

Curved Surface Contact Patches with Quantified Uncertainty

Marsette Vona and Dimitrios Kanoulas

Abstract—We introduce a set of 10 bounded curved-surface patch types suitable for modeling local contact regions both in the environment and on a robot. We present minimal geometric parameterizations using the exponential map for spatial pose both in the usual 6DoF case and also for patches with revolute symmetry that have only 5DoF. We then give an algorithm to fit any patch type to point samples of a surface, with quantified uncertainty both in the input points (including nonuniform variance, common in data from range sensors) and in the output patch. Finally, we outline how such patches can be composed into a spatial *patch map* of the available contact surfaces both on and around a robot.

I. INTRODUCTION

Whether locomoting or manipulating, most robots make contact with environment surfaces. Contact is well-studied (e.g. [1]) but, arguably, there is not yet any accepted general system for modeling the shape and pose of potential contact surface patches, including both patches on the robot (e.g. finger tips, foot soles, etc) and also in the surrounding environment. This is especially true when (a) curved, bounded patches with (b) geometrically meaningful minimal parameterizations and (c) quantified uncertainty are desired.

We introduce a general-purpose set of ten curved and flat patch types (Fig. 1, Table I) suitable for both natural and man-made surfaces. Eight come from the general second-order polynomial approximation to a smooth surface at a given point—the principal quadric—which is always a paraboloid, possibly degenerated to a plane [2]. We add two non-paraboloid types to better model common man-made spherical and cylindrical surfaces, and we pair each surface type with a specific boundary curve to capture useful symmetries and asymmetries. We have implemented the proposed models and algorithms in the *Surface Patch Library* (SPL), with sourcecode available on our website [3].

Why curved patches? One of our long-term research interests is legged locomotion on large rocks. Flat areas can be rare in such natural environments. More broadly, contact surfaces in man-made environments are also often curved—railings, doorknobs, steering wheels, knobs, etc. Though curved surfaces can be approximated by sets of smaller planar patches [4], the job can often be done with fewer and larger curved patches. Curved surface geometry is more complex, but it may still be an advantageous trade-off to reason about fewer and larger patches. For example, a spherical robot foot stepping into a divot on a rock might be modeled as the interaction between just one spherical and one elliptic paraboloid patch (on foot and rock, respectively). If

the surfaces were approximated using collections of smaller planar patches the interaction could require combinatorial reasoning about many possible contacting pairs.

By “geometrically meaningful minimal parameterizations” we mean that each patch is defined by the fewest possible parameters, and that these have direct geometric interpretations—rotations, translations, curvatures, lengths, and angles. Geometric (vs. algebraic) parameterizations also support reasoning [5] about possible actions with patches, and allow some representation of spatial uncertainty with geometric error ellipsoids in task space. Minimality is desirable because redundant (non-minimal) parameterizations can slow the numerical optimizations used in surface fitting [6] and must be handled specially in uncertainty modeling [7].

It is often important get both a best estimate of patch parameters and a quantification of the uncertainty therein. We develop full uncertainty quantifications based on Gaussian modeling with covariance matrices. Though this is not the only way to represent uncertainty, it does cover common situations, and covariance matrices enable data fusion based on the Kalman filter. We intend our models to be usable in Kalman-type SLAM (simultaneous localization and mapping) algorithms that maintain a dynamic local *patch map*, Fig. 4, of *contact patch features* around a robot, including both environment surfaces and contact pads on the robot itself (potentially uncertain due to kinematic error).

Next we cover related work and key mathematical tools. We then give details of the patch models, followed by an algorithm to fit a patch to heteroskedastic (nonuniform variance) point cloud data from common types of range sensor. Such fitting is a typical step in recovering surface shape and pose (other important steps, including segmentation of patches, are not addressed here). We demonstrate practicality in experiments with simulated and real range data. Finally, we show how sensed environment-surface patches and kinematically uncertain patches on a robot can be combined into a single patch map with quantified uncertainty.

A. Related Work

Detecting and modeling potential contact surfaces around a robot is a common task, especially for locomotion in uncertain environments. Many prior systems are grid based [8], while some attempt to recover surface models [9]. These are typically *dense* approaches in that they attempt to model all of the terrain in view; we propose to only map a *sparse* set of patches “of interest” for the problem at hand (we acknowledge that selecting these may be a challenge). Also, our patch-based approach can homogeneously model contact surfaces both in the environment and on the robot itself,

The authors are with the College of Computer and Information Science, Northeastern University, Boston, Massachusetts
vona@ccs.neu.edu, dkanou@ccs.neu.edu

whereas most prior work considers modeling environment surfaces exclusively.

A few other works do take a sparse approach but are restricted to planar patches, ranging from large flats in man-made environments [7], [10], [11] down to small “patchlets” [12]. These mainly fall in the area of *range image segmentation*, which also has a significant history [13]. Some work has been done with curved surfaces [14], but the main focus still appears to be on planes [11]. Also, as in traditional image segmentation, the goal of range image segmentation is usually to produce a dense labeling of the whole image.

One of our main results is an algorithm to fit instances of curved, bounded patches to noisy point samples. Fitting planes is well studied [15], including uncertainty [16] and fitting heteroskedastic range data [7]. For curved surfaces quadrics are a natural option; Petitjean [2] surveyed quadric fitting, but there were few results that (a) quantified uncertainty, (b) recovered geometric parameterizations, and (c) fit bounded patches. In [5], Dai et al describe recovery of paraboloid geometric parameters¹ by linear least squares, without considering uncertainty. In [17] Wang et al studied quadric extraction in the context of range image segmentation, including quantified uncertainty in the algebraic (not geometric) patch parameters, but not on the input points. Our fitting algorithm quantifies both input and output uncertainty and recovers geometric parameters of bounded patches.

II. TOOLS

We now introduce some less common mathematical tools used in our algorithms. Due to space constraints we must refer readers elsewhere for review of some topics, including implicit and explicit (parametric) surfaces [18], Gaussian uncertainty modeling, and propagation of uncertainty [19].

A. Extrinsic and Intrinsic Surface Parameters

An instance of a patch will be a vector of real parameters which define both its shape (curvature and boundary) and its 3D rigid-body pose. We call the former *intrinsic* and the latter *extrinsic* parameters [20]. We must consider different issues to achieve minimal parametrization for each, and the distinction also enables the option to model shape (intrinsic) and pose (extrinsic) uncertainty separately. Minimal intrinsic parametrization for the proposed patches will be given by (a) one parameter for each variable curvature, and (b) a minimal parametrization of the boundary curve. However, minimal extrinsic parametrization depends on the *continuous symmetry class* of the patch. For example, a patch with two different curvatures (Fig. 2) has no continuous symmetry: its rigid body pose—here any element in the special Euclidean group $SE(3)$ —has six degrees of freedom (DoF). But a planar patch with a circular boundary has a continuous rotation symmetry and only five extrinsic DoF. Remarkably, it has been shown that there are exactly seven continuous symmetry classes in 3D [20]: revolute, prismatic, planar, spherical, cylindrical, helical, and general (the first six correspond to

the lower kinematic pairs; the last represents *no* continuous symmetry). Since we only consider patches with boundaries, we need only the general (no continuous symmetry, 6 DoF pose) and revolute (one continuous rotation symmetry, 5 DoF pose) classes—continuous translation symmetries are not possible for bounded patches.

