

Collision-Free State Estimation

Lawson L.S. Wong, Leslie Pack Kaelbling, and Tomás Lozano-Pérez

Abstract—In state estimation, we often want the maximum likelihood estimate of the current state. For the commonly used joint multivariate Gaussian distribution over the state space, this can be efficiently found using a Kalman filter. However, in complex environments the state space is often highly constrained. For example, for objects within a refrigerator, they cannot interpenetrate each other or the refrigerator walls. The multivariate Gaussian is unconstrained over the state space and cannot incorporate these constraints. In particular, the state estimate returned by the unconstrained distribution may itself be infeasible. Instead, we solve a related constrained optimization problem to find a good feasible state estimate. We illustrate this for estimating collision-free configurations for objects resting stably on a 2-D surface, and demonstrate its utility in a real robot perception domain.

I. INTRODUCTION

As robots interact with increasingly complex environments, the problem of state estimation, or filtering, becomes much more complex. Consider the problem of a mobile-manipulation robot, attempting to localize objects inside a refrigerator before planning to pick one up. The state of the problem is the positions and orientations of the objects within the refrigerator; we need a representation of distributions over states, which can be used for recursive estimation. Both the mean and the (co)variance of the estimate will be necessary for selecting motion and sensing actions.

It has been traditional in many localization problems to use a joint multivariate Gaussian distribution over the state space: it is compact and mathematically and computationally convenient, but in our problem it is inappropriate, even as an approximation, due to inherent physical constraints on the configuration of multiple objects. The multivariate Gaussian has infinite support, yet the space of object configurations is highly constrained. Objects cannot overlap with other objects in the refrigerator, or with the refrigerator walls.

In this paper, we explore the use of constrained distributions similar to truncated multivariate Gaussian distributions for filtering the joint poses of multiple nearby physical objects. A truncated Gaussian is composed of a non-truncated Gaussian distribution over the underlying space together with lower and upper bounds on the support. In the resulting distribution, points not in the domain have zero probability, and the Gaussian density is renormalized so that it integrates to one between the bounds. Our problem is more complicated

because the allowed domain in general will have a complex joint constraint structure instead of simple bounds. In the case of objects in the refrigerator, the constraints are used to rule out any configuration of the objects that would require them to interpenetrate each other or the refrigerator walls.

Given this representation, the next question is, how we can use it for estimation. Intuitively, it seems that it might be difficult to maintain the state of a filter with such constraints on the state. Fortunately, it has been shown that a very simple approach works well: perform recursive state estimation in the unconstrained space, and then when it is necessary to generate a maximum *a posteriori* (MAP) or mean estimate, solve a constrained optimization problem to find the maximum of the constrained posterior.

In the following, we briefly discuss related work, illustrate the problem of estimation in a very simple domain with generalization to other domains, and then demonstrate its utility in a real robot perception domain.

A. Related Work

There has been considerable interest in general constrained state estimation, particularly in the context of Kalman filtering ([1], [2], [3], [4]). These works mainly focus on linear state equalities and inequalities; nonlinear constraints have been considered with linearization techniques. A survey of such techniques is given by Simon [5]. The main result from many of these works is that constrained state estimation can be performed reasonably well by filtering as in the unconstrained case, and projecting the solution back to the state constraint surface. We are therefore concerned with how to perform this projection step. In previous work, constraints in practice have been upper/lower bounds on various physical state quantities, for example aircraft engine health [6], odometry [7], road boundaries when tracking a car [8], acceleration of a moving object [9]. Our current problem constrains object configurations to be collision-free. To our knowledge, this type of constraint has not been previously considered within the context of Kalman filtering.

Particle filters provide a flexible alternative for tracking complex, non-Gaussian states. However, they suffer from the curse of dimensionality and become computationally intractable in high dimensions. To mitigate this, Grundmann et al. [10] used a particle filter that first updated object states separately, then evaluated the joint estimates using a rule set. The rule set encoded physical constraints such as non-interpenetration and gravity, and was evaluated using a physics simulator. States that violate physical constraints become highly unlikely in the posterior and are effectively eliminated from the particle representation. However, as we

This work was supported in part by the NSF under Grant No. 019868, in part by ONR MURI grant N00014-09-1-1051, in part by AFOSR grant AOARD-104135, and in part by the Singapore Ministry of Education under a grant to the Singapore-MIT International Design Center.

Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge, MA 02139
{lsw,lpk,tlp}@csail.mit.edu

will examine in section V, particle filtering may not produce good estimates of the maximum likelihood estimate.

The modeling of joint state dependencies has been of particular interest to the multi-target tracking and data association community, where explicitly modeling correlations and occlusions is useful. Recent methods such as [11] and [12] typically derive a complex measurement model, then use particle methods to handle the otherwise intractable filter. Much of the complexity in these techniques however is devoted to data association problem, which we are not concerned with in this work. The approach we explore is different in that our measurement model is simple (Gaussian), yet complex dependencies are supported using constraints.

II. MODEL AND NOTATION

We assume that there are M three-dimensional objects with known shape models resting stably on a 2-D surface. Each object is indexed by j and is characterized by its state parameters $s_j = \{x_j, y_j, \theta_j\}$. The collection of all object states $\{s_j\}_{j=1}^M$ will be denoted by \mathbf{s} . The state parameters \mathbf{s} are subject to boundary and collision constraints. We assume a boundary box constraint: that the entirety of each object (not just the centroid) is located within the space $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$. For collision constraints, two objects i, j are in collision if the shape models parametrized by their states s_i, s_j have non-zero volume in their intersection. We assume that we have sufficient shape-model knowledge to compute this. The subset of the joint parameter space for which these constraints are satisfied will be denoted the feasible region, and its complement the infeasible region. Our prior for \mathbf{s} is uniform over the feasible region.

We obtain N_j observations $\{s_j^{(i)}\}_{i=1}^{N_j}$ of s_j with zero-mean Gaussian noise.¹ The covariance matrix Σ of the noise is unknown, but can be estimated by the sample covariance matrix of the observations, or with an extended or unscented Kalman filter. For simplicity of exposition, we assume that there are no inter-object dependencies in the noise model. Specifically, the covariance matrix Σ is block diagonal, and we denote the object-specific covariances by $\{\Sigma_j\}_{j=1}^M$. Generalization to arbitrary covariance matrices is straightforward. Note that unlike our prior, this noise distribution is unconstrained, i.e., an observed $s^{(i)}$ can lie in the infeasible region. This may occur, for example, if the true \mathbf{s} is close to the constraint boundary and observations are noisy. The joint distribution on the observed data, parametrized by the true state \mathbf{s} , is:

$$\mathbb{P}\left(\left\{\left\{s_j^{(i)}\right\}_{i=1}^{N_j}\right\}_{j=1}^M; \mathbf{s}\right) \propto \prod_{j=1}^M \prod_{i=1}^{N_j} \exp\left\{-\frac{1}{2}\left(s_j^{(i)} - s_j\right)' \Sigma_j^{-1} \left(s_j^{(i)} - s_j\right)\right\} \quad (1)$$

where s' denotes the transpose of s .

Our objective is to get a MAP estimate $\hat{\mathbf{s}}$ of the parameters. If our parameter space was unconstrained, this corresponds to the sample mean of the observations. However, because

¹Technically it is more accurate to use a circular distribution for θ , such as the von Mises distribution. We use a wrapped Gaussian distribution for simplicity, noting that the quadratic term in the Gaussian exponent corresponds to the second-order Taylor expansion of the log von Mises distribution, and hence a Gaussian serves as a good approximation.

our prior places zero density in the infeasible region, the posterior density in this region will also be zero. When we compose the Gaussian data likelihood given by Equation 1 with the uniform prior over the feasible region, the posterior density will therefore have Gaussian shape over the feasible region, and zero otherwise. This posterior distribution is similar to a multivariate truncated Gaussian distribution [13], in the sense that it has Gaussian shape over some constrained region. However, truncated Gaussians only have upper and lower bounds on their parameters, whereas in our case the collision constraints may create ‘holes’ in the feasible region. In particular, this shows that our feasible region is generally not a convex set, a point we will return to later.

III. A SIMPLE CASE WITH CYLINDERS

We illustrate the solution method first for the simple case of objects that are cylinders with known radii $\{r_j\}_{j=1}^M$. Using cylinders allows us to ignore the orientation parameter; for this section we assume that the state vectors are $s_j = \{x_j, y_j\}$. Because cylinders have uniform cross-section, we can restrict our attention to a 2-D slice of the original problem. The collision-free constraint reduces to a distance constraint (where object centers must be at least some distance apart), namely $\|s_i - s_j\|_2 \geq (r_i + r_j)$.

