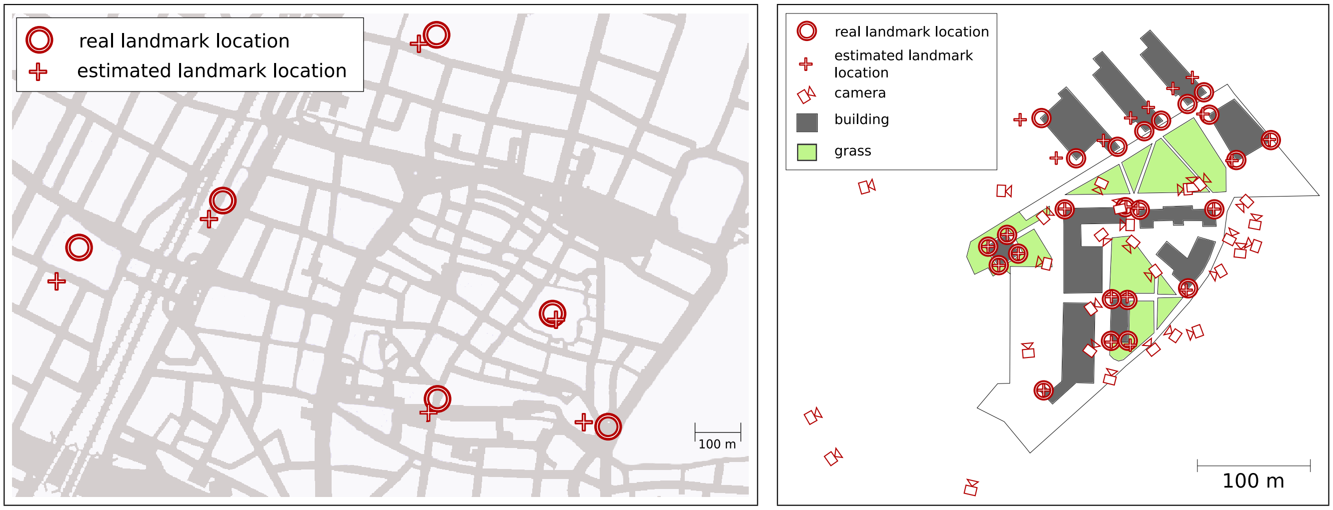

Fig. 3. True and estimated landmark locations in Freiburg downtown overlayed on a street map (left) as well as for the Freiburg campus experiment overlayed on a building plan (right). Note that due to copyright reasons, we do not visualize the results using the original satellite images.

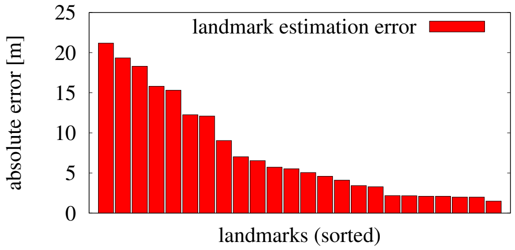

Fig. 4. Individual estimation errors of the different landmarks in the campus experiment. The landmarks are sorted by their errors for better visibility.

In the first simulation run, we used six images to estimate the positions of six partially visible landmarks. We applied our algorithm using RPROP and gradient descent. We applied gradient descent in two settings, using $\varepsilon = 0.01$ and $\varepsilon = 0.001$. Figure 5 shows the evolution of the real error versus the number of iterations in a typical run of the experiment. As can be seen, RPROP shows the best performance: the algorithm converges quickly to the correct configuration (zero error). In contrast to this, gradient descent with $\varepsilon = 0.001$ converges significantly slower. We repeated the experiment with a value of $\varepsilon = 0.01$. In this setting, the optimization oscillates and does not converge to the correct solution. This illustrates the sensitivity of the factor ε in gradient descent. In contrast to this, RPROP yielded a substantially better performance without the need to manually choose parameters. RPROP adapted these parameters automatically and was able to converge to the correct configuration quickly. To provide a statistical evaluation, Figure 6 depicts the results of the experiment with different configurations averaged over 10 runs. The error bars show the 95% confidence intervals. As can be seen in Figure 7, a similar convergence behavior can be observed when increasing the size of the scene as well as the number of cameras and landmarks.

