Enhanced Simultaneous Camera Calibration and Path Estimation

Melanie B. Rudoy, Charles E. Rohrs
Massachusetts Institute of Technology
Digital Signal Processing Group
77 Massachusetts Avenue, Cambridge, MA 02139
{mbs, crohrs}@mit.edu

***Abstract*— This paper addresses two issues related to the simultaneous calibration of a network of imaging sensors and the recovery of the trajectory of a single target moving among them. The non-overlapping fields of view for the cameras do not cover the entire scene, resulting in times for which no measurements are available. A Bayesian framework is imposed on the problem in order to compute the MAP (maximum a posteriori) estimate for both the trajectory of the target and the translation and rotation of each camera within the global scene. First, three model order reduction techniques that decrease the dimension of the search space and the number of terms in the objective function are presented, thereby reducing the computational requirements of the search algorithm used to solve the optimization problem. Next, the problem of finding a solution that is consistent with the set of observation times is addressed, so that the target's estimated state does not fall within the field of view of the sensor network at a time for which no measurement is available. Three techniques that treat the missing measurements as additional inequality or equality constraints within the MAP optimization framework are presented.**

I. Introduction

In many situations it is not possible to calibrate a network of sensors prior to the start of data collection. It may be undesirable to throw away good data while the sensors undergo automatic calibration using training data or external training sources embedded in the environment. In many practical scenarios, such as inside buildings or in urban environments, it may not be possible to use GPS or other RF techniques for self-localization [1][2]. There are many drawbacks to these various calibration techniques, including functional, memory, power, and processing requirements for each node. There are many scenarios in which the camera network must remain passive and undetectable by outside observers, thereby prohibiting the use of active training sources or active radio links. In these scenarios, a passive joint calibration and path recovery scheme is desired.

It has been previously demonstrated that it is possible to perform calibration and path recovery at the same time, using only the raw data collected locally by each camera [3]. The non-overlapping fields of view for the cameras do not cover the entire global scene, as depicted in Fig. 1, and therefore there are times for which no measurements are available. One approach to recover the calibration parameters and the target trajectory is to impose a Bayesian framework on the problem and to compute the maximum a posteriori (MAP) estimate. A Gaussian prior probability is imposed on the camera calibration parameters, and the target's trajectory is assumed to evolve according to linear, Gauss-Markov dynamics. The measurement likelihood function is nonlinear due to the fact that the cameras may be arbitrarily rotated within the global map, and subsequently the objective function is also nonlinear and nonconvex. However, the objective function is well approximated by a quadratic form, and therefore a Newton-Raphson search method is adopted. While this technique solves the problem at hand, this batch algorithm quickly becomes computationally intractable as more data becomes available.

First, three model order reduction techniques to reduce the dimensionality of the search space and the number of terms in the objective function are presented. The first technique removes states from the augmented path corresponding to times for which no measurements are available. Once the solution to the primary MAP optimization problem has been obtained, the path corresponding to missing measurements can be computed using the a priori target motion model and the estimates of the target's state upon exit and return to the camera network. The second technique uses position measurements and derived local velocity estimates at the entry and exit points to compute the optimal sensor configuration. The last model order reduction technique exploits the small measurement noise to treat the observations as equality constraints in the optimization problem.

At times, the target's estimated trajectory passes within the field of view of a particular camera even though no corresponding measurements are available, thus violating the assumption that the probability of detection is unity everywhere within the sensor network. Most estimation schemes rely only on the target's a priori motion model to impute its overall trajectory, without taking into consideration the additional information provided about where the target is prohibited from being when no observations are available. For instance, in a Kalman filter, if a measurement is missing at time k, the optimal action in the mean-squared error sense is to replace the update step, $\hat{x}_{k|k}$, with the optimal predictor, $\hat{x}_{k|k-1}$ [4]. This action may place the target within the field of view of the sensor network at time k, contradicting the fact that no measurement is available.

Three distinct methods to ensure that the final solution is

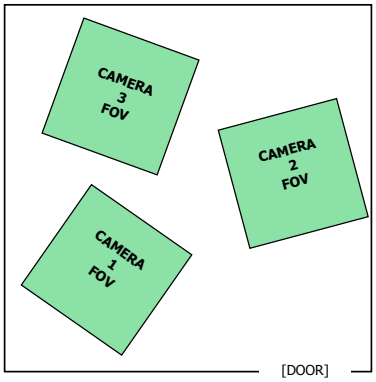


Fig. 1. Sample camera configuration (aerial view).

feasible with respect to the probability of detection assumption are presented. The first approach recasts the problem within a nonlinear, mixed-integer programming framework, resulting in a disjunctive constraint set. Though optimal, this solution quickly becomes computationally intractable as the number of missing measurements increases. The second method, designed to approximately solve the mixed-integer program, is an adaptive Newton-Raphson search algorithm that systematically explores only the feasible subsets of the search space, adding and removing equality constraints as needed. In the search process, the algorithm pushes path segments with no measurements around the corners of the sensors, rather than through the field of view of the network. Finally, another approximation is discussed involving the use of circular constraints to approximate the actual square sensor boundaries, which results in a single, although highly constrained, optimization problem to solve.