In section II we showed that the posterior is proportional to the observed data likelihood given by Equation 1 when \mathbf{s} is in the feasible region, and zero otherwise. We can find the MAP estimate by maximizing the log-posterior. For \mathbf{s} in the infeasible region, the log-posterior is undefined, and cannot be a maximizer. For \mathbf{s} in the feasible region, it is given by:

$$\ell(\mathbf{s}) = \text{const} - \sum_{j=1}^M N_j \left[\frac{1}{2} s_j' \Sigma_j^{-1} s_j - \bar{s}_j' \Sigma_j^{-1} s_j \right] \quad (2)$$

where we have rewritten the inner sum in terms of the sample means $\bar{\mathbf{s}} = \{\bar{s}_j\}_{j=1}^M$ and removed constants in the exponent. This leads to the following constrained minimization problem, where we explicitly restrict $\hat{\mathbf{s}}$ to be in the feasible region:

$$\hat{\mathbf{s}} = \arg \min_{\{s_j\}_{j=1}^M} \sum_{j=1}^M N_j \left[\frac{1}{2} s_j' \Sigma_j^{-1} s_j - \bar{s}_j' \Sigma_j^{-1} s_j \right] \quad (3)$$

$$\begin{aligned} \text{subject to } & x_{\min} + r_j \leq x_j \leq x_{\max} - r_j & 1 \leq j \leq M \\ & y_{\min} + r_j \leq y_j \leq y_{\max} - r_j & 1 \leq j \leq M \\ & \|s_i - s_j\|_2 \geq (r_i + r_j) & 1 \leq i, j \leq M \end{aligned}$$

Without the final constraint, the optimization problem is a standard quadratic programming (QP) problem. However, the final constraint, expressed equivalently as a quadratic by squaring both sides, makes the problem a (non-convex) quadratically-constrained QP. The non-convexity arises due to ordering issues; if we consider the 1-D analog of the parameter constraint, $|x_i - x_j| \geq r$, x_i may be either to the left or right of x_j , but the solver will have to check both cases. This fragmentation occurs for each parameter pair in the constraints, creating a difficult combinatorial search space for global optimization procedures.

If we assume that the initial values (state estimates) we provide to the solver are sufficiently close to the true state,

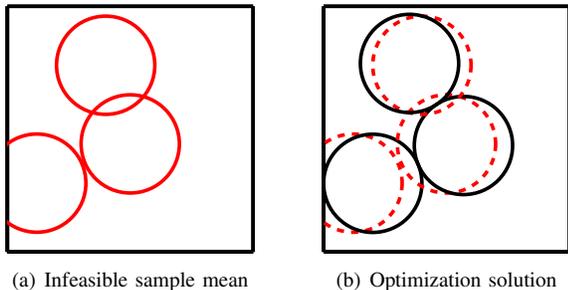


Fig. 1. Demonstration of the simple case of circular objects with a synthetic example. The sample mean in (a) corresponds to a configuration that is infeasible, where two objects are in collision and one is out of bounds. The optimization solution is shown in (b), superimposed on the original infeasible estimate (dashed red).

we may be satisfied with local minima. Because the non-convexity of the problem is related to the spatial ordering of the objects, if we assume that the initial estimates satisfy the correct ordering, then we are likely to find a local minimum that is in fact the global one. Given reasonable observations, we expect the sample means \bar{s} to obtain the correct ordering, so using \bar{s} as initial values will likely give satisfactory solutions. Starting from \bar{s} , we solve the above problem with sequential quadratic programming (SQP), which linearizes the collision constraints, solves the subsequent QP, and iterates until a feasible local minimum is found.

We use a synthetic example shown in figure 1 to highlight some characteristics of the optimization problem. In this example, we set the observation noise to be significantly greater in the horizontal direction. Specifically, we set each Σ_j to be diagonal, with variance in x_j to be 10 times that of y_j . Suppose we collected observations with sample means \bar{s} corresponding to the configuration in the left panel, where two objects are in collision and one is out of bounds. The right figure shows the infeasible configuration (dashed red) and the optimization solution (solid black). The previously violated constraints are now satisfied and tight. We also see that the objects were corrected mostly in the horizontal direction, the direction of greater variance. Recall that our optimization objective is closely related to the (squared) Mahalanobis distance between the parameter estimate and the sample mean, i.e., a squared difference weighted by the inverse covariance matrix Σ^{-1} . Hence parameters with larger precisions (smaller variances) will be relatively more important in the objective, and deviations from the sample mean will be penalized more. The relative sizes of variances therefore naturally trade off correction costs. In our current example, the greater horizontal variance led to a correction that mostly shifted in the horizontal direction.

