Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Landmark Selection for Task-Oriented Navigation

Ronen Lerner
and Ehud Rivlin
Department of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel
Email: ronenl@cs.technion.ac.il
ehudr@cs.technion.ac.il

Ilan Shimshoni
Department of Management Information Systems
University of Haifa
Haifa 31905, Israel
Email: ishimshoni@mis.haifa.ac.il

***Abstract*— Many vision-based navigation systems are restricted to use only a limited number of landmarks when computing the camera pose. This limitation is due to the overhead of detecting and tracking these landmarks along the image sequence. A new algorithm is proposed for subset selection from the available landmarks. This algorithm searches for the subset that yields minimal uncertainty for the obtained pose parameters. Navigation tasks have different types of goals: moving along a path, photographing an object for a long period of time etc. The significance of the various pose parameters differs for different navigation tasks. Therefore, a requirements matrix is constructed from a supplied severity function, which defines the relative importance of each parameter. This knowledge can then be used to search for the subset that minimizes the uncertainty of the important parameters, possibly at the cost of greater uncertainty in others. It is shown that the task-oriented landmark selection problem can be defined as an integer-programming problem for which a very good approximation can be obtained. The problem is then translated into a Semi-Definite Programming representation which can be rapidly solved. The feasibility and performance of the proposed algorithm is studied through simulations and lab experimentation.**

I. Introduction

In this paper the problem of *landmark-based navigation* is examined. Landmarks are distinctive features in the surrounding scene for which the 3D location is known with respect to some global coordinate system. Consider an autonomous vehicle equipped with a camera. In order to perform vision-based navigation, a set of predefined landmarks is supplied and the 2D projections on the camera's image-plane are identified and tracked during the vehicle's movement. Given the 3D and 2D data, the navigation problem is defined as the estimation of the camera *pose* (position and orientation) with respect to the global reference frame.

During the last two decades robust pose estimation algorithms have been developed by the computer vision community. These algorithms can integrate an arbitrary number of landmarks in the pose computation, leading to accurate and numerically stable results (e.g., [1], [2], [3], [4]). However, for each used landmark, its 2D measurements need to be extracted from each image along the robot's trajectory. A feature-extraction algorithm may be used in the first frame in order to identify the landmarks in the image, and some tracking algorithm will be used in the consecutive frames to obtain the 2D feature displacement. As a result, due to performance limitations, many real-time navigation systems are restricted to use only a very small number (usually 4-10) of landmarks. In [5], for example, a navigation system is presented where only four landmarks are simultaneously tracked.

If the number of available landmarks is small as well, the system will use all the visible landmarks at hand. However, if the system is equipped with a large landmark database, a subset needs to be selected from the visible landmarks as the camera moves. An example of such a scenario is an unmanned aerial vehicle (UAV) that utilizes a digital map and an ortho-photo of the observed terrain. In this configuration, the 3D locations of any point on the terrain is known, and any visually distinctive point can thus be used as a landmark. The number of potential landmarks in such a case is large, and a subset must be chosen. Another example is a Simultaneous Localization and Map Building (SLAM) system such as the one in [6], [7]. These systems estimate the camera motion and simultaneously track new features along the path of the robot's movement. The 3D locations of the tracked features are reconstructed and added to a landmark database. As a result, the database is progressively enlarged and after a while there will be too many visible landmarks in order to track them all.

While the navigating platform moves, new landmark subsets should be occasionally selected. The need for a new subset may arise, for example, when one of the landmarks leaves the camera field of view or after the camera has moved more than a certain distance since the last subset was chosen. Whenever a new subset is required, an initial guess of the camera pose can be utilized to filter the landmarks which are supposed to be visible at the moment and to predict their projection location on the image before actually detecting them. At this stage we face an important question: how do we choose the subset from the filtered landmarks wisely, in a manner that will lead to the best pose estimate according to the requirements of the specific navigation task? This *task-oriented landmark selection problem* stands at the center of the present work.

In most previous works (e.g., [6], [8]) the landmark selection problem was addressed from the image appearance point of view, where the 3D location of the landmarks was disregarded and the selection criterion based solely on a measure of

1-4244-0259-X/06/$20.00 ©2006 IEEE

distinction of the 2D features in the captured image. In [5], [9], [10] and [11], as in the present work, the 3D structure of the selected landmark constellation and its influence on the obtained accuracies was studied. In [9] a weak-perspective projection was assumed (which is inadequate for general landmark-based navigation), while in [5], [10] and [11] the navigation problem was restricted to a two-dimensional world where only three pose parameters had to be estimated.