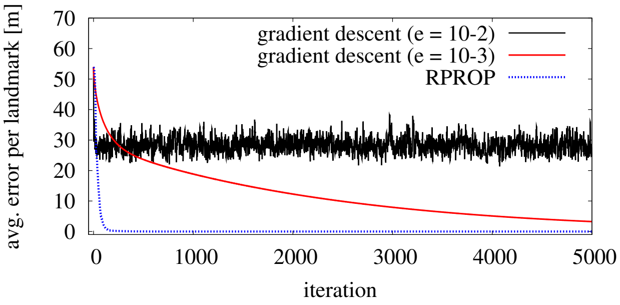

Fig. 5. Convergence behavior of a single optimization run, during which six partially visible landmarks have been observed from six geo-referenced camera locations using gradient descent with two different scale factors (ε) and RPROP. The figure illustrates that an improperly chosen ε parameter leads to oscillation.

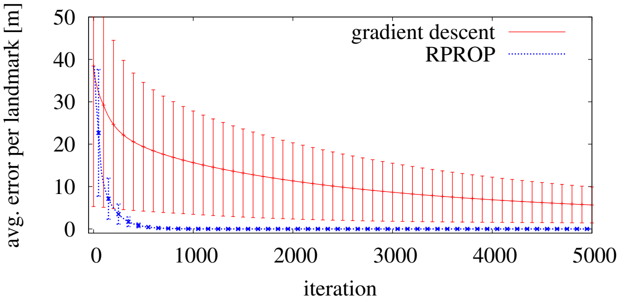

Fig. 6. Empirical evaluation of the convergence behavior of gradient descent versus RPROP. The error-bars in this plot give the 95% confidence interval for 10 runs. On a 2 GHz laptop computer, the average computation times per iteration of RPROP and gradient descent were 35 μs and 8 μs, respectively.

While RPROP is guaranteed to always converge to a solution, it is still a local optimization approach that might yield a *local* minimum. However, we found that the starting point of the optimization is not a critical choice. With randomly chosen starting locations, our approach converged in all our real world experiments to the same solution.

A more crucial precondition for success is the amount of images and landmarks and their spatial arrangement. Obviously, if only a single landmark is detected by each camera or less than two landmarks are common between different

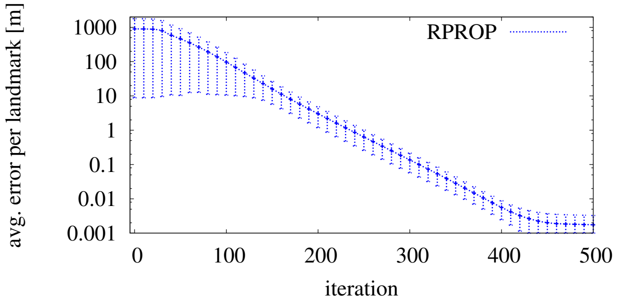

Fig. 7. Convergence behavior on an extremely large dataset. The locations of 10,000 partially visible landmarks and 1000 camera images were initialized randomly. The error-bars give the 95% confidence intervals for 10 runs. On a 2 GHz laptop computer, the average computation time per iteration was 4.5 s.