IV. GENERAL COLLISION-FREE STATE ESTIMATION

Returning to the original problem ($s_j = \{x_j, y_j, \theta_j\}$), we can formulate a similar optimization problem:

$$\hat{s} = \arg \min_{\{s_j\}_{j=1}^M} \sum_{j=1}^M N_j \left[\frac{1}{2} s_j' \Sigma_j^{-1} s_j - \bar{s}_j' \Sigma_j^{-1} s_j \right] \quad (4)$$

$$\begin{aligned} \text{subject to } & \text{shape}(j) \in [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}], 1 \leq j \leq M \\ & \text{collision}(i, j) == \text{FALSE}, \quad 1 \leq i, j \leq M \end{aligned}$$

For general object shapes, the constraints become difficult to express. For example, if all shapes are polygons given by a finite list of vertices, checking the boundary constraint for an object requires computing all its vertices given its state, and checking that all of them lie within the given range. Collision constraints will generally require geometric collision checks to verify. When assuming we have known shape models, we therefore mean that given a potential pose for an object, we can efficiently compute whether it satisfies the boundary constraints (e.g., by computing its vertices). We also assume that these shape models can be given to a black-box collision checker which checks the collision constraints.

In practice, the above formulation is difficult for iterative optimization techniques such as SQP. Such techniques iteratively perform a quadratic approximation to the Lagrangian, which in turn requires first- and second-derivative information from the constraints. Derivatives inform algorithms of the direction to optimize along. Although such information is not always analytically available, finite difference methods can approximate the derivatives. However, for the binary constraint functions given above, there is no useful gradient information, because local neighborhoods are either flat or step functions. To fix this, we transform boundary constraints to the difference between each (x_j, y_j) and each boundary line (assuming the boundary is a convex region). These linear differences reflect how far each variable is from its constraint boundary. We approximate the degree of constraint violation with the volume of collision (intersection) between two objects when they collide, and zero otherwise. The SQP solver can use this to approximate changes in constraint violation and navigate to feasible solution regions.

SQP is an iterative optimization algorithm that has to explicitly check the constraints and approximate their gradient at some given candidate solution s' during each iteration. Collision checking is generally slower compared to the other components of the optimization, and checking all $\binom{M}{2}$ object pairs can be undesirable, especially if M is large. We use a simple heuristic that ignores all object pairs that are guaranteed to be collision-free at the current s' . For example, if we can enclose objects i and j with balls of radii r_i and r_j respectively, and the distance between their centers exceeds $(r_i + r_j)$, then clearly they do not collide. Checking this is more efficient than calling an (external) collision checker. Under the mild assumption that for each object, all but a small constant number of objects are sufficiently far away to be ignored by this heuristic, the number of collision checks needed grows linearly in M instead of quadratically, and empirically provides significant computational improvements.

V. RESULTS

We demonstrate this method in robot perception problems where scenes may be cluttered and objects may be placed very close together, as one may expect to find in a common household environment. Using several simple objects, we created scenes such as those in figure 3 where the objects are near each other (gaps \approx 1cm). The objects have uniform cross-section so the collision-checking can

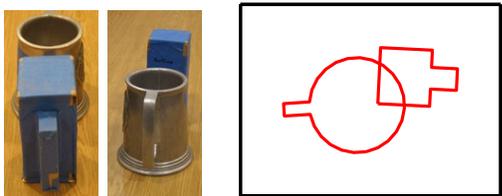


Fig. 2. In our experimental setup, objects are placed near each other on a table, and the PR2 estimated their poses from multiple viewpoints. When objects are consistently occluded in viewpoints, such as in the above case where the two objects cannot be recognized together from the same viewpoint, infeasible joint state estimates may arise.

be done efficiently in 2-D, and are modeled with simple polygonal shapes. We used a Willow Garage PR2 robot platform to perform perception tasks, and for pose estimation we used the method in [14] (using stereo camera data only). To solve the optimization problem (4), the SQP solver in the MATLAB Optimization Toolbox was used. Collision checking between object pairs was performed by computing the cross-sectional area of overlap.