None of the aforementioned works addressed task-oriented considerations when selecting the landmark subset. Both [5] and [9] used the condition-number of the pose covariance matrix as the landmark selection criterion. This criterion does not reflect the different severity of errors in the different pose parameters. For example, a unit error in the camera position (e.g., 1 cm) should not be considered equivalent to an angular unit error (e.g., 1 radian) in its orientation. Additionally, the purpose of the pose computation should not be overlooked. The navigation system usually supports a control system that uses the pose estimates to perform some predefined task. According to the requirements of the specific task, some of the pose parameters may be considered more essential than others. For example, if the platform needs to follow a predefined path, then accurately identifying its location along the path is not as important as identifying any drifts from the path. Another example is the task of landing an airplane on a landing track. Obviously, the set of relevant parameters and accuracies during landing differs from those that need to be controlled for maintaining straight and level flight. In this paper we present a new criterion for *task-oriented* landmark selection which prefers a subset of landmarks that minimizes the error in some of the pose parameters even at the expense of larger errors in the other parameters according to the requirements of the task.

The paper continues as follows. Section II reviews the topic of pose estimation from landmarks and its uncertainty. Section III shows how the system designer can use a severity function to specify the adequacy of different poses for the specific navigation task. This function can be used to construct a requirement matrix that reflects the importance of the different pose parameters for the task at hand. A method to evaluate how well the different subsets conform to the requirements of the navigation task is developed in Section IV. Section V presents a good and efficient approximate solution to the subset selection problem which uses Semi-Definite Programming. A solution for this class of optimizations can be found easily and rapidly, qualifying the proposed algorithm for real-time navigation systems. Experimental results on simulated and real data are presented in Section VI. We conclude in Section VII.

II. Landmark-Based Navigation

Before considering the *task-oriented landmark selection problem* we briefly summarize the *landmark-based navigation problem*. Let $P_i \in \mathbb{R}^3$ $(i = 1, ..., n)$ be a set of available landmarks. The 3D location of these points is assumed to be known with respect to some reference coordinate system W. In order for an autonomous vehicle to navigate in this scene, it is equipped with a calibrated camera, for which another Cartesian coordinate system, denoted C, is attached. Traditionally, the origin of this system coincides with the camera's center of projection and the Z-axis is oriented along the optical axis. The *pose* of the camera with respect to W can be represented by an orthonormal rotation matrix, $R \in SO(3)$, and by the camera position vector $t \in \mathbb{R}^3$ such that

$$^{C}P_i = R^T \left(P_i - t\right), \tag{1}$$

where $^{C}P_i$ is the representation of P_i in the camera's system C. Due to the orthonormality of R, the camera's orientation has only three degrees of freedom, usually represented by the Euler-angles ϕ, θ, and ψ, which reflect the rotation around the X, Y, and Z axes respectively. Thus, the camera pose is fully defined by a 6D parameter vector, $\Theta = (\phi, \theta, \psi, t^T)^T$.

In the camera frame, the 3D landmarks are perspectively projected to their 2D location in the image-plane. Let $f_i : \mathbb{R}^6 \mapsto \mathbb{R}^2$ be the perspective projection function of the i th landmark:

$$f_i(\Theta) = (u_i, v_i)^T = \left(^{C}P_{ix}/^{C}P_{iz}\ ,\ ^{C}P_{iy}/^{C}P_{iz}\right)^T. \tag{2}$$

Given the 3D landmarks and their corresponding 2D camera measurements $(\hat{u}_i, \hat{v}_i)^T$, the navigation problem is to accurately and robustly estimate the camera pose parameters - $\hat{\Theta}$. Usually, these parameters are estimated by a non-linear optimization procedure that minimizes the squared error between the camera's 2D measurements and the landmark projections (which are calculated using the pose hypothesis):

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{n} \| f_i(\Theta) - (\hat{u}_i, \hat{v}_i)^T \|^2. \tag{3}$$

A. The Pose Covariance Matrix

The 2D measurements obtained from the camera are not error-free. These errors occur due to errors in the feature detection procedure and are commonly modelled as independently and identically distributed Gaussian additive errors. Let σ_I be the standard deviation of this isotropic Gaussian distribution. In the absence of these errors the exact pose Θ would have been obtained; however, in realistic scenarios these errors propagate through the optimization process and lead to the perturbed estimate of the pose $\hat{\Theta}$.