II. Motion and Measurement Models

The target's state consists of four variables representing position and velocity in the global coordinate system. The horizontal direction is denoted by u and the vertical as v, so that the target's state at each time step is given by

$$x_t = \begin{bmatrix} u_t & \dot{u}_t & v_t & \dot{v}_t \end{bmatrix}^T.$$

In the a priori model, the state evolves according to linear, Gaussian, Markovian dynamics, as:

$$x_{t+1} = Ax_t + \nu_t \tag{1}$$

where ν_t is a zero mean Gaussian random variable with covariance Σ_ν. From the above characterization, it is well known that the transition density is also Gaussian:

$$p\left(x_t|x_{t-1}\right) = \mathcal{N}\left(x_t; Ax_{t-1}, \Sigma_\nu\right) \tag{2}$$

Let the variable x denote the stacked position and velocity trajectory across all time. Denote the target's initial location as x_0, where x_0 is often modelled as a Gaussian random variable. The prior probability $p(x)$ over the entire stacked trajectory is given by:

$$p(x) = p(x_0)\prod_{t=1}^{T} p\left(x_t|x_{t-1}\right) \tag{3}$$

The observations are supplied by a network of non-overlapping cameras, each of which reports to a central processor the position of the target in its own local coordinate system when the target appears in its field of view, along with a time stamp and unique camera ID. There are three unknown parameters associated with each camera, corresponding to horizontal and vertical translation and rotation about a reference direction (e.g. north), denoted as $\mu^i = \begin{bmatrix} p_u^i & p_v^i & \theta^i \end{bmatrix}^T$, as shown in Fig. 2. For simplicity, let $p^i = \begin{bmatrix} p_u^i & p_v^i \end{bmatrix}^T$ and let $\mu = \begin{bmatrix} \mu^1 & \mu^2 & \cdots & \mu^N \end{bmatrix}^T$ be the collection of stacked sensor parameters for all N cameras. The rotation matrix for sensor i is:

$$R(\theta^i) = \begin{bmatrix} \cos(\theta^i) & \sin(\theta^i) \\ -\sin(\theta^i) & \cos(\theta^i) \end{bmatrix}$$

Denote $p(\mu)$ as the prior distribution on the entire set of sensor calibration parameters, which is Gaussian with mean μ_0 and covariance Σ_μ. In $p(\mu)$, the location of one sensor is fixed, due to the fact that any configuration of x and μ is equivalent to the same configuration arbitrarily translated and rotated, due to what is known as gauge freedom [3]. The relationship between the measured local positions in the i^{th} sensor and the true state is nonlinear and can be expressed as:

$$z_t^i = \pi^i(x_t) + \omega_t = R(\theta^i)\left(Cx_t - p^i\right) + \omega_t \tag{4}$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

extracts the position information from the state vector and ω_t is zero mean, Gaussian measurement noise with covariance $\Sigma_\omega = \sigma_z^2\mathbf{I}$. The likelihood of the measurements is given by:

$$p\left(z_t^i|x_t, \mu^i\right) = \mathcal{N}\left(z_t^i; \pi^i(x_t), \Sigma_\omega\right) \tag{5}$$

Using the fact that all of the measurement noises are independent, the measurements are independent when conditioned on the trajectory and sensor parameters. Therefore, the likelihood function for the entire collection of measurements given the full target trajectory and entire collection of sensor calibration parameters is given by:

$$p(z|x, \mu) = \prod_{t\in\mathcal{M}} p\left(z_t^i|x_t, \mu^i\right) \tag{6}$$

where $\mathcal{M}$ is the set of times for which measurements are available.

III. Initial Solution

The solution proposed in [3] consists of computing the most probable a posteriori trajectory and sensor calibration

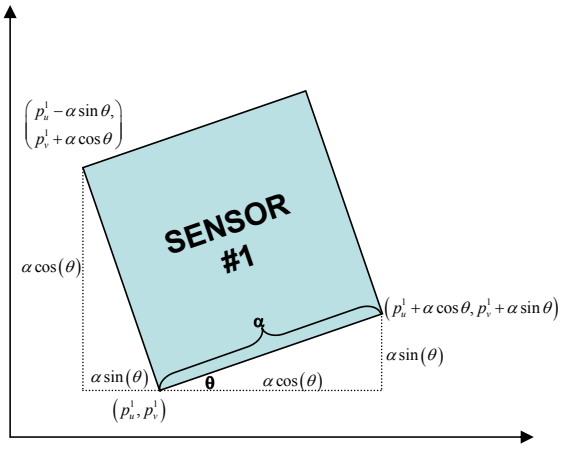


Fig. 2. Sensor geometry.