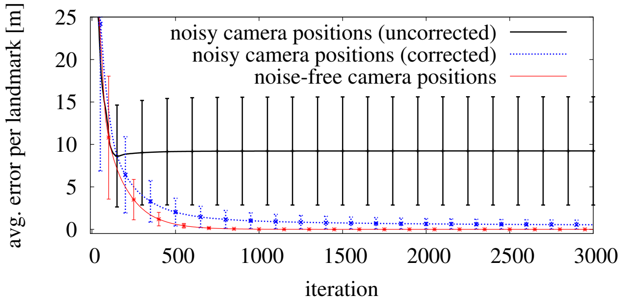

Fig. 8. Influence of the x/y-position error (resulting from the GPS) on the estimate of the landmark locations evaluated over 10 runs. The black line shows the performance of a run with a Gaussian noise of $\sigma = 10\,\mathrm{m}$ without optimizing the camera positions, while the blue line indicates the error when also optimizing the camera locations. The red line shows the result if the camera poses are error-free. As can be seen, our approach is able to compensate for this position error. The performance of the full optimization (blue) approaches the error-free model (red).

images, the system is under-constrained geometrically and cannot be solved. In our real world experiments and also in a large variety of simulated scenarios, enough geometric constraints are present to find a close to optimal solution.

Figure 8 depicts a statistical experiment which illustrates that our approach can easily compensate for the GPS inaccuracies of the geo-referenced photographs. As in all simulated experiments, we added a Gaussian noise with $\sigma = 10\,\mathrm{m}$ in the x and y-positions of the cameras. As can be seen from the diagram, our approach compensates for this noise and converges to the same estimate as if no noise was present.

VI. Conclusions

In this paper, we presented an approach to estimate the positions of landmarks based on a set of labeled, geo-referenced photographs, under absence of any camera heading information. We believe that such an approach is a first step towards allowing a mobile robot to use additional, publicly available sources of information like the image portal Flickr or Google Image Search. Our technique formulates the problem of estimating the positions of photographed objects, such as buildings in a city, as an optimization problem and uses the resilient backpropagation (RPROP) algorithm to solve it. We implemented our method and used it to estimate the locations of different buildings based on photographs taken in the city of Freiburg, Germany. Our experiments show that RPROP significantly outperforms gradient descent in this task.

Despite this encouraging results, there is further space for optimizations. To actually use image databases in a completely autonomous way, our system needs means to robustly eliminate outliers and to build representations of objects for determining correspondences. So far, we assumed the labels to be known.

To summarize, we presented an approach that is able to localize labeled objects based on geo-referenced photographs and to simultaneously estimate the unknown camera headings. This is a first step towards making such information available to robots allowing them to improve their service for applications in which spatial knowledge about the environment is required.

Acknowledgments

This work has partly been supported by the EC under contract numbers FP6-IST-034120-muFly and FP6-IST-045144-RAWSEEDS, by the DFG under contract number SFB/TR-8, and by the German Ministry for Education and Research (BMBF) through the DESIRE project.

References

[1] M. Bosse, P.M. Newman, J.J. Leonard, and S. Teller. An ATLAS framework for scalable mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Taipei, Taiwan, 2003.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(6), 2007.

[3] F. Dellaert. 4d-cities. Invited Talk at the Int. Symposium on 3D Data Processing, Visualization and Transmission, 2006.

[4] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, San Francisco, CA, USA, 2000.

[5] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proc. of the CVPR '97*, Washington, DC, USA, 1997.

[6] T. Lemaire, S. Lacroix, and J. Sola. A practical 3d bearing-only slam algorithm. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[8] P.M. Newman. *On the structure and solution of the simultaneous localization and mapping problem*. PhD thesis, University of Sydney, Australia, 1999.

[9] P.M. Newman and J.J. Leonard. Consistent convergent constant time slam. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[10] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. of the IEEE Intl. Conf. on Neural Networks*, April 1993.

[11] G. Schindler, S.B. Kang, and F. Dellaert. Inferring temporal order of images from 3d structure. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[12] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

[13] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics*, 25(3), 2006.

[14] S. Teller, M. Bosse, M. Jethwa, and A. Khripin. Automated, scalable model capture of urban environments. LCS Research Abstracts, MIT Laboratory for Computer Science, 2003.

[15] S. Thrun and colleagues. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006.