The Jacobian J_i of the i th landmark is the 2×6 matrix containing all f_i's partial derivatives, and the Jacobian matrix of all the landmarks which participate in the pose computation is defined as the concatenation of all the respective J_is:

$$J = [J_1^T, \ldots, J_n^T]^T. \tag{4}$$

Following the derivations in [12], a first-order approximation of the error propagation from image measurements to the pose parameters is given by the covariance matrix of Θ:

$$\Sigma_\Theta = (J^T J)^{-1} J^T \Sigma_I J (J^T J)^{-1}, \tag{5}$$

where Σ_I in the above expression is the image measurements' covariance matrix, which reflects the errors in the 2D measurements. Since it was assumed that these errors are i.i.d and isotropic, Σ_I takes the special form of a diagonal matrix with constant value σ_I^2 along the diagonal. Hence, the expression for the pose covariance matrix may be simplified as follows:

$$\begin{aligned}\Sigma_\Theta &= \left(J^T J\right)^{-1} J^T \left(\sigma_I^2 I\right) J \left(J^T J\right)^{-1} \\ &= \sigma_I^2 \left(J^T J\right)^{-1} = \sigma_I^2 \left(\sum_{i=1}^{n} J_i^T J_i\right)^{-1}. \end{aligned} \tag{6}$$

The diagonal of the pose covariance matrix contains the variances of the six pose parameters, while the off-diagonal elements represent the dependencies between these parameters. This symmetric matrix also represents a 6D ellipsoid (usually known as the *uncertainty ellipsoid*) in the pose configuration space. The main axes of this ellipsoid are in the direction of Σ_Θ's eigenvectors and their lengths correspond to the square-roots of Σ_Θ's eigenvalues. One can think of this ellipsoid as an approximation of the volume in which the real pose is located up to some certainty. For accurate pose estimates this volume will be relatively small.

III. The Requirement Matrix Construction

It is clear that different selections of landmark subsets will lead to different pose accuracies. As an illustrative example, consider the choice of a subset containing landmarks with very small distances between them. Their projection rays will form a very narrow bundle, which will in turn lead to a very inaccurate pose estimate as compared to a subset of landmarks that are far away from each other. The *landmark selection problem* is simply defined as the problem of finding the best subset – the one that will lead to the most accurate pose.

As was already shown in section II-A, the pose accuracy is not represented by a scalar but rather by a 6×6 covariance matrix. Any landmark subset will lead to a different covariance matrix. This leads to a fundamental question: given two covariance matrices, which one is "better"? Each covariance matrix reflects an uncertainty ellipsoid. If one of the ellipsoids contains the other, then it is clear that the smaller one should be preferred. For example, one can see that ellipsoids B and C in Fig. 1 are preferable to ellipsoid A. However, the choice between ellipsoids B and C is less obvious and should take into account the requirements of the specific navigation task. For example, if for some reason the x-parameter is much more important than the y-parameter to our navigation task, it may be preferable to choose ellipsoid C over ellipsoid B although it has higher uncertainty along the (less important) y direction. Additionally, the configuration space should not be perceived as a Euclidian space. An error of one angular unit (e.g., radian) in the camera orientation is not equivalent to an error of one metric unit (e.g., centimeter) in the pose translation vector. In order to deal with the aforementioned issues, a 6×6 *requirements matrix* should be constructed and supplied by the system designer who is familiar with the requirements of the specific navigation task. The requirements matrix, denoted Σ_R, should be symmetric, positive semi-definite, and reflect the importance of the different pose parameters (particularly the correct balance between angular and translational errors). This matrix will be used to induce a Mahalanobis norm on

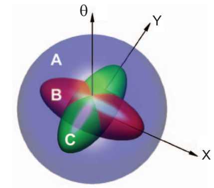


Fig. 1. Comparison between uncertainty ellipsoids in a 3D pose configuration space. It is clear that ellipsoids B and C are preferable comparing to A, but the choice between these two is less obvious and should take into account the requirements of the specific navigation task

the pose configuration space, so the severity of every pose perturbation - $\Delta\Theta$ can be evaluated:

$$perturbation\ severity = \|\Delta\Theta\|_R^2 = \Delta\Theta^T \cdot \Sigma_R \cdot \Delta\Theta \tag{7}$$

where $\|\cdot\|_R$ denotes the Mahalanobis norm that is induced by Σ_R.

In order to construct this matrix, a *pose-severity* function, denoted $S(\Theta)$, is defined by the system designer. This function evaluates how "bad" the pose is for the specific task. For example, if our task is to photograph an object in the scene, then a proper severity function could be the 2D distance between the object's projection and the principle-point. Such a severity function reflects the desire to keep the object at the center of the image. Another classical example appears when the task is to follow some predefined trajectory. In this case, a reasonable severity function could measure the distance of the camera from the trajectory. Landing an airplane is an example of such a task: the trajectory leads the airplane along the landing track in a smooth and tangential manner.