parameters given the measurement set according to:

$$\begin{aligned}(x^*, \mu^*) &= \arg\max_{x,\mu} p(x, \mu|z) \\ &\propto \arg\max_{x,\mu} p(z|x, \mu)p(x)p(\mu) \end{aligned} \quad (7)$$

Equivalently, one can maximize the $\log$, which is equal to minimizing the arguments of the exponentials for each Gaussian term:

$$\begin{aligned}(x^*, \mu^*) &= \arg\min_{x,\mu} \sum_{t\in\mathcal{Z}} \frac{1}{\sigma_z^2} \left|\left|z_t^i - \pi^i(x_t)\right|\right|^2 \\ &+ x^T\Sigma_x^{-1}x + (\mu-\mu_0)^T\Sigma_\mu^{-1}(\mu-\mu_0) \end{aligned} \quad (8)$$

A Newton-Raphson search algorithm is used to find the optimal x and μ. A vector r can be determined so that Eq. 8 can be rewritten as:

$$(x^*, \mu^*) = \arg\min_{x,\mu} \; r^T r$$

Newton-Raphson is an iterative algorithm of the form:

$$X^{k+1} = X^k - (J^T J)^{-1} J^T r$$

where J is the Jacobian of r with respect to the unknown parameters. The complexity of the Newton-Raphson algorithm is quadratic with respect to the number of rows in J, as it requires a QR decomposition using Householder reflections to solve the least squares problem.

IV. Model Order Reduction

Due to the fact that the MAP solution processes all of the data in batch form, it is extremely important to optimize the performance of the algorithm. The following three techniques seek to reduce the dimension of the search space and to reduce the number of terms in the objective function.

A. Separation Into Multiple Optimization Problems

The first technique removes from the augmented state vector x those entries in the path corresponding to times for which no measurements are available. It can be shown that the estimates of the sensor parameters and remaining path entries produced by this method are exactly the same as those produced by the original technique, as no information is lost by this reformulation [5]. The paths corresponding to missing measurements are computed in parallel to one another after the primary optimization has been solved, using the knowledge of the target's dynamics and the newly formed estimates for the sensor locations and positions of the target as it leaves and returns to the network's field of view.

When the target leaves the range of the sensor network, the target's last known state is propagated through a new motion model that accounts for k-step transitions, and the skipped times are computed separately. The general relationship between the state at time t and the state at time $t+k$ is derived by recursively applying Eq. 1, resulting in:

$$x_{t+k} \quad = \quad A^k x_t + \sum_{i=0}^{k-1} A^i \nu_{t+i}$$

Since all of the ν's are i.i.d. Gaussian random variables with zero mean and covariance Σ_ν, $P(x_{t+k}|x_t)$ is also Gaussian, with mean $A^k x_t$ and covariance equal to:

$$cov\left[x_{t+k}|x_t\right] = \sum_{i=0}^{k-1} cov\left[A^i\nu_{t+i}\right] = \sum_{i=0}^{k-1} A^i\Sigma_\nu\left(A^i\right)^T$$

where the last equality follows due to the fact that the ν_t's are i.i.d. Thus the transition density for the target's state at time t given its state at time $t-k$ is:

$$p\left(x_t|x_{t-k}\right) = \mathcal{N}\left(x_t;\;\; A^k x_{t-k}, \sum_{i=0}^{k-1} A^i\Sigma_\nu\left(A^i\right)^T\right) \quad (9)$$

This multi-step transition density replaces the corresponding $k-1$ single step densities in Eq. 3. Again, a Newton-Raphson search algorithm is used to find the optimal x and μ.

The task of recovering the most probable path traversed by the target while outside the field of view of the sensor network is solved by formulating a new linear least squares problem each time the target exits and returns to the network. The objective is to maximize the probability of this unobserved trajectory between sensors given estimates for the initial and final positions and velocities. The sensor parameters and target trajectory inside the sensors are assumed to be given from the solution to the primary optimization problem. A new vector x is formed by stacking the target's state vectors for all missing measurements between a single exit and return to the network. Let N denote the number of missing measurements corresponding to one segment for which the target is outside the field of view of the sensor network. The new optimization problem becomes:

$$x^* = \arg\max_x \prod_{t=k+1}^{k+N} p\left(x_t|x_{t-1}\right)$$

where x_k and x_{k+N} are treated as fixed quantities. Equiva-

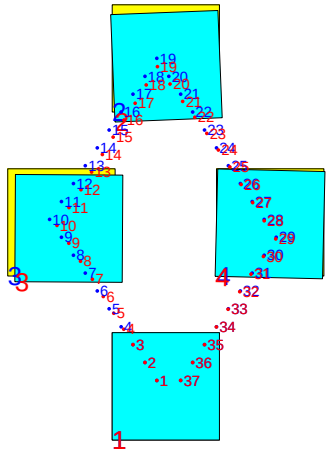
lently, one can maximize the log of the above expression:

$$x^* \quad = \quad \arg\max_x \sum_{t=k+1}^{k+N} \log\left(p\left(x_t|x_{t-1}\right)\right) \qquad (10)$$