B. Pose Representation with the Exponential Map

We require two extrinsic parameterizations: one with six parameters for asymmetric patches and one with five parameters for patches with revolute symmetry. It is well known that, because the Lie-manifold of the special orthogonal group $SO(3)$ (the rotation subgroup of $SE(3)$) is non-Euclidean, there is no singularity-free minimal parametrization of $SE(3)$. For the general case we thus select a minimal parametrization with singularities that are easiest to handle for our application. One of the core computations will be patch fitting by iterative optimization, and for this Grassia showed in [6] that a useful pose representation is²

$$[\mathbf{r}^T \ \mathbf{t}^T]^T \in \mathbb{R}^6 \text{ with } (\mathbf{r}, \mathbf{t}) \in \mathbb{R}^3 \times \mathbb{R}^3 \quad (1)$$

where \mathbf{t} is a translation and \mathbf{r} is an *orientation vector* giving an element of $SO(3)$ via an exponential map. Grassia observed that in this parametrization singularities are avoidable by a fast dynamic reparameterization, reviewed below.

We use Rodrigues’ rotation formula for the exponential map³ $R(\mathbf{r}): \mathbb{R}^3 \rightarrow SO(3) \subset \mathbb{R}^{3 \times 3}$ (Grassia used quaternions):

$$\begin{aligned} R(\mathbf{r}) &= I + [\mathbf{r}]_{\times} \alpha + [\mathbf{r}]_{\times}^2 \beta & (2) \\ \theta &\triangleq \|\mathbf{r}\|, \quad \alpha \triangleq \frac{\sin \theta}{\theta}, \quad \beta \triangleq \frac{1 - \cos \theta}{\theta^2} \\ \mathbf{r} &= \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \quad [\mathbf{r}]_{\times} \triangleq \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \end{aligned}$$

As promised, the (\mathbf{r}, \mathbf{t}) representation has a direct geometric interpretation: \mathbf{t} is just a translation, and (wlog for $\theta \neq 0$) θ gives the right-hand-rule rotation angle about the spatial axis defined by the unit vector \mathbf{r}/θ . While exponential map approaches are not new [21], [22], matrices in $se(3) \subset \mathbb{R}^{4 \times 4}$, the Lie algebra of $SE(3)$, are typically used instead of (\mathbf{r}, \mathbf{t}) . Though elegant, the former do not satisfy our goals of minimal parametrization and direct geometric interpretation.⁴

Using the fact that counterclockwise rotation by θ is equivalent to clockwise rotation by $2\pi - \theta$, Grassia’s reparametrization converts any \mathbf{r} into a canonical one with $\|\mathbf{r}\| \leq \pi$.

Algebraically, (\mathbf{r}, \mathbf{t}) corresponds to an element

$$\begin{bmatrix} R(\mathbf{r}) & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

of $SE(3)$, a 4×4 homogeneous rigid body transform, and can thus define the pose of a local coordinate frame L (and

²We explicitly notate transposes; orientation is crucial esp. for Jacobians.

³Despite division by $\theta = \|\mathbf{r}\|$, (2) converges to I as $\theta \rightarrow 0$. For numerical stability we use the series expansion approximations $\alpha \approx 1 - \theta^2/6$ and $\beta \approx 1/2 - \theta^2/24$ for small θ (threshold depends on machine precision).

⁴It is true that there is a 1:1 correspondence between matrices in $se(3)$ and elements of the (\mathbf{r}, \mathbf{t}) parametrization; conceptually, we have invoked this correspondence and simplified the results directly in terms of (\mathbf{r}, \mathbf{t}) .

¹They *verify* that the fit result is a paraboloid and extract its parameters. They do not consider *constraining* the fit to paraboloids (vs other quadrics).

a patch therein) relative to a world frame W : $R(\mathbf{r})$ is a basis for L and \mathbf{t} is its origin. The transformation of a point \mathbf{q}_l in L to \mathbf{q}_w in W , and the reverse, are familiar functions $X_{f,r} : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$

$$\mathbf{q}_w = X_f(\mathbf{q}_l, \mathbf{r}, \mathbf{t}) \triangleq R(\mathbf{r})\mathbf{q}_l + \mathbf{t} \quad (3)$$

$$\mathbf{q}_l = X_r(\mathbf{q}_w, \mathbf{r}, \mathbf{t}) \triangleq R(-\mathbf{r})(\mathbf{q}_w - \mathbf{t}) = R(\mathbf{r})^T(\mathbf{q}_w - \mathbf{t}) \quad (4)$$

where (4) makes use of the inverse transform

$$(\mathbf{r}, \mathbf{t})^{-1} \triangleq (-\mathbf{r}, -R(-\mathbf{r})\mathbf{t}) = (-\mathbf{r}, -R(\mathbf{r})^T\mathbf{t}). \quad (5)$$

Eqns. (1–4) constitute our 6 DoF pose parametrization. For the 5 DoF case, observe that only one of the three basis vectors of L need be specified; rotation symmetry allows the others to make any mutually orthogonal triple. Only two DoF are required, equivalent to specifying a point on the unit sphere. We do this by re-using (1–4) with r_z fixed at 0:

$$(\mathbf{r}_{xy}, \mathbf{t}) \in \mathbb{R}^2 \times \mathbb{R}^3 \text{ corresp. } ([\mathbf{r}_{xy}^T \ 0]^T, \mathbf{t}) \in \mathbb{R}^3 \times \mathbb{R}^3. \quad (6)$$

The geometric interpretation of $(\mathbf{r}_{xy}, \mathbf{t})$ is the same as for (\mathbf{r}, \mathbf{t}) , except that \mathbf{r}_{xy} is constrained to the xy plane. For any given canonical \mathbf{r} , a canonical \mathbf{r}_{xy} always exists that satisfies

$$R([\mathbf{r}_{xy}^T \ 0]^T)\hat{\mathbf{z}} = R(\mathbf{r})\hat{\mathbf{z}} \text{ with } [\hat{\mathbf{x}} \ \hat{\mathbf{y}} \ \hat{\mathbf{z}}] \triangleq I_{3 \times 3}. \quad (7)$$

\mathbf{r}_{xy} can be calculated as

$$\mathbf{r}_{xy}(\mathbf{r}) = \begin{bmatrix} \hat{\mathbf{x}}^T \\ \hat{\mathbf{y}}^T \end{bmatrix} \begin{cases} \mathbf{r} & \text{if } \theta_{xy} \approx \pi \\ (\hat{\mathbf{z}} \times \hat{\mathbf{z}}_l) / \alpha_{xy} & \text{otherwise} \end{cases} \quad (8)$$

$$\hat{\mathbf{z}}_l \triangleq R(\mathbf{r})\hat{\mathbf{z}}, \theta_{xy} \triangleq \text{atan2}(\|\hat{\mathbf{z}} \times \hat{\mathbf{z}}_l\|, \hat{\mathbf{z}}^T \hat{\mathbf{z}}_l), \alpha_{xy} \triangleq \frac{\sin \theta_{xy}}{\theta_{xy}}$$

As in Brockett's product of exponentials [21], 6 DoF poses can be composed to make any kinematic chain. Let

$$(\mathbf{r}_n, \mathbf{t}_n)^{\phi_n}, \dots, (\mathbf{r}_1, \mathbf{t}_1)^{\phi_1} \text{ with } \phi_i \in \{+1, -1\} \quad (9)$$

be the poses (equiv. transforms) in the chain from end to base in order from right to left. Then the pose $(\mathbf{r}_c, \mathbf{t}_c)$ of a patch attached to the end of the chain relative to the base is

$$(\mathbf{r}_c, \mathbf{t}_c) = (\mathbf{r}(R_n \cdots R_1), (X_n \circ \cdots \circ X_1)(\mathbf{0})) \quad (10)$$

$$R_j \triangleq R(\phi_j \mathbf{r}_j), X_j(\mathbf{q}) \triangleq \begin{cases} X_f(\mathbf{q}, \mathbf{r}_j, \mathbf{t}_j) & \text{if } \phi_j = +1 \\ X_r(\mathbf{q}, \mathbf{r}_j, \mathbf{t}_j) & \text{if } \phi_j = -1 \end{cases}$$

substituting $\mathbf{r}_{xy}(\mathbf{r}_c)$ for 5 DoF patches, and using the *log map* $\mathbf{r}(R) : SO(3) \rightarrow \mathbb{R}^3$ corresponding to the inverse of (2). We give an algorithm to compute $\mathbf{r}(R)$ in Appendix II.