Next, a close camera pose Θ_0 which is optimal according to the severity function S is chosen. Thus, the value and gradient of S at this pose vanish. In this simplified case, the second order approximation of S at any perturbed Θ is:

$$S(\Theta) = \frac{1}{2}\left(\Theta - \Theta_0\right)^T H_S(\Theta_0)\left(\Theta - \Theta_0\right), \tag{8}$$

where H_S is the Hessian matrix of S. The vector $(\Theta - \Theta_0)$ represents the perturbation $(\Delta\Theta)$ in the pose's configuration space. Comparing the above result to (7), we observe that the Hessian is proportional to the desired Σ_R and thus may serve as the requirements matrix.

IV. Task-Oriented Grading of Landmark Subsets

In [5] and [9] each landmark subset was graded according to the condition number of its covariance matrix, which is defined as the ratio between the largest and smallest eigenvalues. In terms of uncertainty ellipsoids, the condition number is the squared lengths ratio of the longest and shortest main-axes, thus measuring the "roundness" of the ellipsoid. Using this criterion will bring us to choose the landmark subset with the most spherical uncertainty ellipsoid. Note that in our 3D example of Fig. 1, ellipsoid A would have been chosen according to the condition number. Another problem with this criterion is that it perceives the configuration space as a Euclidian space, and thus doesn't reflect the real severity balance between angular and translational errors or between the different pose parameters according to the requirements of the task.

A new grading criterion for landmark subsets is proposed. First, instead of grading according to the uncertainty ellipsoid's roundness, we would like to use a criterion that reflects its size. Two straightforward alternatives are the summation and the multiplication of the covariance matrix eigenvalues. These quantities can be easily obtained as the covariance matrix's trace and determinant respectively. At first sight, it seems like the product of the eigenvalues would be a better choice since it is proportional to the squared volume of the uncertainty ellipsoid. However, such a criterion might prefer an ellipsoid with very long axis when the rest of the axes are very short and hence compensate for the long one. When summing the eigenvalues, on the other hand, the squared lengths of the axes are summed and hence will be relatively large even if only one of the axes is long. Additionally, the length of the uncertainty ellipsoid's main axes will not be measured using the Euclidian norm but rather by the Mahalanobis norm induced by the requirements matrix that was supplied.

Given a covariance matrix Σ_Θ that was obtained from a landmark subset, the grading criterion is developed as follows. Let $\Sigma_\Theta = M_\Theta \Lambda_\Theta M_\Theta^T$ be the eigenvectors-eigenvalues decomposition of Σ_Θ. $M_\Theta = [\hat{m}_1, \cdots, \hat{m}_6]$ is an orthonormal matrix in which the eigenvectors of the covariance matrix are its columns, and Λ_Θ is a diagonal matrix containing the eigenvalues - λ_i. Therefore, the grade of the covariance matrix, which is defined to be the sum of squared Mahalanobis lengths of the ellipsoid's main axes, is:

$$grade = \sum_{i=1}^{6} \left\| \sqrt{\lambda_i} \cdot \hat{m}_i \right\|_R^2 = \sum_{i=1}^{6} \lambda_i \cdot \hat{m}_i^T \Sigma_R \hat{m}_i, \tag{9}$$

In contrast to a simple summation of Σ_Θ's eigenvalues, here we obtained their *weighted* sum. The weights $\hat{m}_i^T \Sigma_R \hat{m}_i$ represent the severity of the pose errors in the $\hat{m}_i$ direction.

During the optimization process, where the landmark subset with the minimal grade is sought, the grade function is evaluated many times for different subsets. Therefore, in order to reduce the overhead of the optimization, it would be desirable to avoid the eigenvectors-eigenvalues decomposition of Σ_Θ. Since it is only the *sum* of squared lengths that we need for the grade computation, this function takes much simpler form as:

$$grade = tr\left[\Sigma_R \Sigma_\Theta\right], \tag{10}$$

where $tr\left[\,\right]$ represent the matrix trace. The two grade definitions (9) and (10) are equivalent since:

$$\begin{aligned} tr\left[\Sigma_R\Sigma_\Theta\right] &= tr\left[\Sigma_R M_\Theta \Lambda_\Theta M_\Theta^T\right] = tr\left[\sqrt{\Lambda_\Theta} M_\Theta^T \Sigma_R M_\Theta \sqrt{\Lambda_\Theta}\right] = \\ &= tr\left[\left[\sqrt{\lambda_i\lambda_j}\hat{m}_i^T\Sigma_R\hat{m}_j\right]_{i,j=1,\ldots,6}\right] = \sum_{i=1}^{6} \lambda_i \cdot \hat{m}_i^T \Sigma_R \hat{m}_i \ , \end{aligned}$$

where $\sqrt{\Lambda_\Theta}$ is the diagonal matrix containing the square roots of the six eigenvalues – $\sqrt{\lambda_i}$. In the above manipulation we used a cyclic permutation of the matrices in the trace. Such a permutation is known to preserve the trace.