There are many situations in which robots do not simply statically perceive scenes; they may receive observations from different viewpoints. For example, when working in a home, robots will likely have to move around both during and between tasks. To simulate some of these effects, we moved the PR2 around the table on which the objects were placed and collected observations from different angles, as shown in figure 2. At each new viewpoint, we adjusted the PR2 cameras such that the objects were always near the middle of the perceived point clouds to maximize detection performance. The predicted object poses were then transformed to a fixed reference frame in which we performed state estimation.

Two sources of error are predominant within our experimental setup. On each observation, we estimate the object poses from the point cloud. Pose estimation, in particular the orientation, is difficult for many non-trivial objects. Errors introduced in this process are inherited in our setup. As shown in Table I, the variance in orientation is sometimes large and may even fluctuate depending on the scene (compare the two LBlock variances). A variance ≈ 0.3 corresponds to a θ standard deviation of about half a radian, or 30° . Moving the PR2 around can be beneficial for perception because objects may be occluded at some viewpoints or may be difficult to perceive in some orientations. On the other hand, moving around can introduce odometry errors, which have magnitudes at least matching the inter-object gaps. Because of this, odometry errors can have adverse effects on our task, especially if certain objects are consistently occluded or too far away (see figure 2). We will see both types of errors come into play in our experiments.

Under the unconstrained Gaussian observation model, the sample mean of the parameters will always be the mode of the distribution and hence the maximum likelihood estimate. In the second column of figure 3, we show scenarios where this naïve estimate results in an infeasible object configuration. We provide the sample mean and covariance matrix to our optimization procedure, which solves for a feasible collision-free configuration that (locally) minimizes

TABLE I
PARAMETER CORRECTIONS AND VARIANCES FOR FIGURE 3 SCENARIOS

Object	Parameter	Correction	Variance
RectCup	x (m)	-0.0043	0.0024
	y (m)	0.0002	0.0019
	θ (rad)	0.1684	1.2119
LBlock	x (m)	0.0016	0.0011
	y (m)	-0.0003	0.0027
	θ (rad)	0.0010	0.0067
Mug	x (m)	-0.0014	0.0000
	y (m)	0.0000	0.0000
	θ (rad)	0.0000	0.0006
RectCup	x (m)	0.0289	0.0001
	y (m)	0.0016	0.0001
	θ (rad)	-0.0001	0.0309
Rectangle	x (m)	-0.0022	0.0004
	y (m)	0.0095	0.0022
	θ (rad)	0.0621	0.3187
LBlock	x (m)	-0.0015	0.0028
	y (m)	0.0018	0.0042
	θ (rad)	-0.0924	0.4233
Mug	x (m)	0.0025	0.0004
	y (m)	-0.0072	0.0015
	θ (rad)	-0.0004	0.0233

the quadratic objective from the previous section, shown in the third column of figure 3.

From the superimposed estimates in the fourth column of the figure, we can identify some qualitative features of each scenario. For example, in the first row, we see that the L-shaped LBlock stays close to its sample mean, whereas the RectCup is rotated away with a small translation. In the second row, we might have expected the same thing to happen for the Mug; however, instead of rotating away like the RectCup, a translation was performed instead. Looking at the variances of θ in Table I for these two objects explains this qualitative difference. The θ parameter for RectCup has a very high variance compared to other parameters in its estimation problem, hence its rotational correction is favored. RectCup was a difficult object for our pose estimator. In contrast, Mug was very easy to estimate with its larger handle and unique shape, hence has a significantly lower θ variance. In light of this, translation was ultimately preferred.

Errors from odometry can also lead to infeasible estimates. For the scenario shown in figure 2 and the second row of figure 3, we limited the robot to two viewpoints, aligned along the axis between the two objects, so that one object was always occluded by the other. For example, when viewed from the left of the image, only the circular Mug was visible. At each viewpoint, the pose estimate was relatively accurate, as shown in Table I. However, because each viewpoint only gave an estimate for one object, the odometry error incurred when moving between the two viewpoints could not be removed based on the observations. The error in the objects' position is apparent for this scenario. Although our solution did not recover the original configuration, it is at least feasible and closer to the actual state.