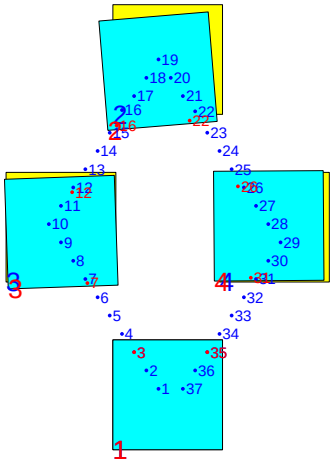
Due to the quadratic objective function, the solution can be computed directly or after a single iteration of a Newton-Raphson search algorithm. It can also be shown that the solution to the above optimization problem gives exactly the same estimates as a Kalman smoother [5]. The forward pass corresponds to propagating the target's state as it leaves the sensor network through the target's motion model, while the backward pass incorporates knowledge of where the target returns to form the smoothed estimates.

B. Using Only Entry and Exit Measurements

Further model reduction is possible by applying the multistep transition density concept within the sensors. Suppose on a given pass though a sensor, six or seven time steps are recorded, as in Fig. 3(a). Each of the measurements contributes information to the position and velocity estimates for the entire target trajectory. However, it is possible to use the position measurements within a single pass through a sensor to get a good local estimate for the velocity at the entry and exit points. Those velocity estimates are then treated like velocity measurements, while all of the interior measurements are discarded. The resulting solution is not equal to the solution using all of the measurements across the entire path, including visits to other sensors; however, in most cases it is a good approximation. The number of measurements within a particular pass through a sensor directly impacts the error of these local velocity estimates. Fig. 3(b) shows the result of the joint calibration and tracking using only the entry and exit velocities. The velocities for a given pass are determined using a Kalman smoother over the set of local measurements.



(a)



(b)

Fig. 3. Example of camera calibration using only entry and exit measurements. The yellow (lighter) squares represent the true sensor locations, while the blue (darker) squares represent the estimated locations. The true path is shown in blue and the estimated path is shown in red. (a) MAP solution using all of the steps. (b) MAP solution using only entry/exit information with local velocity estimate.

C. Treating Measurements as Constraints

In most cases, the measurement noise from an imaging sensor is extremely low relative to other sources of error in the problem. In these cases, it is possible to treat the measurements as nonrandom quantities. The answer reached by this method is not exactly equal to the original solution; however, the difference between the two methods disappears as σ_z approaches zero.

For each measurement in the data set, use Eq. 4 to solve for the corresponding position variables of the path. The new optimization problem becomes:

$$(x^*, \mu^*) = \arg\min_{x,\mu} \; x^T \Sigma_x^{-1} x + \mu^T \Sigma_\mu^{-1} \mu \qquad (11)$$

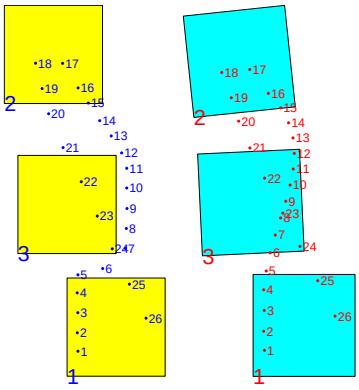
subject to:

$$C\, x_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix} = R^{-1}\left(\theta^i\right) z_k + p^i \quad \forall\, k \in \mathcal{M}$$

where $\mathcal{M}$ is the set of all times for which measurements are available. These equality constraints can then be incorporated into the objective function directly by making the appropriate substitutions within the minimization expression.

V. Missing Measurements

This section addresses the issue of how to fully utilize measurement times to find a sensor configuration and target trajectory that is consistent with this information. The MAP solution may place the target inside the field of view of a sensor at a time for which no measurement is available, as illustrated by the example in Fig. 4. If the target were actually inside the range of that camera at that time, there would have been a corresponding measurement in the data set, thus resulting in a contradiction. Knowledge of the times for which



(a) (b)

Fig. 4. Motivating Example for Missing Measurement Algorithms. (a) True sensor locations and true path traversed by the target. (b) MAP solution. The path segment containing times $\{6, 7, 8, 9, 10, 11, 12\}$ is infeasible, as no measurements occur during these times.

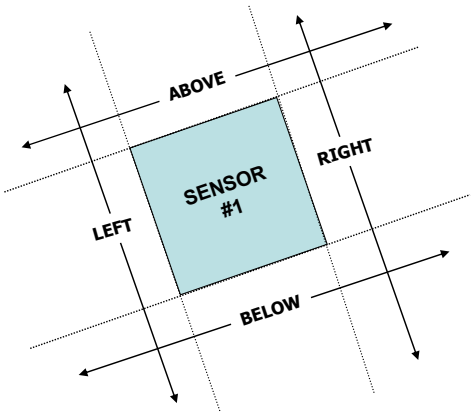


Fig. 5. Feasible regions for the target when no measurement is available.

no measurements are available provide additional constraints to impose on the optimization problem in order to find a feasible solution.