V. Approximate Solution for the Subset Selection Problem

Equipped with the task-oriented grading criterion, we can address the central problem of this work: the *task-oriented landmark selection problem*. Given a set of n available and visible landmarks, we would like to obtain the best landmark subset of some predefined size k ($k < n$). This problem can be posed as an integer programming optimization problem by introducing n indicator variables, $\alpha_i \in \{0,1\}$ $(i = 1, \ldots, n)$, each indicating whether the corresponding landmark was selected to the subset. Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ be the vector concatenation of these variables. Stipulating the participation of each J_i in (6) according to its corresponding α_i and ignoring the constant factor σ_I^2 yields the subset's covariance matrix:

$$\Sigma_\Theta(\alpha) = \left(\sum_{i=1}^{n} \alpha_i \cdot J_i^T J_i \right)^{-1}. \tag{11}$$

By substituting (11) into (10), the integer-program becomes:

$$\begin{aligned} \arg\min_{\alpha} \quad & tr\left[\Sigma_R \left(\sum_{i=1}^{n} \alpha_i \cdot J_i^T J_i\right)^{-1}\right] \\ s.t: \quad & \\ & \sum_{i=1}^{n} \alpha_i = k \quad , \quad \alpha_i \in \{0,1\}. \end{aligned} \tag{12}$$

The first constraint guarantees that the obtained subset size will be as required, while the second constraint enforces the Boolean behavior of the indicators.

Computing the exact solution for this program is NP-Hard. However, a very good approximation can be obtained by solving the problem *relaxation*, where the Boolean restriction of the α_is is replaced by the relaxed constraint $0 \le \alpha_i \le 1$. The objective function of this program is convex and can be solved using any non-linear optimization toolbox (e.g., [13]) to obtain the *fractional* solution. In order to decide which of the landmarks should be selected, a rounding heuristic should be applied to the obtained fractional α_i variables. A well known rounding method [14] proceeds as follows: each of the fractional α_is is perceived as the probability that the corresponding landmark will be selected to the subset. Hence, several subsets are randomly constructed according to these probabilities; next, the grade of each subset is evaluated according to (10), and the subset with the minimal grade is chosen. Note that although the subsets' size expectation is k as desired (due to the subset size constraint), the actual size of the randomly generated subsets may be slightly different. Therefore, the random subsets should be corrected by adding or discarding landmarks in order to reach the necessary size, where the choice of which landmarks to add/discard is in accordance with the fractional α_is' values: for subsets that are too large we will discard the landmarks with the lowest α_i, and for subsets that are too small we will add the landmarks with the largest α_i. Experimental results, which are presented in section VI, demonstrate that this scheme can obtain very good approximations for the optimal solution.

A. Posing the Relaxed Program as an SDP

The relaxed problem is a constrained non-linear optimization problem. Although it can be solved using general optimization toolboxes (e.g., [13]), the overhead of converging to an accurate solution might be large, thus disqualifying the proposed method for real-time navigation systems. However, this problem can be easily converted to a *Semi-Definite Programming (SDP)* problem for which powerful and very efficient algorithms exist [14], [15], [16], [17]. One can think of SDP as an extension of the well-known linear programming, in which the linear inequality constraints are extended by the so-called *Linear Matrix Inequality (LMI)* constraint. Such an LMI constraint on the α variables should be in the form:

$$\sum_{i=1}^{n} A_i \cdot \alpha_i + C \succeq 0, \tag{13}$$

where A_i and C are symmetric matrices and the notation $P \succeq Q$ reflects that $P - Q$ should be positive semi-definite. Despite its name, one can see that such a constraint can express non-linear behavior through the requirement of the matrix positiveness.

Note that in the SDP formulation the objective function is still required to be linear in the problem's variables. In order to transfer the non-linearity of $\Sigma_\Theta(\alpha)$ from the objective function to the problem constraints (where non-linearity can be handled), we introduce 21 additional slack variables arranged in a 6×6 symmetric matrix Y. Using the new variables the problem can be redefined as:

$$\begin{aligned} &\arg\min_{\alpha,Y} \quad tr\left[\Sigma_R Y\right] \\ &s.t: \\ &\qquad Y \succeq \Sigma_\Theta(\alpha) \\ &\qquad \sum_{i=1}^{n} \alpha_i = k \quad , \quad 0 \le \alpha_i \le 1. \end{aligned} \tag{14}$$

In order to verify that (12) and (14) are equivalent, one needs to show that the following two conditions hold:

- Every feasible solution of (12) can be extended to a feasible solution of (14) by setting some values to Y such that the two objective functions coincide (which implies that the optimum of (12) $\ge$ the optimum of (14)).
- The objective function of (12) is a lower bound of the objective function in (14) for any given α and Y(implying that the optimum of (12) $\le$ the optimum of (14)).