We will need the partial derivatives of (4)

$$\frac{\partial \mathbf{q}_l}{\partial \mathbf{q}_w} = R^T, \frac{\partial \mathbf{q}_l}{\partial \mathbf{r}} = \frac{\partial R^T}{\partial \mathbf{r}}(\mathbf{q}_w - \mathbf{t}), \frac{\partial \mathbf{q}_l}{\partial \mathbf{t}} = -R^T, R \triangleq R(\mathbf{r}) \quad (11)$$

the Jacobian of (2)—including its use as part of $\partial \mathbf{q}_l / \partial \mathbf{r}$ in (11)—and the Jacobians of (8) and (10):

$$\frac{\partial R}{\partial \mathbf{r}}, \frac{\partial \mathbf{r}_{xy}}{\partial \mathbf{r}}, \frac{\partial (\mathbf{r}_c, \mathbf{t}_c)}{\partial (\mathbf{r}_1, \mathbf{t}_1), \dots, (\mathbf{r}_n, \mathbf{t}_n)}.$$

The latter three are given in Appendix I.

surface	bound	parameters		DoF	world frame equations
		intrin.	extrin.		
ell parab	ellipse	\mathbf{d}_e, \mathbf{k}	\mathbf{r}, \mathbf{c}	10	14,15,17; $\text{sign}(\kappa_x) = \text{sign}(\kappa_y)$
hyp parab	ellipse	\mathbf{d}_e, \mathbf{k}	\mathbf{r}, \mathbf{c}	10	14,15,17; $\text{sign}(\kappa_x) \neq \text{sign}(\kappa_y)$
cyl parab	aa rect	\mathbf{d}_r, κ	\mathbf{r}, \mathbf{c}	9	14,15,18,19; $\mathbf{k} = [0 \ \kappa]^T$
circ parab	circle	d_c, κ	$\mathbf{r}_{xy}, \mathbf{c}$	7	14,15,17; $\mathbf{k} = [\kappa \ \kappa]^T, \mathbf{d}_e = [d_c \ d_c]^T$
plane	ellipse	\mathbf{d}_e	\mathbf{r}, \mathbf{c}	8	14,15,17; $\mathbf{k} = \mathbf{0}$
	circle	d_c	$\mathbf{r}_{xy}, \mathbf{c}$	6	14,15,17; $\mathbf{k} = \mathbf{0}, \mathbf{d}_e = [d_c \ d_c]^T$
	aa rect	\mathbf{d}_r	\mathbf{r}, \mathbf{c}	8	14,15,18,19; $\mathbf{k} = \mathbf{0}$
	c quad	d_q	\mathbf{r}, \mathbf{c}	11	14,15,21,19; $\mathbf{k} = \mathbf{0}$
sphere	circle	d_c, κ	$\mathbf{r}_{xy}, \mathbf{c}$	7	24,25,17; $\mathbf{d}_e = [d_c \ d_c]^T, \kappa d_c \leq 1$
circ cylind	aa rect	\mathbf{d}_r, κ	\mathbf{r}, \mathbf{c}	9	28,29,18,19; $ \kappa d_y \leq 1$

TABLE I

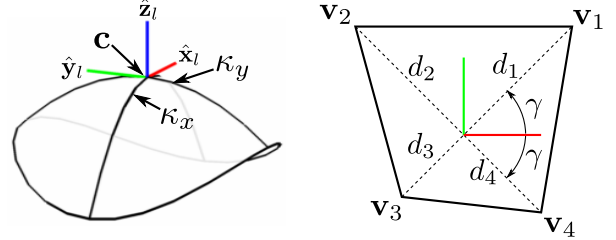


Fig. 2. Left: a paraboloid patch with two negative curvatures (κ_x, κ_y) , symmetry point \mathbf{c} , and local frame basis $[\hat{\mathbf{x}}_l \ \hat{\mathbf{y}}_l \ \hat{\mathbf{z}}_l]$. Right: a convex quadrilateral boundary parametrized by γ and $d_1 \dots d_4$.

III. PATCH MODELS

We now present the details of ten surface patch models (Fig. 1, Table I) based on seven curved surface types. Five of these partition the paraboloids, including the important degenerate case of a plane; the other two add true spherical and circular cylinder patches, non-paraboloids that are common in man-made environments and on robots. For non-planar surfaces we select one specific parametrized boundary shape which trims the surface into a local patch. For planes we allow a choice of four boundary shapes.

The next two sections give the details of the paraboloid and non-paraboloid patch models. This particular system is not the only possible taxonomy; it reflects our design choices in an attempt to balance expressiveness vs minimality.

A. Paraboloids

The best-fit degree-two local polynomial approximation to any smooth surface $S \subset \mathbb{R}^3$ at a given point $\mathbf{c} \in S$, called the *principal quadric*, is always a *paraboloid*—a quadric of one sheet with a central point of symmetry about which the surface has two independent curvatures κ_x, κ_y in orthogonal directions (Fig. 2). These are the *principal curvatures* of S at \mathbf{c} , and \mathbf{c} is the symmetry point. Defining $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{y}}_l$ as unit vectors in the directions of the principal curvatures in the tangent plane to S at \mathbf{c} , the surface normal to S at \mathbf{c} is $\hat{\mathbf{z}}_l \triangleq \hat{\mathbf{x}}_l \times \hat{\mathbf{y}}_l$. If S is considered to be embedded in a world coordinate frame W , then $\mathbf{c} \in \mathbb{R}^3$ is the origin and

$$R \triangleq [\hat{\mathbf{x}}_l \ \hat{\mathbf{y}}_l \ \hat{\mathbf{z}}_l]$$

is a basis for the *principal coordinate frame* (all standard terms) of S at \mathbf{c} , which we also call *local frame* L .