A. Derivation of the Constraint Equations

In order to find a solution consistent with the measurement times, it is necessary to limit the regions in the global map where the target is permitted to be in the absence of measurements. If no measurement is available at time t, the target must be restricted to lie outside the field of view of *each and every* sensor at that time. The target must be either below, above, to the left of, or to the right of each of the sensors, as depicted in Fig. 5. These additional restrictions introduce a set of disjunctive (*either-or*) constraints into the problem.

It is necessary to derive the equations that define the sensor's boundaries in terms of the sensor's internal parameters of translation from the origin, rotation about a reference direction, and size of the field of view in each direction. Note that while these constraints form linear boundaries in the global map, the constraints are nonlinear (and nonconvex) functions of the unknown variables in the overall optimization problem. Figure 2 depicts the geometry of a typical sensor, with all of the quantities labeled that are needed to derive each of the four lines representing the sensor boundaries. The four equations for the lines defining the sensor's walls are given by:

$$\begin{aligned} y_b &= \tan\theta\, x_b + p_v^1 - p_u^1 \tan\theta \\ y_t &= \tan\theta\, x_t + (p_v^1 + \alpha\cos\theta) - \tan\theta(p_u^1 - \alpha\sin\theta) \\ y_l &= -\cot\theta\, x_l + p_v^1 + p_u^1 \cot\theta \\ y_r &= -\cot\theta\, x_r + (p_v^1 + \alpha\sin\theta) + \cot\theta(p_u^1 + \alpha\cos\theta) \end{aligned}$$

When the sensor's rotation angle is $\theta = k\frac{\pi}{2},\ k \in \mathbb{Z}$, the above equations yield slopes of 0 and ∞. In this case, simpler equations arise.

B. Mixed-Integer Nonlinear Programming Formulation

It is possible to determine the solution with lowest cost that is consistent with the missing measurement information by reformulating the problem using a combination of real valued and binary variables. For each missing measurement and each sensor, augment the set of unknowns with a set of four variables $\lambda_i \in \{0, 1\}$, such that:

$$\begin{aligned} v_t &\leq m_b u_t + b_b + \beta\lambda_1 \\ m_t u_t + b_t &\leq v_t + \beta\lambda_2 \\ v_t &\leq m_l u_t + b_l + \beta\lambda_3 \\ m_r u_t + b_r &\leq v_t + \beta\lambda_4 \\ \sum_{i=1}^{4} \lambda_i &= 3 \end{aligned}$$

where β is a *very large* number. The last constraint guarantees that exactly one of the four previous equations holds, while the others are effectively eliminated, since they are automatically satisfied. A set of constraints of the form shown above would need to be added for each missing measurement, for every sensor. If there are M missing measurements and S sensors, this results in MS sets of equations, each with four inequality constraints and one equality constraint.

The task of solving a mixed-integer nonlinear programming problem has been shown to be NP-hard. This is an active research area, and many algorithms have been developed, including the Outer Approximation method [6] and the Branch-and-Bound method [7]. Due to the extremely large number of disjunctive constraints introduced to eliminate the missing measurement infeasibility problem, an approach based on mixed-integer programming is computationally intractable.

C. Adaptive Newton-Raphson Search

This section presents a computationally tractable algorithm to find a feasible solution with respect to missing measurements by adaptively modifying the original Newton-Raphson search. Starting from an initial feasible solution, the largest step possible in the direction of the current gradient is taken, while still remaining feasible with respect to missing