The first condition is easily verified by letting Y be equal to $\Sigma_\Theta(\alpha)$. The second condition is proved in the following lemma:

Lemma 1: Let $Y \succeq \Sigma_\Theta(\alpha)$ and $\Sigma_R \succeq 0$ be defined as before. Then:

$$tr\left[\Sigma_R Y\right] \ge tr\left[\Sigma_R \Sigma_\Theta(\alpha)\right].$$

Proof:

$$tr\left[\Sigma_R Y\right] - tr\left[\Sigma_R \Sigma_\Theta(\alpha)\right] = tr\left[\Sigma_R\left(Y - \Sigma_\Theta(\alpha)\right)\right] =$$

$$tr\left[UU^T\left(Y - \Sigma_\Theta(\alpha)\right)\right] = tr[U^T \underbrace{\left(Y - \Sigma_\Theta(\alpha)\right)}_{\succeq 0} U] \ge 0.$$

In the above derivation, the requirements matrix was decomposed using Cholesky decomposition into $\Sigma_R = UU^T$. The next step is based on the well-known property that multiplying positive semi-definite matrix A from both sides by any matrix, $U^T AU$, will not effect its positiveness. The last inequality results from the fact that the matrix trace is equal to the sum of its eigenvalues, which are all non-negative for positive semi-definite matrices. ■

In order to represent the non-linear constraint in (14) as an LMI, the *Schur complement lemma* will be used:

Lemma 2 (Schur complement lemma): Let

$$A = \begin{bmatrix} B & C^T \\ C & D \end{bmatrix}$$

be a symmetric matrix where B is positive definite. Then, A is positive semi-definite iff $D - CB^{-1}C^T$ is positive semi-definite.

See [14] for a proof of this lemma. Thus, the constraint $Y \succeq \Sigma_\Theta(\alpha)$ can be replaced by:

$$\begin{bmatrix} \Sigma_\Theta^{-1}(\alpha) & I \\ I & Y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \alpha_i J_i^T J_i & I \\ I & Y \end{bmatrix} \succeq 0. \tag{15}$$

Finally, the LMI representation of our constraint is:

$$\sum_{i=1}^{n} \begin{bmatrix} J_i^T J_i & 0 \\ 0 & 0 \end{bmatrix} \alpha_i + \sum_{j=1}^{21} \begin{bmatrix} 0 & 0 \\ 0 & E_j \end{bmatrix} y_j + \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \succeq 0, \tag{16}$$

where E_i $(i = 1, \ldots, 21)$ are 6×6 matrices with all elements equal to zero except the entries of the corresponding y_i in Y which are set to one.

The relaxed problem in its new formulation can be fed into an SDP toolbox such as [18], [19], [20] to rapidly obtain its solution. Such toolboxes solve semi-definite problems using *interior point* algorithms which simultaneously optimize two problems: the original minimization problem, which is known as the *primal* problem and its *dual* maximization problem. As in the linear programming scenario, the optimum of the two problems coincides. This gives us a very simple stopping criterion for the optimization process, by monitoring the decreasing gap between the two solutions. The convergence speed of interior point algorithms is known to be exponential. Together with the convexity of the problem, this implies that a correct solution with the desired accuracy can be obtained in almost a fixed number of iterations regardless of the quality of the initial guess. On a Pentium 4 machine full convergence is reached after about 0.1 seconds for 10 available landmarks and about 0.5 seconds in the case of 100 available landmarks. Much faster results can be obtained with a small compromise on the obtained accuracy by stoping the iterative process before full convergence is reached. Keeping in mind that the selection procedure should be activated only once in a while, its time

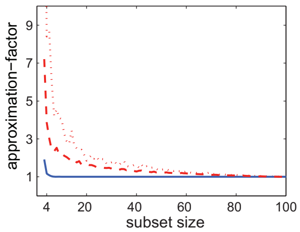


Fig. 2. Approximation factors that were obtained for different subset sizes. The blue solid line was obtained by the algorithm, the red dotted line by taking the best of 10 uniformly selected subsets , and the red dashed line by taking the best of 100 uniformly selected subsets.

consumption is not too high for reasonable problem sizes, and thus can be integrated into real-time navigation and control systems of autonomous robots.