We also compared our optimization approach to a baseline method. Since we assumed a joint Gaussian observation model, we can in principle perform rejection sampling on the constrained parameter space, which has the same Gaussian shape in the feasible regions. Specifically, we estimate

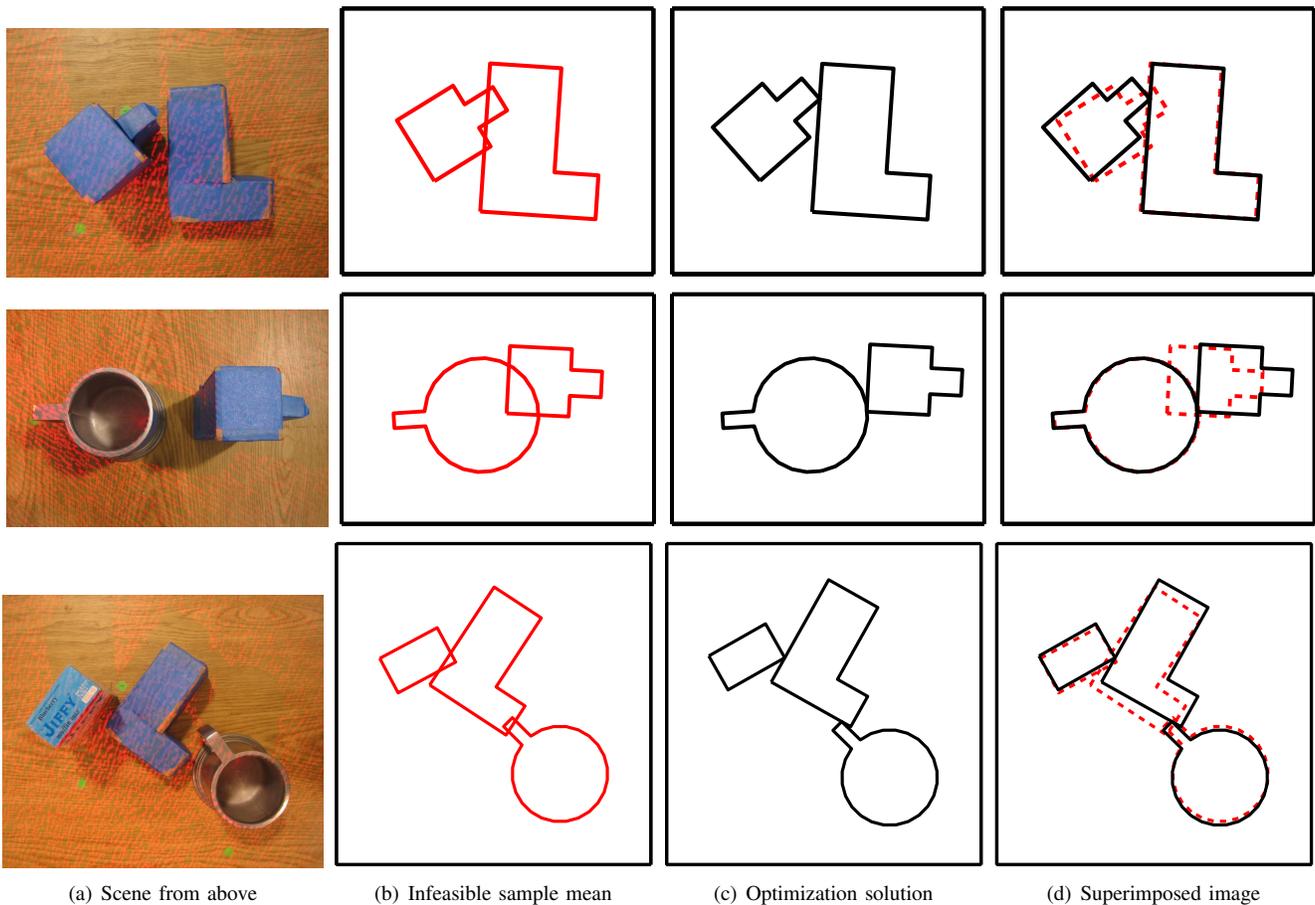


Fig. 3. Scenarios where parameter sample means from pose estimates corresponded to infeasible configurations, shown in column (b). The optimization solution estimates in column (c) are feasible, and previously violated constraints are now satisfied tightly. The superimposed images in column (d) illustrate that large corrections are usually applied in parameters with high variance, as given in Table I. For reference, the true object configurations are shown in column (a). Note that this view is not available to the robot; in practice, the PR2 receives angled, occluded views such as those in figure 2. Such viewpoints make it difficult to get a joint state estimate of all objects. Instead, individual object estimates from multiple viewpoints must be combined with odometry information to obtain a joint estimate. This process introduces additional error and uncertainty, and potentially leads to infeasible joint state estimates.