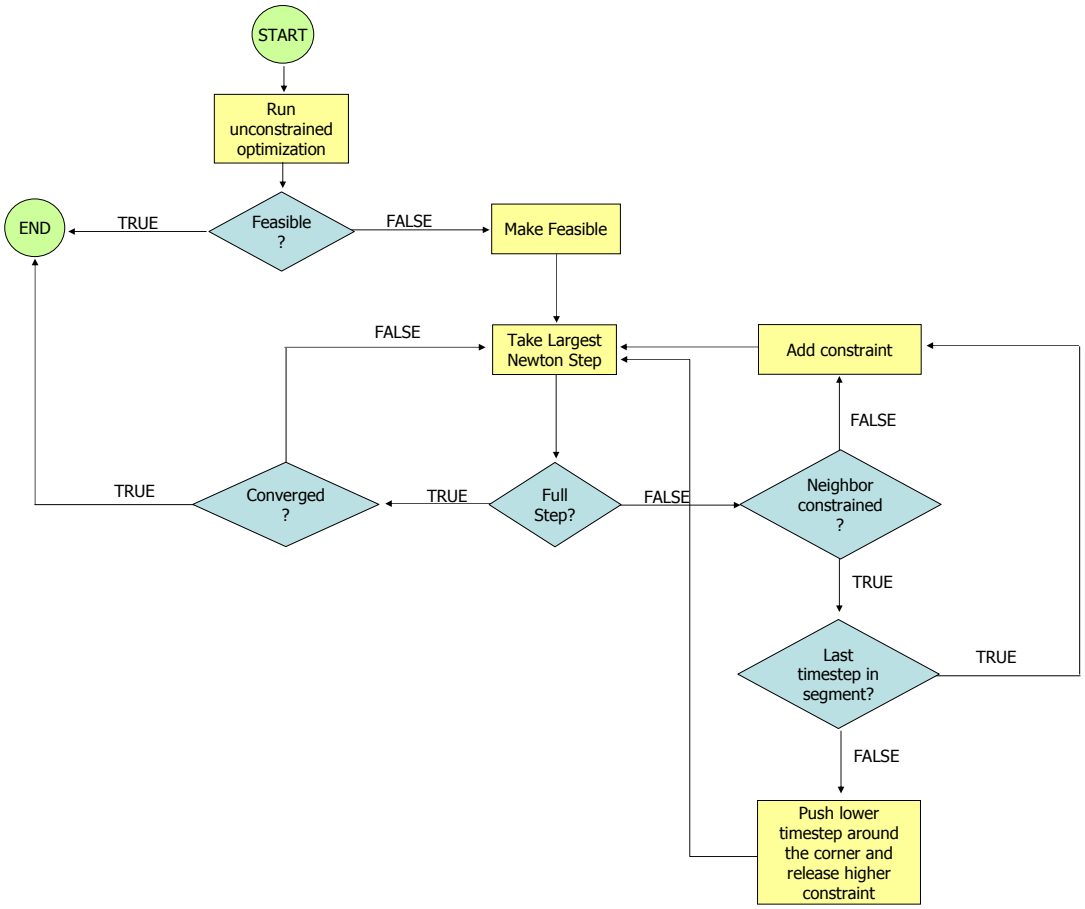


Fig. 6. Modified Newton-Raphson Search: Algorithm

measurements. When a boundary between the feasible and infeasible regions is reached, the new set of active constraints is computed according to a simple set of rules, and the new search direction is computed from this point. In this manner the algorithm explores only the feasible regions of the search space. While the active constraint set changes over time, the overall optimization is computationally tractable, unlike the full blown mixed integer programming problem described in the previous section.

The algorithm behaves according to the flow diagram given in Fig. 6. The *Make Feasible* block moves all of the times for each infeasible segment to a point outside the field of view of all of the sensors, but close to the point of entry for that segment. In the example given by Fig. 4, there is a single infeasible segment consisting of times 6 through 12, and the algorithm would initially move all of these estimates to a feasible location between the top of sensor 1 and the bottom of sensor 3, as shown in Fig. 7(a).

Recall that each Newton-Raphson step is taken as:

$$y^{k+1} = y^k - (J^T J)^{-1} J^T r$$

where $y = \begin{bmatrix} x & \mu \end{bmatrix}^T$. The two points y^{k+1} and y^k in parameter space define a line segment given by their convex combination:

$$(1-\lambda)y^k + \lambda y^{k+1}$$

where $\lambda \in [0,1]$ and $\lambda = 1$ corresponds to taking a full step. Suppose y^k is a feasible solution, but y^{k+1} is not. Then it must be true that for at least one value of λ, the line crosses the boundary between the feasible and infeasible regions of parameter space. Starting from the initial feasible solution, the *Take Largest Newton Step* block computes the largest value of λ so that the next solution remains feasible, as demonstrated in Fig. 7(b).

Since a full step is not possible, the algorithm must determine which constraints to add. The algorithm detects that the target at time 12 is attempting to cross into the field of view of sensor 3. Since there were no previous active constraints, the algorithm simply constrains time 12 to lie on the bottom of sensor 3, and then takes a new step, the result of which is shown in Fig. 7(c). Careful inspection of the plot shows that this newest solution places time 13 on the boundary of sensor 3. The naive choice is to constrain time 13 to be on the top wall of sensor 3. However, the algorithm operates under the guiding principle that if going *through* the sensor is infeasible, then the segment must go *around the corner*. Hence, the algorithm pushes time 12 around the southeast