VI. Results

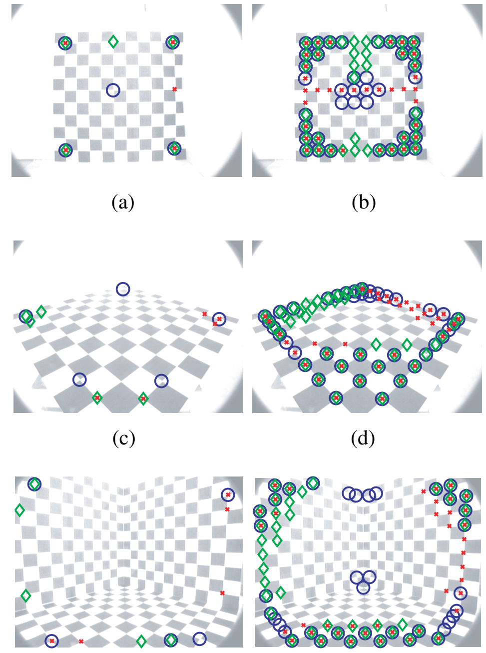
In this section the performance and the advantages of the proposed algorithm are demonstrated through simulations and lab experimentation.

Obtaining the dual value as part of the SDP solution is advantageous: it is a lower bound on the primal grade optimum which is in itself a lower bound on any feasible integral solution of the original problem. Hence, one can use the dual value to obtain an *upper* bound on the approximation factor of any evaluated landmark subset. In Fig. 2 the approximation factors obtained by the algorithm are evaluated. A set of 100 landmarks was synthetically generated from which subsets of different sizes were selected. One can see that the obtained approximation is very good, almost 1 for any subset size larger than 3. For comparison, groups of 10 and 100 subsets were selected uniformly as well (the dotted and dashed red lines in Fig. 2). As could be expected, the approximations obtained by this method were similar to those of the proposed algorithm when the subset size was near 100. However, a clear and drastic advantage can be observed in the more realistic scenarios where small subsets are selected.

In order to demonstrate the advantage of the proposed algorithm in real scenarios, two lab experiments were conducted: one with still images and the other with a video that was captured while a robotic arm was preforming some tasks.

A. Still Images Experiment

For the first experiment two environments were constructed: the first one contained 100 coplanar landmarks that were defined by the squares' corners on a 10×10 chessboard (see Fig. 3(a) or 3(c)), and the other contained 300 landmarks from 3 orthogonal chessboards (see Fig. 3(e)). A calibrated camera was placed in various positions and orientations in the two environments and images of 640×480 were captured. First, the ground-truth camera pose was calculated from all visible landmarks. Next, different tasks were defined and subsets with size ranging from 4 to 50 landmarks were selected accordingly using the task-oriented algorithm. Fig. 3 shows the selected subsets of size 5 and 40 for three examined tasks: the first requires only the X-component of the camera position, the second requires the Y-component, and the third task is one in which an object is photographed as described in section III. One can see that different subsets were automatically selected as a result of the different task definitions.



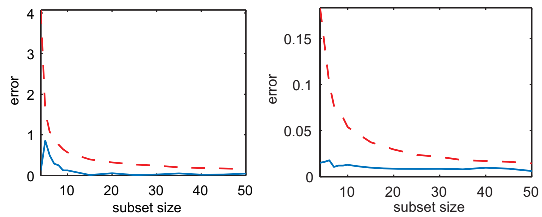
(a) (b) (c) (d) (e) (f)

Fig. 3. Subset selection of 5 and 40 landmarks for different tasks. (a) and (b) show coplanar landmarks parallel to the image plane, (c) and (d) show coplanar landmarks in general position, (e) and (f) show landmarks placed on 3 orthogonal planes. In all six images the markers represent the selected landmarks according to different navigational tasks: a red 'x' - a task of computing the X component of camera position, a green diamond - a task of computing the Y component of camera position, a blue circle - a photographing task.

Next, for each examined task and subset size, additional 500 subsets were uniformly selected for comparison. Fig. 4 compares the weighted (Mahalanobis) error of the pose obtained by the algorithm's selected subset to the mean weighted error of the poses when using the uniformly selected subsets. All these subsets were selected from the environment presented in Fig. 3(e). A clear advantage of the proposed algorithm can be observed for all subset sizes although this advantage diminishes for large subsets, as in Fig. 2.

B. Robot Experiment

In this experiment a path-following task was performed using landmarks that were selected by the proposed algorithm. A video camera with a resolution of 720×428 pixels was attached to a robotic arm (see Fig. 5(a)). This arm can be manipulated in 6 d.o.f and supplies the trajectory in which it was maneuvered up to sub-millimetric accuracy. This positional information that was gathered from the odometry sensors of the robot was not used during the navigation task but rather



(a) (b)

Fig. 4. The weighted error obtained by different subset sizes for the scene presented in Fig. 3(e). The dashed line shows the pose mean error when using the uniformly selected subsets, the blue solid line shows the pose error when using the selected subset of the algorithm. (a) the navigation task requires the X component of camera position, (b) shows the results for the photographing task.