Using the log map, the transform

$$(\mathbf{r}, \mathbf{c}) \triangleq (\mathbf{r}(R), \mathbf{c})$$

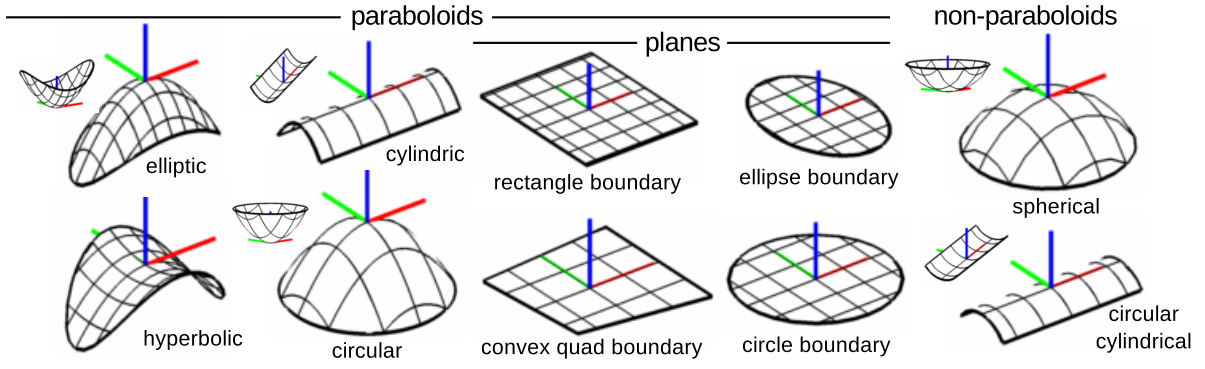


Fig. 1. Examples of all patch types, each with axes of the local coordinate frame. Concave variants shown inset.

takes points from L to W , enabling a short derivation of equations for a general paraboloid parametrized by $\mathbf{k} \triangleq [\kappa_x \ \kappa_y]^T$, \mathbf{r} , and \mathbf{c} . Starting in L where the paraboloid is in standard position, with $p_{li} : \mathbb{R}^3 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ and $p_{le} : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$0 = p_{li}(\mathbf{q}_l, \mathbf{k}) \triangleq \mathbf{q}_l^T \text{diag}([\mathbf{k}^T \ 0]^T) \mathbf{q}_l - 2\mathbf{q}_l^T \hat{\mathbf{z}} \quad (12)$$

$$\mathbf{q}_l = p_{le}(\mathbf{u}, \mathbf{k}) \triangleq [\hat{\mathbf{x}} \ \hat{\mathbf{y}}] \mathbf{u} + \frac{1}{2} \mathbf{u}^T \text{diag}(\mathbf{k}) \mathbf{u} \hat{\mathbf{z}} \quad (13)$$

are the implicit and explicit forms, respectively, with $\mathbf{q}_l \in \mathbb{R}^3$ a point on the patch in L and $\mathbf{u} \in \mathbb{R}^2$ parameters of the explicit form. Moving to $\mathbf{q}_w \in \mathbb{R}^3$ in world frame W only requires composing (12,13) with (3,4), yielding

$$0 = p_{wi}(\mathbf{q}_w, \mathbf{k}, \mathbf{r}, \mathbf{c}) \triangleq p_{li}(X_r(\mathbf{q}_w, \mathbf{r}, \mathbf{c}), \mathbf{k}) \quad (14)$$

$$\mathbf{q}_w = p_{we}(\mathbf{u}, \mathbf{k}, \mathbf{r}, \mathbf{c}) \triangleq X_f(p_{le}(\mathbf{u}, \mathbf{k}), \mathbf{r}, \mathbf{c}) \quad (15)$$

$$p_{wi} : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}, \quad p_{we} : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3.$$

Note that in this formulation \mathbf{u} is always the projection of \mathbf{q}_l onto the local frame xy plane:

$$\mathbf{u} \triangleq \Pi_{xy} \mathbf{q}_l = \Pi_{xy} X_r(\mathbf{q}_w, \mathbf{r}, \mathbf{c}), \quad \Pi_{xy} \triangleq [\hat{\mathbf{x}} \ \hat{\mathbf{y}}]^T. \quad (16)$$

In the general case $\kappa_x \neq \kappa_y$, giving 7 or 8 DoF paraboloids—6 pose DoF plus up to two curvatures. (Boundary parameterizations will add DoF.) Six DoF pose is required because $\kappa_x \neq \kappa_y$ implies no continuous rotation symmetries, only discrete symmetries about \mathbf{c} . It is standard to separate three surface types where $\kappa_x \neq \kappa_y$ (Fig. 1): *elliptic paraboloids* have two nonzero curvatures with equal signs, *hyperbolic paraboloids* have two nonzero curvatures with opposite signs, and *cylindric paraboloids* have one nonzero curvature. In all cases $\hat{\mathbf{z}}_l$ is the outward pointing surface normal and positive/negative curvatures correspond to concave/convex directions on the patch, respectively.⁵

We bound elliptic and hyperbolic paraboloid patches with ellipses in the xy plane of the local frame L , axis aligned and centered at \mathbf{c} . If $\mathbf{d}_e \triangleq [d_x \ d_y]^T$ are the ellipse radii then the bounded patch is the subset of the full surface (12–15) where, with $e : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$, \mathbf{u} satisfies

$$0 \geq e(\mathbf{u}, \mathbf{d}_e) \triangleq \mathbf{u}^T \text{diag}([1/d_x^2 \ 1/d_y^2]) \mathbf{u} - 1. \quad (17)$$

⁵To reduce ambiguity wlog choose $|\kappa_x| < |\kappa_y|$, though some ambiguity is unavoidable due to bilateral symmetries.

For cylindric paraboloid patches, replace the ellipse boundary with an axis aligned rectangle with half-widths $\mathbf{d}_r = [d_x \ d_y]^T$. In the xy plane of L the vertices are

$$\mathbf{v}_1 \triangleq \mathbf{d}_r, \mathbf{v}_2 \triangleq [-d_x \ d_y]^T, \mathbf{v}_3 \triangleq -\mathbf{v}_1, \mathbf{v}_4 \triangleq -\mathbf{v}_2 \quad (18)$$

in counterclockwise order, and the bounding condition can be stated as, with $q : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$,

$$0 \geq q(\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4) \triangleq \quad (19)$$

$$\max(l(\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2), l(\mathbf{u}, \mathbf{v}_2, \mathbf{v}_3), l(\mathbf{u}, \mathbf{v}_3, \mathbf{v}_4), l(\mathbf{u}, \mathbf{v}_4, \mathbf{v}_1))$$

where $l : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ is the implicit form for a 2D line given two points on the line; \mathbf{u} is on or to the left of the directed line through \mathbf{v}_i towards \mathbf{v}_j iff

$$0 \geq l(\mathbf{u}, \mathbf{v}_i, \mathbf{v}_j) \triangleq (\mathbf{u} - \mathbf{v}_i)^T [\mathbf{v}_j - \mathbf{v}_i]_{\perp}, \quad \begin{bmatrix} x \\ y \end{bmatrix}_{\perp} \triangleq \begin{bmatrix} y \\ -x \end{bmatrix}. \quad (20)$$

For the special case $\kappa_x = \kappa_y$ we identify two more surface types (Fig. 1): *circular paraboloids* have both curvatures equal and non-zero, and planes have both curvatures zero. Both of these have continuous rotation symmetry about $\hat{\mathbf{z}}_l$, so we use the 5-DoF pose parametrization $(\mathbf{r}_{xy}, \mathbf{c})$, provided that the patch boundary also has the same continuous rotation symmetry. The latter holds for circular boundaries, which we use for circular paraboloids. Let κ be the surface curvature and d_c the bounding circle radius; circular paraboloids are then defined by (12–17) with $\mathbf{k} = [\kappa \ \kappa]^T$, $\mathbf{r} = [\mathbf{r}_{xy}^T \ 0]^T$, $\mathbf{d}_e = [d_c \ d_c]^T$, and with the dimensions of the function domains correspondingly reduced.