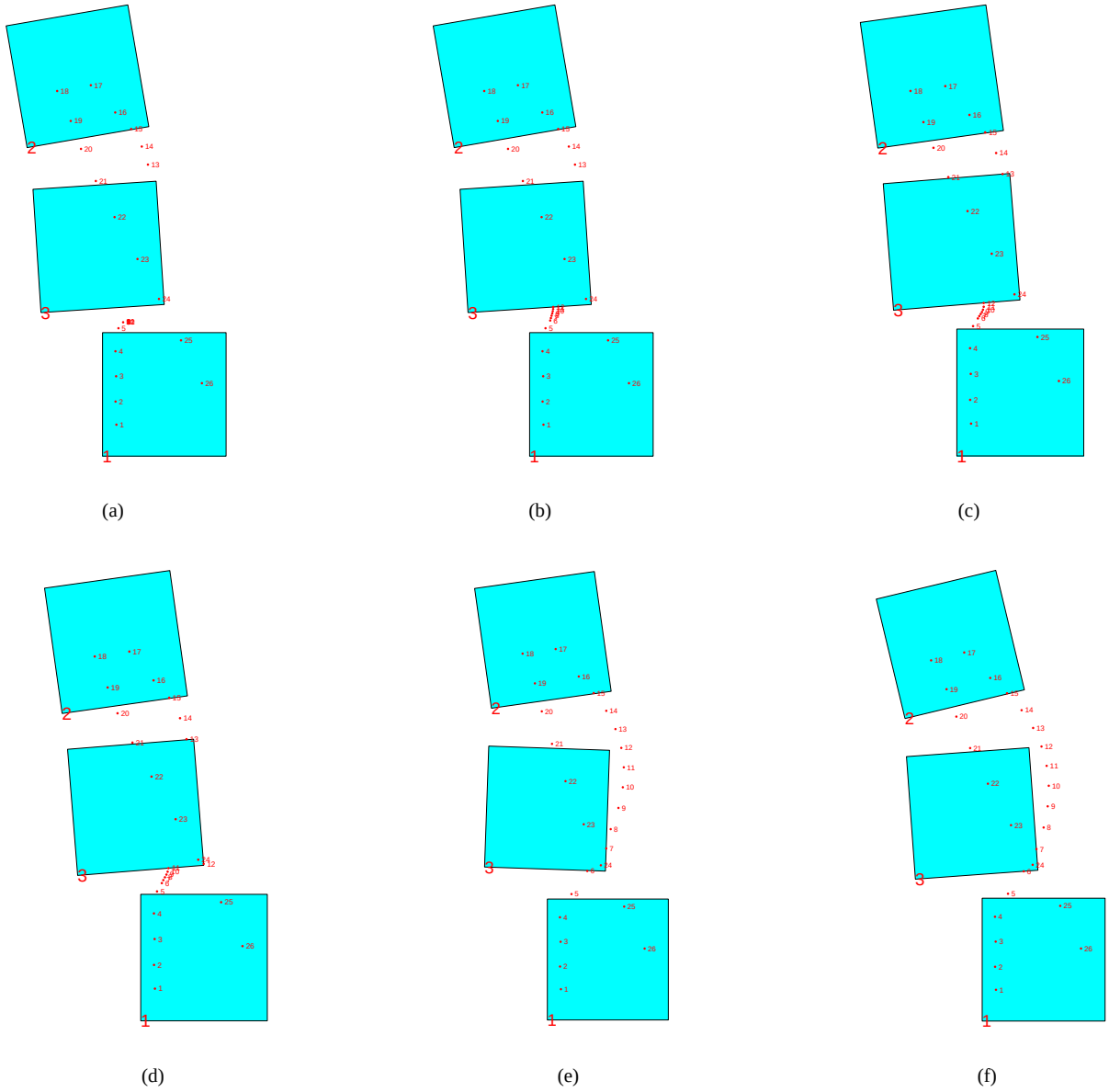


Fig. 7. Modified Newton-Raphson Search: Step-by-Step Example. (a) Initial Feasible Solution. (b) Solution after largest first step taken. (c) Solution after position 12 is constrained to the bottom of sensor 3. (d) Position 12 is pushed around the corner, and 11 is constrained to the bottom of sensor 3. (e) Position 6 approaches the bottom of sensor 3. (f) Final feasible solution. Position 6 is constrained to the bottom of sensor 3, and 7 is constrained to the right side.

corner of the sensor, releasing the constraint on the bottom wall, and adding a constraint to the right wall. The result of this operation is given by Fig. 7(d). At this point time 11 approaches the bottom boundary of sensor 3. The same principle of pushing the path around the corner is applied, and time 11 is constrained to the right wall of sensor 3, while the constraint for time 12 is released. The process is repeated until the last time in the segment attempts to enter the sensor, as shown in Fig. 7(e). The last time is not pushed around the corner, but rather it is constrained to the bottom wall, without releasing the constraint for time 7. The final result is shown in Fig. 7(f). This adaptive Newton search algorithm is not guaranteed to find the constrained solution with minimum cost over the set of all possible constrained solutions. However, it successfully finds a feasible solution, and it does this in a automatic and systematic manner.