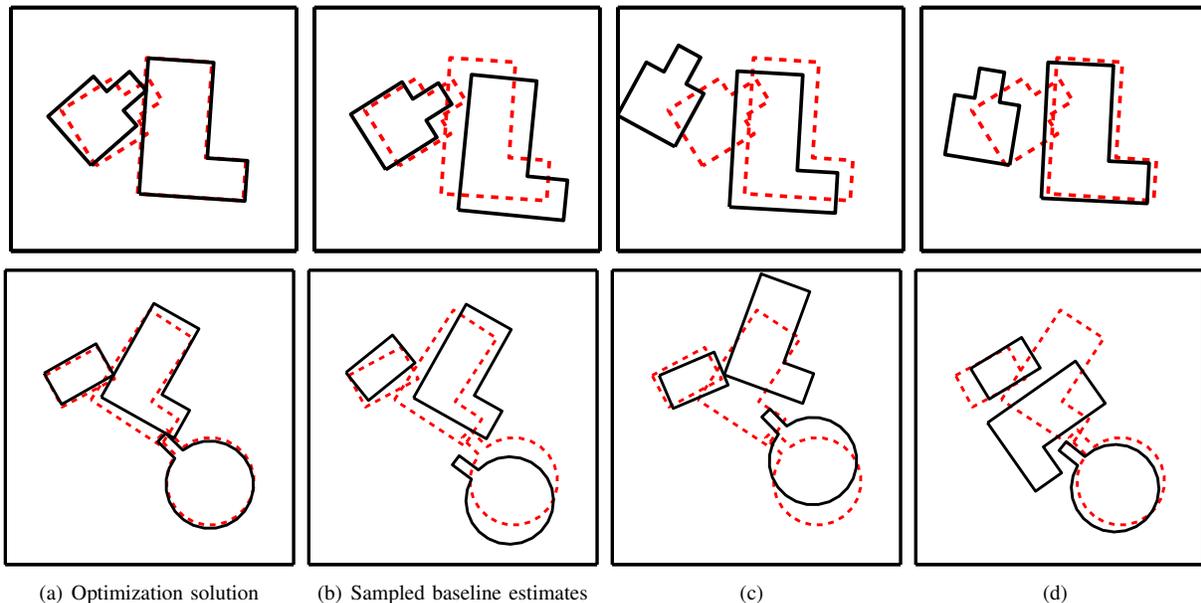


Fig. 4. Random samples from the baseline method described in section V for two scenarios from figure 3. Column (a) is the optimization solution (solid black), superimposed on the sample mean configuration (dashed red). Columns (b)-(d) show baseline estimates for each scenario from three independent trials. Each estimate had the lowest objective value among the ≈ 500 random samples generated for each trial.

the observation model parameters with the sample mean and covariance as before, but instead of performing local optimization, we simply generate samples from this joint Gaussian distribution and accept them if they are feasible. This is similar to what a naïve particle filter would do, where the feasibility check performs a ‘hard’ version of the physics simulation-based rule set in [10]. We then use the sample that has the greatest likelihood as the revised state estimate (i.e., it performs the best out of all samples on the optimization problem). Given sufficient samples, sampling can in principle recover the estimate returned by optimization. Moreover, because sampling is not affected by the non-convexity of the feasible parameter space, it may even return a superior estimate (although this did not happen in our trials). To provide a fair comparison, we allowed the sampling scheme the same amount of computation time the optimization procedure took (this includes the constraint checking for the rejection step, since this is the most time-consuming). In practice, this allowed about 500 samples, of which the acceptance ratio was usually between $\frac{1}{3}$ and $\frac{1}{2}$.

Figure 4 show the best samples from three independent trials of this baseline scheme on two of the scenarios from figure 3. As expected, they are collision-free since this was a requirement for acceptance, and the resulting configurations are generally close to the original infeasible estimates. However, there are two important features to note. First, the baseline estimates have high variance and are not consistent across multiple trials, whereas the optimization approach gives a unique local minimum from the same initial estimate. Second, the samples do not satisfy violated constraints tightly; we see that the objects are not touching, unlike the optimization solution. Although this appears fine in the two-object scenario, in the three-object one, where the LBlock is tightly squeezed between two other objects, sampling has difficulty achieving this property. The LBlock escapes to the top in the first two estimates and slips below in the third. In contrast, the optimization solution satisfies active constraints tightly and fits the LBlock in the gap. Because the baseline scheme is similar to particle filtering as mentioned above, particle filters will likely encounter similar issues when finding the maximum likelihood state estimate.