For the important case of paraboloids degenerated to planes we give a choice of four boundary types: ellipses, circles, rectangles, or general convex quadrilaterals (developed next). For all except circles, the planar patch loses its continuous rotation symmetry and requires full 6-DoF pose parametrization; the patch is defined by (12–15) with $\mathbf{k} = \mathbf{0}$ (and correspondingly reduced function domains) and either (17) or (19). Planar patches with circular boundary are the same as circular paraboloids but with $\mathbf{k} = \mathbf{0}$.

For convex quadrilateral boundaries, keep \mathbf{c} at the intersection of the diagonals $\overline{\mathbf{v}_1 \mathbf{v}_3}$ and $\overline{\mathbf{v}_2 \mathbf{v}_4}$ (Fig. 2), where $\mathbf{v}_{1..4}$ are the vertices in counterclockwise order in the xy plane of local frame L . Define γ as half the angle between the

diagonals and $d_{1..4} \geq 0$ the half-diagonal lengths such that

$$\begin{aligned} \mathbf{v}_i &\triangleq d_i [\cos \phi_i \ \sin \phi_i]^T & (21) \\ \phi_1 &\triangleq \gamma, \ \phi_2 \triangleq \pi - \gamma, \ \phi_3 \triangleq \pi + \gamma, \ \phi_4 \triangleq -\gamma \\ &0 < \gamma < \pi/2. \end{aligned}$$

Then the quadrilateral is defined by (19) using vertices (21) parametrized by $\mathbf{d}_q \triangleq [d_1 \ d_2 \ d_3 \ d_4 \ \gamma]^T$. Convexity is ensured by construction, and only five parameters are needed even though a general planar quadrilateral has 8 DoF—the remaining three (a rotation about the plane normal and two in-plane translations) are contributed by the extrinsic pose.

B. Spheres and Circular Cylinders

Spheres and circular cylinders are common on robots and in man-made environments. Though still quadrics, neither is a paraboloid, suggesting two additional patch types (Fig. 1). (We do not model complete spheres or cylinders, only bounded patches of hemispheres and half-cylinders.)

Again starting in local frame L , the implicit and explicit equations of an upright hemisphere with apex at the origin and curvature κ (hence possibly infinite radius $|1/\kappa|$) are⁶, with $s_{li} : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}$ and $s_{le} : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^3$,

$$0 = s_{li}(\mathbf{q}_l, \kappa) \triangleq \kappa \mathbf{q}_l^T \mathbf{q}_l - 2\mathbf{q}_l^T \hat{\mathbf{z}}, \quad 0 \leq \kappa \mathbf{q}_l^T \hat{\mathbf{z}} \leq 1 \quad (22)$$

$$\mathbf{q}_l = s_{le}(\mathbf{u}, \kappa) \triangleq [\hat{\mathbf{x}} \ \hat{\mathbf{y}}] \mathbf{u} + (\hat{\mathbf{z}}/\kappa) \left(1 - \sqrt{1 - \kappa^2 \mathbf{u}^T \mathbf{u}}\right). \quad (23)$$

Composing these with (3,4) gives the world frame forms $s_{wi} : \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, $s_{we} : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$

$$0 = s_{wi}(\mathbf{q}_w, \kappa, \mathbf{r}_{xy}, \mathbf{c}) \triangleq s_{li}(X_r(\mathbf{q}_w, [\mathbf{r}_{xy}^T \ 0]^T, \mathbf{c}), \kappa) \quad (24)$$

$$0 = s_{we}(\mathbf{u}, \kappa, \mathbf{r}_{xy}, \mathbf{c}) \triangleq X_f(s_{le}(\mathbf{u}, \kappa), [\mathbf{r}_{xy}^T \ 0]^T, \mathbf{c}). \quad (25)$$

Circular half-cylinder surfaces are similar but (a) have no dependence on x_l and (b) require 6 DoF pose:

$$0 = c_{li}(\mathbf{q}_l, \kappa) \triangleq \mathbf{q}_l^T K \mathbf{q}_l - 2\mathbf{q}_l^T \hat{\mathbf{z}}, \quad 0 \leq \kappa \mathbf{q}_l^T \hat{\mathbf{z}} \leq 1 \quad (26)$$

$$\mathbf{q}_l = c_{le}(\mathbf{u}, \kappa) \triangleq [\hat{\mathbf{x}} \ \hat{\mathbf{y}}] \mathbf{u} + (\hat{\mathbf{z}}/\kappa) \left(1 - \sqrt{1 - \kappa^2 \mathbf{u}^T Y \mathbf{u}}\right) \quad (27)$$

$$K \triangleq \text{diag}([0 \ \kappa \ \kappa]^T), \quad Y \triangleq [0 \ 1]^T [0 \ 1]$$

$$0 = c_{wi}(\mathbf{q}_w, \kappa, \mathbf{r}, \mathbf{c}) \triangleq c_{li}(X_r(\mathbf{q}_w, \mathbf{r}, \mathbf{c}), \kappa) \quad (28)$$

$$0 = c_{we}(\mathbf{u}, \kappa, \mathbf{r}, \mathbf{c}) \triangleq X_f(c_{le}(\mathbf{u}, \kappa), \mathbf{r}, \mathbf{c}). \quad (29)$$

To maintain revolute symmetry we use circular boundary for spherical patches: \mathbf{u} must satisfy (17) with $\mathbf{d}_e = [d_c \ d_c]^T$ and $|\kappa|d_c \leq 1$. For circular cylinder patches we use rectangular boundary: \mathbf{u} must satisfy (18,19) with $|\kappa|d_y \leq 1$.

IV. PATCH FITTING

It is natural to consider recovering patches from point samples, especially now that high quality range sensors are easily available, including stereo and structured light systems like the Microsoft/PrimeSense Kinect, time-of-flight cameras, and laser scanners. Much work has been done on range image *segmentation* [14], but it is still an interesting question how to best segment point samples of a surface for the purpose of mapping patches; we leave this open for

now, and focus on the underlying *fitting* problem, which is neither trivial—we have found very few prior reports on fitting curved surface patches to noisy data with quantified uncertainty both in the inputs (the points) and the outputs (the patch parameters). Though linear least squares (LLS) can fit a quadric surface to points [5], and its extension to linear χ^2 maximum likelihood fits data corrupted by white noise, the problem appears to become nonlinear when the points are heteroskedastic (i.e. have nonuniform variance). Also, we want to fit bounded paraboloids, spheres, and circular cylinders, not just unconstrained quadrics.

Whether based on stereo or time-of-flight, range data does in fact exhibit heteroskedasticity—typically there is much more uncertainty in range than aim [7], [12], the variance changes with range, and because the measurement rays usually have a single center of projection, the error ellipsoids for the sampled points are not co-oriented: each is elongated in the direction of its own measurement ray (Fig 3).

Our fitting algorithm handles these issues. The **inputs** are N sample points $\mathbf{q}_i \in \mathbb{R}^3$ with covariance matrices $\Sigma_i \in \mathbb{R}^{3 \times 3}$; general surface type⁷ $s \in \{\text{parab}, \text{plane}, \text{sphere}, \text{ccyl}\}$; if $s = \text{plane}$, boundary type $b \in \{\text{ellipse}, \text{circle}, \text{aarect}, \text{cquad}\}$ (b is implied if $s \neq \text{plane}$); and a boundary containment probability $\Gamma \in (0, 1]$. The **outputs** are the fitted patch type (s, b) , parameters $\mathbf{p} \in \mathbb{R}^p$, and covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$, where p is the DoF of the patch type (Table I).