D. Circular Constraints

The constraints derived in section V-A exactly define the square sensor boundaries within the global map. In order to fully describe the region outside the field of view of each and every sensor, it is necessary to use disjunctive constraints. This requires the computation of many parallel optimization problems, resulting in a procedure that is computationally intractable. Another approach to the problem is to use circular constraints to approximate the sensor boundaries, as shown in

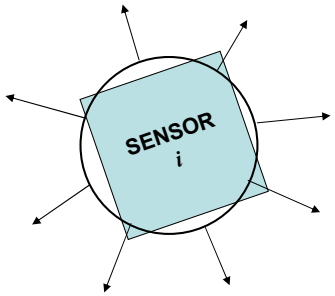
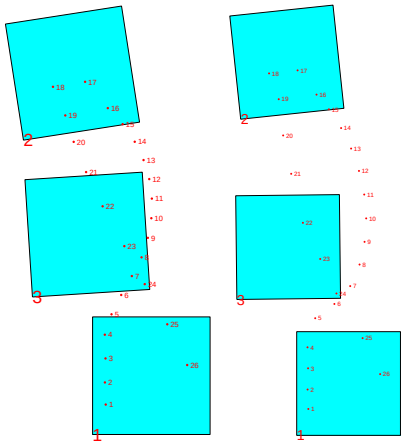


Fig. 8. Use of circular constraints to approximate square sensor boundaries.



(a) (b)

Fig. 9. Example using circular constraints. (a) Circles inscribed within the square sensors. (b) Circles circumscribed around the square sensors.

Fig. 8. This results in a single, although highly constrained, optimization problem to solve.

The region outside the field of view of the sensor can be described in a single equation through the use of circular constraints. The center coordinate for each sensor is given by:

$$(u_c, v_c)^T = \begin{pmatrix} 0.5\,(2p_u + \alpha\cos(\theta) - \alpha\sin(\theta)) \\ 0.5\,(2p_v + \alpha\cos(\theta) + \alpha\sin(\theta)) \end{pmatrix}$$

and thus each circle can be expressed as:

$$r^2 \leq (u_t - u_c)^2 + (v_t - v_c)^2$$

where r is the radius of the circle. A constraint of this form is required for each sensor and for each time during which no measurement is available. Let S denote the number of sensors and let U denote the number of missing measurements. Thus there are SU simultaneous inequality constraints, rather than the set of *either-or* constraints that arise from the use of square sensor boundaries. This technique trades off the ability to exactly describe the regions of the global map where the target is permitted to be when no measurements are available with the ability to pose the problem as a single nonlinear optimization problem. As a result, the solution found using this method may not be the solution with lowest overall cost.

The question arises how to pick the radius r of the circle that best approximates the sensor boundary. The circle inscribed in the square may result in a solution that places the target within the corners of the field of view of the actual sensor, as shown in Fig. 9(a). The circle circumscribing the square may push the target further away from the sensor than necessary, resulting in a solution with larger cost that the true globally optimal solution, as shown in Fig. 9(b). One possible approach is to systematically vary the radius to find the smallest r needed in order to push the entire infeasible path segment just beyond the field of view of each sensor.

VI. Conclusion

Previous work has shown that it is possible to perform simultaneous target path estimation and calibration using MAP estimation. MAP estimation is not thought to be computable in real time, as it processes all of the data in batch form. Recursive algorithms such as the Kalman or particle filter require smoothing to correct the previous track estimates given new data that is available later. The MAP estimator presented here is an inherent smoother, however, the algorithm quickly becomes computationally intractable as the number of measurements increases. This paper demonstrated three improvements to reduce that computational complexity, making real time MAP estimation possible.

This paper also presented three methods to improve the accuracy of the MAP solution by making use of the missing measurement information. Additional inequality constraints were placed within the optimization framework in order to prevent the target from entering the field of view of the sensor network at times for which no measurements were available.

Acknowledgment

This research was supported in part by participation in the Georgia Institute of Technology MURI 2004 sponsored by Army Research Office (ARO) under award W911NF-04-1-0190 and was supported in part by participation in the Advanced Sensors Collaborative Technology Alliance (CTA) sponsored by the U.S. Army Research Laboratory under Cooperative agreement DAAD19-01-2-008.

References

[1] Cevher, Volkan. A Bayesian Framework for Target Tracking using Acoustic and Image Measurements. PhD thesis. School of Electrical and Computer Engineering, Georgia Institute of Technology, 2005.

[2] Moses, Randolph, Dushyanth Krishnamurthy, and Robert Patterson. An Auto-Calibration Method for Unattended Ground Sensors. In ICASSP, 2002.

[3] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous Calibration and Tracking with a Network of Non-Overlapping Sensors. In CVPR, June 2004.

[4] Chui, C. K. and G. Chen. Kalman Filtering with Real Time Applications. Springer-Verlag, New York, 1987.

[5] M. Rudoy. Simultaneous Camera Calibration and Path Estimation. Masters thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2006.

[6] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. 36, 3 (Dec. 1986), 307-339.

[7] E. M. L. Beale. “Integer Programming”, in The State of the Art in Numerical Analysis (D. Jacobs, ed.) Academic Press: London, 409-448.