(a) (b)

Fig. 5. (a) The video camera mounted on 6 d.o.f robotic arm. (b) A frame captured by the camera. The red circles mark the 6 landmarks that were selected at this frame using the selection algorithm.

was collected and saved as a ground truth for the algorithm evaluation.

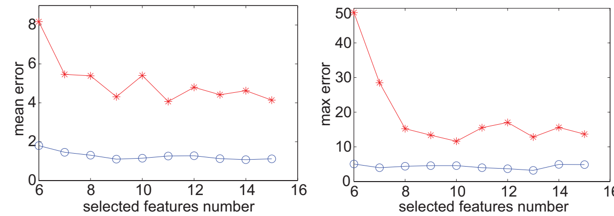
A scene was constructed from 34 landmarks lying on three orthogonal planes. Thirty of them were located in a relatively dense cluster while the other 4 were dispersed in different locations (see Fig. 5(b)).

While the camera was in motion, subsets of different sizes (6-15 landmarks) were selected whenever required: when one of the landmarks left the field of view, or after the camera pose shifted beyond a certain threshold. Fig. 5(b) shows an example of such a subset selected by the proposed algorithm. Note that the dispersed landmarks of the scene were automatically selected by the algorithm.

As part of the control loop, the camera pose was constantly estimated on the basis of the selected landmarks and, as a consequence, the robotic arm trajectory was periodically adjusted. For any of the tested subset sizes both the mean and maximum task-related errors of the obtained trajectories are reduced when using the algorithm to select the landmarks, as can be observed in Fig. 6(a) and 6(b).

VII. Conclusion

In this paper a new algorithm for landmark selection was proposed. Due to performance limitations a real-time navigation system can usually use only a small number of landmarks for the pose computation. It was shown that by defining the specific task requirements in the form of a requirements matrix, different subsets from the available landmarks are automatically selected. The obtained subset yields minimal uncertainty for the pose parameters according to the Mahalanobis



(a) (b)

Fig. 6. Mean (a) and maximal (b) millimetric errors of the trajectories obtained when selecting subsets of different sizes. The blue line shows the results obtained when using the selection algorithm. The red line shows the results obtained when using arbitrary selection of landmarks.

metric defined using the requirements matrix. Simulations and experimentations verify the advantages of integrating the proposed algorithm in real-time navigation systems.

References

[1] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *Int. J. of Comp. Vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[2] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidy, and M. Kim, "Pose estimation from corresponding point data," *IEEE Trans. on Systems, Man, And Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

[3] Y. Hel-Or and M. Werman, "Absolute orientation from uncertain point data: A unified approach," in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 1992, pp. 77–82.

[4] Y. Liu, T. Huang, and O. Faugeras, "Determination of camera location from 2-d to 3-d line and point correspondences," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 28–37, 1990.

[5] D. Burschka, J. Geiman, and G. Hager, "Optimal landmark configuration for vision-based control of mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 3917–3922.

[6] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 2051–2058.

[7] C. Harris, *Active vision*. Cambridge, MA, USA: MIT Press, 1993.

[8] C. Tomasi and J. Shi, "Good features to track," in *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 1994, pp. 593–600.

[9] A. Bruckstein, R. Holt, T. Huang, and A. Netravali, "Optimum fiducials under weak perspective projection," *Int. J. of Comp. Vision*, vol. 35, no. 3, pp. 223–244, 1999.

[10] C. B. Madsen and C. S. Andersen, "Optimal landmark selection for triangulation of robot position," *J. of Robotics and Autonomous Systems*, vol. 13, no. 4, pp. 227–292, 1998.

[11] K. Sutherland and W. Thompson, "Inexact navigation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1993, pp. I: 1–7.

[12] R. Haralick, "Propagating covariance in computer vision," *Int. J. of Patt. Recog. and Art. Intell.*, vol. 10, no. 5, pp. 561–572, 1996.

[13] "Matlab optimization toolbox," 1995, The Math Works, http://www.mathworks.com/products/optimization/.

[14] A. Ben-Tal and A. Nemirovski, Eds., *Lectures on Modern Convex Optimization*. MPS-SIAM Series on Optimization, 2001.

[15] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[16] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. Optim.*, vol. 5, no. 1, pp. 13–51, 1995.

[17] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior–point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, 1996.

[18] S. Benson, Y. Ye, and X. Zhang, "DSDP toolbox," http://www-unix.mcs.anl.gov/∼benson/dsdp/.

[19] K. Fujisawa, M. Kujima, K. Nakata, and M. Yamashita, "SDPA toolbox," http://grid.r.dendai.ac.jp/sdpa/.

[20] J. Sturm, "SeDumi toolbox," http://fewcal.kub.nl/sturm/software/sedumi.html.