The algorithm proceeds in 9 steps. The first three fit an unbounded surface; the rest are largely concerned with fitting the bounds, which can include final resolution of the patch center and orientation (in steps 6 and 9) where the bounding shape breaks symmetries of the underlying surface.

1. (i) Fit a plane with LLS, ignoring Σ_i . Unless $s = \text{parab}$, re-fit the plane with *weighted Levenberg-Marquardt* (WLM), detailed below, including Σ_i , using (24) with $\kappa = 0$. (ii) Set $\mathbf{c} \leftarrow \bar{\mathbf{q}} - \hat{\mathbf{z}}_l^T (\bar{\mathbf{q}} - \mathbf{c}) \hat{\mathbf{z}}_l$ (perp. proj. of $\bar{\mathbf{q}} \triangleq \text{avg}(\mathbf{q}_i)$ on plane).
2. If $s \neq \text{plane}$: (i) With $\mathbf{k} = [0 \ 0]^T$, $\mathbf{r} \triangleq [\mathbf{r}_{xy}^T \ 0]^T$, and \mathbf{c} from 1 as initial estimates, according to s fit an unbounded paraboloid, sphere, or circ cyl with WLM on (14,24,28). (ii) If $s = \text{sphere}$ keep \mathbf{r}_{xy} from 1. If $s = \text{circ cyl}$ set $\mathbf{r} = \mathbf{r}([\hat{\mathbf{x}}_l \ \hat{\mathbf{y}}_l \ \hat{\mathbf{z}}_l])$ (log map) where $\hat{\mathbf{z}}_l$ is the normal of the plane from 1, $\hat{\mathbf{x}}_l$ is along the fitted cylinder axis, and $\hat{\mathbf{y}}_l \triangleq \hat{\mathbf{z}}_l \times \hat{\mathbf{x}}_l$.
3. If $s = \text{parab}$, refine it based on the fitted curvatures $\mathbf{k} = [\kappa_x \ \kappa_y]^T$: If $\max(|\kappa_x|, |\kappa_y|) < \epsilon_k$ (a small threshold), set $s = \text{plane}$, $b = \text{ellipse}$, and \mathbf{r}_{xy} using (8). Else if $\min(|\kappa_x|, |\kappa_y|) < \epsilon_k$ swap axes s.t. $|\kappa_y| > \epsilon_k$, then set $s = \text{cyl parab}$ and $\kappa = \kappa_y$. Else if $|\kappa_x - \kappa_y| < \epsilon_k$ set $s = \text{circ parab}$, $\kappa = \text{avg}(\kappa_x, \kappa_y)$, and \mathbf{r}_{xy} using (8). Else if $\text{sign}(\kappa_x) = \text{sign}(\kappa_y)$ set $s = \text{ell parab}$. Else set $s = \text{hyp parab}$.
4. If $s \neq \text{plane}$, set b based on s and⁸ $\lambda \triangleq \sqrt{2} \text{erf}^{-1}(\Gamma)$.
5. Project the data $\mathbf{q}_i \in \mathbb{R}^3$ to $\mathbf{u}_i = [x_i \ y_i]^T \in \mathbb{R}^2$ using (16). Set $\bar{x} \triangleq \text{avg}(x_i)$, $\bar{y} \triangleq \text{avg}(y_i)$, $v_x \triangleq \text{avg}(x_i^2)$, $v_y \triangleq \text{avg}(y_i^2)$.

⁷Taking s, b as inputs allows constrained fitting of specific types; they could be automatically found by checking all possibilities for the best fit.

⁸We use 1D and 2D moments to fit approximate boundaries, see e.g. [23].

⁶In the limit as $\kappa \rightarrow 0$ (22–29) all reduce to planes.

6. If $s \in \{\text{cyl parab, circ cyl}\}$ set $\mathbf{d}_r = \lambda[\sqrt{v_x - \bar{x}^2} \sqrt{v_y}]^T$ and $\mathbf{c} \leftarrow X_f(\bar{x}\hat{\mathbf{x}}, \mathbf{r}, \mathbf{c})$.

7. If $s \in \{\text{circ parab, sphere}\}$ set $d_c = \lambda \max(\sqrt{v_x}, \sqrt{v_y})$.

8. If $s \in \{\text{ell parab, hyp parab}\}$ set $\mathbf{d}_e = \lambda[\sqrt{v_x} \sqrt{v_y}]^T$.

9. If $s = \text{plane}$, \mathbf{r}_{xy} and \mathbf{c} will be available from either 1 or 3.

3. Set $\mathbf{c} \leftarrow X_f(\bar{x}\hat{\mathbf{x}} + \bar{y}\hat{\mathbf{y}}, \mathbf{r}_{xy}, \mathbf{c})$ and (c.f. [23])

$$l_{+,-} \triangleq \sqrt{-\ln(1-\Gamma)(\alpha + \phi \pm \sqrt{\beta^2 + (\alpha - \phi)^2})} \quad (30)$$

$$\alpha \triangleq v_x - \bar{x}^2, \quad \beta \triangleq 2(\text{avg}(x_i y_i) - \bar{x}\bar{y}), \quad \phi \triangleq v_y - \bar{y}^2.$$

If $b = \text{circle}$ set $d_c = \max(l_+, l_-)$ and \mathbf{r}_{xy} from \mathbf{r} using (8). If $b \in \{\text{ellipse, aarect}\}$ set $\mathbf{d}_{e,r} = [l_+ \ l_-]^T$. If $b = \text{conv quad}$ set⁹

$$\mathbf{d}_q = [d \ d \ d \ d \ \gamma]^T, \quad d \triangleq \sqrt{l_-^2 + l_+^2}, \quad \gamma \triangleq \text{atan2}(l_-, l_+). \quad (31)$$

If $b \neq \text{circle}$, using plane normal $\hat{\mathbf{z}}_l \triangleq R([\mathbf{r}_{xy}^T \ 0]^T) \hat{\mathbf{z}}$ set

$$\mathbf{r} = \mathbf{r}([\hat{\mathbf{x}}_l \ \hat{\mathbf{y}}_l \ \hat{\mathbf{z}}_l]) \quad (\text{log map}) \quad (32)$$

$$\theta \triangleq (1/2) \text{atan2}(\beta, \alpha - \phi), \quad \hat{\mathbf{x}}_l \triangleq [\cos \theta \ \sin \theta \ 0]^T, \quad \hat{\mathbf{y}}_l \triangleq \hat{\mathbf{z}}_l \times \hat{\mathbf{x}}_l.$$

Σ is calculated by first order error propagation through the above computations, see our implementation [3] for details.

A. Weighted Levenberg-Marquardt

Levenberg-Marquardt (LM) is a standard iterative nonlinear optimization [24]. It can find a parameter assignment $\mathbf{p}_{\text{opt}} \in \mathbb{R}^p$ that locally minimizes the sum-of-squares residual r of a differentiable objective function $f : \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}$ applied to a dataset $\mathbf{q}_i \in \mathbb{R}^d$, $1 \leq i \leq N$, starting from an initial estimate \mathbf{p}_0 . That is, it finds

$$\mathbf{p}_{\text{opt}} = \underset{\mathbf{p} \text{ near } \mathbf{p}_0}{\text{argmin}} \ r, \quad r \triangleq \sum_{i=1}^N e_i^2, \quad e_i \triangleq f(\mathbf{q}_i, \mathbf{p}). \quad (33)$$

Implementations typically take as inputs functions f and $\partial f / \partial \mathbf{p}$, the data \mathbf{q}_i , and \mathbf{p}_0 , and return both \mathbf{p}_{opt} and a covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ representing its uncertainty. A well known extension is to replace e_i with $E_i \triangleq e_i / \sigma_i$ where $\sigma_i > 0$ are constant standard deviations modeling uncertainty in e_i . The residual is then called χ^2 , and \mathbf{p}_{opt} locally maximizes the likelihood of the ‘‘observations’’ e_i .