VI. DISCUSSION AND FUTURE WORK

So far we have only considered objects that have uniform cross-section, which allowed us to reduce the collision-checking problem to a 2-D one, where the collision volume is the area of overlap between two shapes. For 3-D shapes, the generalization is straightforward, where we use 3-D volume instead of area. For objects with complicated shape models this may be difficult to compute. One compromise is to compute the 2-D area of overlap between cross-sections of objects at various discretized heights. Their sum, for example, can then be used as an approximation to volume.

Despite the straightforward generalization of collision volume to 3-D objects, there may be better alternatives to smoothly approximate the collision constraint. We argued that smooth approximations are necessary because SQP

needs to estimate derivatives of the constraints to navigate to the feasible region. Collision volume is positive and smooth when objects collide, but is zero and flat for objects not in collision. Although informative derivatives are more essential for the infeasible region (to escape and avoid such spaces), empirically we found solution methods to be more efficient when feasible region derivatives are informative as well. A possible substitute for the collision volume is the collision depth (when in collision), which smoothly transitions to the closest distance between two objects (when not in collision).

VII. CONCLUSIONS

We have developed a constrained optimization approach to solve the state estimation problem for object configurations with boundary and collision constraints. Optimization allows us to find good feasible maximum *a posteriori* state estimates in a principled fashion. We demonstrated the utility of this approach on real robot perception scenarios, where object configurations corresponding to the sample mean of the observed state were infeasible. Using our approach we recovered close feasible state estimates that were superior to a baseline sampling scheme.

The method developed in this paper can be used in a broad variety of semantic mapping and object manipulation tasks, providing an efficient and effective way to incorporate collision constraints into a recursive state estimator, obtaining optimal or near-optimal solutions.

REFERENCES

- [1] D. Simon and T. Chia, “Kalman filtering with state equality constraints,” *IEEE Trans. Aero. Elec. Sys.*, vol. 38, pp. 128–136, 2002.
- [2] N. Gupta and R. Hauser, “Kalman filtering with equality and inequality state constraints,” *ArXiv e-prints*, 2007.
- [3] B. O. Teixeira, J. Chandrasekar, L. A. Tórreres, L. A. Aguirre, and D. S. Bernstein, “State estimation for linear and non-linear equality-constrained systems,” *Intl. J. Control*, vol. 82, pp. 918–936, 2009.
- [4] A. Monin, “Dynamic estimation of linear systems constrained by bounds,” *IEEE Trans. Sig. Proc.*, vol. 57, pp. 4095–4099, 2009.
- [5] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,” *IET Control Theory & Applications*, vol. 4, pp. 1303–1318, 2010.
- [6] D. Simon and D. L. Simon, “Constrained Kalman filtering via density function truncation for turbofan engine health estimation,” *Intl. J. Systems Science*, vol. 41, pp. 159–171, 2010.
- [7] H. Myung, H.-K. Lee, K. Choi, S.-W. Bang, Y.-B. Lee, and S.-R. Kim, “Constrained Kalman filter for mobile robot localization with gyroscope,” in *IROS*, 2006.
- [8] L. Xu and X. Li, “Estimation and filtering of Gaussian variables with linear inequality constraints,” in *Info. Fusion (FUSION)*, 2010.
- [9] V. Sircoulomb, G. Hoblos, H. Chafouk, and J. Ragot, “State estimation under nonlinear state inequality constraints. a tracking application,” in *Mediterranean Conf. on Control and Automation*, 2008.
- [10] T. Grundmann, M. Fiebert, and W. Burgard, “Probabilistic rule set joint state update as approximation to the full joint state estimation applied to multi object scene analysis,” in *IROS*, 2010.
- [11] C. Kreucher, K. Kastella, and I. Hero, A.O., “Multitarget tracking using the joint multitarget probability density,” *IEEE Trans. Aero. Elec. Sys.*, vol. 41, no. 4, pp. 1396–1414, 2005.
- [12] O. Lanz, “Approximate Bayesian multibody tracking,” *IEEE Trans. PAMI*, vol. 28, no. 9, pp. 1436–1449, 2006.
- [13] W. C. Horrace, “Some results on the multivariate truncated normal distribution,” *J. Multivariate Analysis*, vol. 94, pp. 209–221, 2005.
- [14] J. Glover, R. Rusu, and G. Bradski, “Monte Carlo pose estimation with quaternion kernels and the Bingham distribution,” in *RSS*, 2011.