For our use, f is always the implicit form of a surface in world frame, i.e. p_{wi} , s_{wi} , or c_{wi} (14,24,28). The σ_i are not constant, but can be estimated with first order error propagation as¹⁰

$$\sigma_i = \sqrt{\text{var}(f(\mathbf{q}_i, \mathbf{p}))} \triangleq \sqrt{v_f(i, \mathbf{p})} \quad (34)$$

$$v_f(i, \mathbf{p}) \triangleq \left(\frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}_i, \mathbf{p}) \right) \Sigma_i \left(\frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}_i, \mathbf{p}) \right)^T.$$

We define *weighted* LM (WLM) to combine σ_i and f into a meta-objective function $F : [1 \dots n] \times \mathbb{R}^p \rightarrow \mathbb{R}$:

$$F(i, \mathbf{p}) \triangleq f(\mathbf{q}_i, \mathbf{p}) / \sigma_i = f(\mathbf{q}_i, \mathbf{p}) / \sqrt{v_f(i, \mathbf{p})}. \quad (35)$$

⁹We currently fit convex quad the same as aa rect, but note that the former is still useful for designed (vs fit) patches, e.g. part of a foot sole.

¹⁰A heuristic to allow *semi*-definite Σ_i is to clamp small $v_f(i, \mathbf{p})$ to a minimum positive limit.

Both F and its gradient $\partial F / \partial \mathbf{p}$ are implied given \mathbf{q}_i , Σ_i , f , $\partial f / \partial \mathbf{p}$, $\partial f / \partial \mathbf{q}$, and $\partial^2 f / \partial \mathbf{p} \partial \mathbf{q}$ (which is $d \times p$):

$$\frac{\partial F}{\partial \mathbf{p}}(i, \mathbf{p}) = \frac{\frac{\partial f}{\partial \mathbf{p}}(\mathbf{q}_i, \mathbf{p})}{\sigma_i} - e_i \frac{\frac{\partial f}{\partial \mathbf{q}}(\mathbf{q}_i, \mathbf{p}) \Sigma_i \frac{\partial^2 f}{\partial \mathbf{p} \partial \mathbf{q}}(\mathbf{q}_i, \mathbf{p})}{\sigma_i v_f(i, \mathbf{p})}. \quad (36)$$

Given \mathbf{q}_i , Σ_i , f , $\partial f / \partial \mathbf{p}$, $\partial f / \partial \mathbf{q}$, and $\partial^2 f / \partial \mathbf{p} \partial \mathbf{q}$, WLM synthesizes F and $\partial F / \partial \mathbf{p}$ by (35,36) and then applies LM. This is simplified further by the common form of the world-frame surfaces (14,24,28), which are all variants of

$$f_l(\mathbf{q}_l, \mathbf{k}_3) \triangleq \mathbf{q}_l^T K \mathbf{q}_l - 2 \mathbf{q}_l^T \hat{\mathbf{z}} \quad (37)$$

$$f_w(\mathbf{q}_w, \mathbf{p}_s) = f_l(X_r(\mathbf{q}_w, \mathbf{r}, \mathbf{c}), \mathbf{k}_3) \quad (38)$$

$$\mathbf{p}_s \triangleq [\mathbf{k}_3^T \ \mathbf{r}^T \ \mathbf{c}^T]^T, \quad \mathbf{k}_3 \triangleq [\kappa_x \ \kappa_y \ \kappa_z]^T, \quad K \triangleq \text{diag}(\mathbf{k}_3)$$

where some components of \mathbf{k}_3 , and for (24) the last component of \mathbf{r} , are held at zero. The required derivatives of (38) are given by the chain rule from (11) and derivatives of (37).

B. Experimental Results

We tested the fitting algorithm both in real data from a Kinect and in simulation (Fig. 3). In lieu of automatic segmentation, we implemented a simple interactive segmenter—this was used for the physical experiment only; the simulated patches were already isolated. We used the two-parameter pointing/disparity stereo error model proposed by Murray and Little in [12] (based on earlier work by others) to estimate input sample covariances Σ_i for all experiments. The error model parameters we used for the Kinect are $\sigma_{\text{pointing}} = 0.35\text{px}$, $\sigma_{\text{disparity}} = 0.17\text{px}$; the former is from [25], the latter was determined experimentally following [12].

The results graphically show that the algorithm can produce reasonable curved-surface paraboloid patch models for local parts of non-flat environment surfaces. Average times for our unoptimized Matlab implementation are $\sim 225\text{ms}$ to fit $n \approx 150$ sample points on a commodity workstation. (SVD computations within LM are quadratic in n , though runtime also depends on the LM convergence rate.)

V. THE PATCH MAP

Once patches are fitted (or otherwise modeled) they can be assembled into a spatial map including quantified uncertainty, providing a sparse ‘‘summary’’ of nearby surfaces for higher-level reasoning (Fig. 4). Relative to a given frame of reference (a body-centered frame may be appropriate in locomotion) patch models can either be sensed exteroceptively by range sensors or proprioceptively through kinematic and tactile sensing. Exteroception can detect upcoming terrain patches from a distance, but with relatively high uncertainty. Kinematic proprioception senses the pose of contact patches on the robot itself—e.g. heel, toe, foot sole—potentially with relatively low uncertainty. Finally, once a contact is established, the environment patch can be re-measured *exteroceptively* through kinematics and touch, possibly with reduced uncertainty compared to prior exteroception.

All of the classic elements of SLAM [26] are needed to build such a map: propagation of spatial uncertainty through kinematic chains, associating different observations

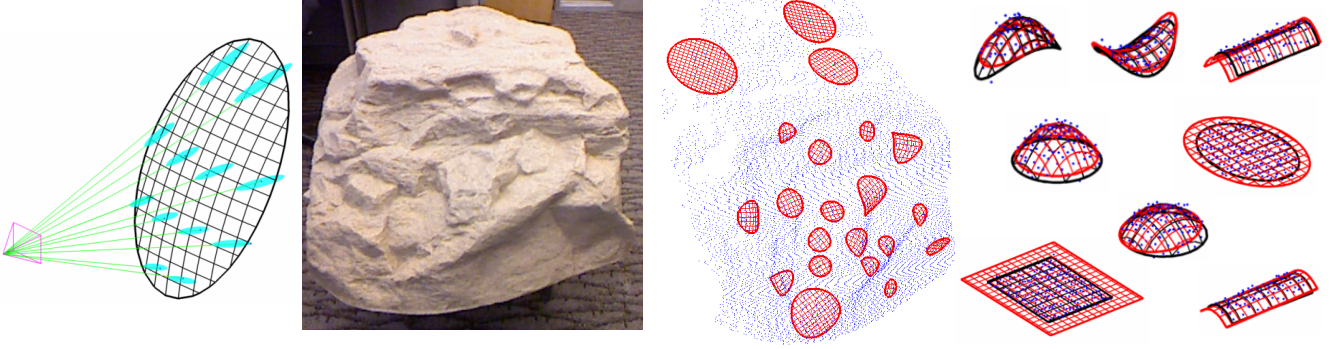


Fig. 3. Left: 95% probability error ellipsoids for simulated stereo range sensing using the pointing/disparity error model of Murray and Little [12] (pointing error exaggerated for illustration). Middle: Experimental dataset; (fake) rock is $\sim 70 \times 30 \times 90$ cm W \times H \times D; ~ 125 k samples collected in a single scan with a Kinect at a distance of ~ 1 m (decimated for display); 21 patches manually segmented and automatically fit. Right: Automatic fits (red) for a variety of paraboloid and non-paraboloid (lower right) patch types in simulated noisy range samples, using Kinect projection and error models.

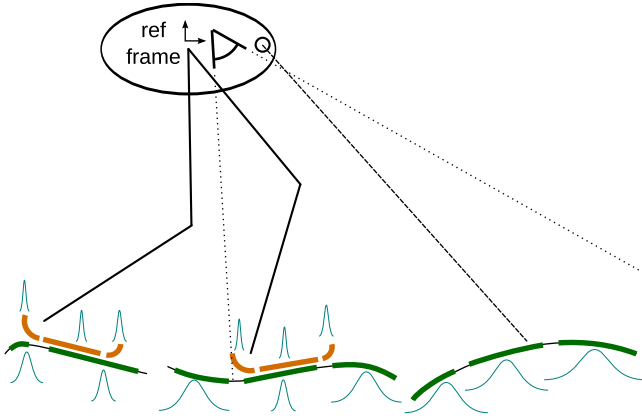


Fig. 4. Concept of the *patch map*: a sparse set of patches locally approximate both environment surfaces (green) and key contact surfaces on a robot (brown). All are spatially mapped with quantified uncertainty (blue Gaussians) relative to a body-centered reference frame.

of the same surface patch, and optimal data fusion. The association problem could be challenging, we leave it open for now. Fusion by Kalman update is supported by the patch covariance matrices. First-order propagation of uncertainty through a chain of transforms with 6×6 covariances S_j is facilitated by the chain Jacobian J_c given in Appendix I:

$$\Sigma_c = J_c S J_c^T, \quad S \triangleq \text{diag}(S_n, \dots, S_1). \quad (39)$$

$\Sigma_c \in \mathbb{R}^{6 \times 6}$ is the covariance of the pose of a patch at the end of the chain relative to the base. For a 5-DoF patch,

$$\Sigma_{c_5} = J_5 J_c S J_c^T J_5^T, \quad J_5 \triangleq \begin{bmatrix} \frac{\partial \mathbf{r}_{xy}}{\partial \mathbf{r}} & 0 \\ 0 & I_{3 \times 3} \end{bmatrix}. \quad (40)$$

VI. CONCLUSIONS AND FUTURE WORKS

We introduced a set of 10 particular bounded curved-surface patch types, an algorithm to fit patches to noisy point samples of a surface, and the concept of a *patch map* to sparsely represent potentially useful contact surfaces in the environment near a robot, and also on the robot itself. The presented patch models all have minimal geometric parameterizations and quantified uncertainty in the form of covariance matrices.

Though surface modeling and surface fitting have been studied extensively, many prior works ignore uncertainty,

are limited to planes, are dense vs sparse, and/or are concerned only with surfaces in the environment. We address all of these issues to some extent. Though some challenging aspects of the problem—such as segmentation and data association—remain future work, we demonstrated the effectiveness of our approach both in real data and in realistic simulation. We invite the reader to try our approach for themselves using the provided implementation [3].

APPENDIX I

We calculate the Jacobian of the exponential map (2) as a $[3 \times 3] \times 3$ row tensor¹¹:

$$\begin{aligned} \frac{\partial R}{\partial \mathbf{r}} &= [\mathbf{r}]_{\times} \frac{\partial \alpha}{\partial \mathbf{r}} + \frac{\partial [\mathbf{r}]_{\times}}{\partial \mathbf{r}} \alpha + [\mathbf{r}]_{\times}^2 \frac{\partial \beta}{\partial \mathbf{r}} + \frac{\partial [\mathbf{r}]_{\times}^2}{\partial \mathbf{r}} \beta \quad (41) \\ \frac{\partial \alpha}{\partial \mathbf{r}} &= \frac{\theta \cos \theta - \sin \theta}{\theta^3} \mathbf{r}^T, \quad \frac{\partial \beta}{\partial \mathbf{r}} = \frac{\theta \sin \theta + 2 \cos \theta - 2}{\theta^4} \mathbf{r}^T \\ \frac{\partial [\mathbf{r}]_{\times}}{\partial \mathbf{r}} &= \begin{bmatrix} 0 & \hat{\mathbf{z}} & -\hat{\mathbf{y}} \\ -\hat{\mathbf{z}} & 0 & \hat{\mathbf{x}} \\ \hat{\mathbf{y}} & -\hat{\mathbf{x}} & 0 \end{bmatrix} \\ \frac{\partial [\mathbf{r}]_{\times}^2}{\partial \mathbf{r}} &= [\mathbf{r}]_{\times} \frac{\partial [\mathbf{r}]_{\times}}{\partial \mathbf{r}} + \frac{\partial [\mathbf{r}]_{\times}}{\partial \mathbf{r}} [\mathbf{r}]_{\times}. \end{aligned}$$

The Jacobian of (8) is, with θ_{xy} , α_{xy} , and $\hat{\mathbf{z}}_l$ from (8),¹²

$$\begin{aligned} \frac{\partial \mathbf{r}_{xy}}{\partial \mathbf{r}} &= \begin{bmatrix} \hat{\mathbf{x}}^T \\ \hat{\mathbf{y}}^T \end{bmatrix} \begin{cases} I & \text{if } \theta_{xy} \approx \pi \\ \frac{\partial}{\partial \mathbf{r}} \frac{\hat{\mathbf{z}} \times \hat{\mathbf{z}}_l}{\alpha_{xy}} & \text{otherwise} \end{cases} \quad (42) \\ \frac{\partial}{\partial \mathbf{r}} \frac{\hat{\mathbf{z}} \times \hat{\mathbf{z}}_l}{\alpha_{xy}} &= [\hat{\mathbf{z}}]_{\times} \left(\frac{I}{\alpha_{xy}} - RZ(\gamma_{xy} R^T (I-Z) + I) \right) \frac{\partial R}{\partial \mathbf{r}} \hat{\mathbf{z}} \\ R \triangleq R(\mathbf{r}), \quad Z \triangleq \hat{\mathbf{z}} \hat{\mathbf{z}}^T, \quad \gamma_{xy} &\triangleq \frac{\theta_{xy} - \sin \theta_{xy} \cos \theta_{xy}}{\sin^3 \theta_{xy}}. \end{aligned}$$

Viewing (10) as a vector function,

$$[\mathbf{r}_n^T \ \mathbf{t}_n^T \ \dots \ \mathbf{r}_1^T \ \mathbf{t}_1^T]^T \in \mathbb{R}^{6n} \rightarrow [\mathbf{r}_c^T \ \mathbf{t}_c^T]^T \in \mathbb{R}^6,$$

its Jacobian J_c is $6 \times 6n$ where 6×6 block j from right to left

¹¹(41) remains finite as $\theta \rightarrow 0$. Small angle approximations for α and β were given in footnote 3; their derivatives can be approximated as $\partial \alpha / \partial \mathbf{r} \approx (\theta^2/30 - 1/3) \mathbf{r}^T$ and $\partial \beta / \partial \mathbf{r} \approx (\theta^2/180 - 1/12) \mathbf{r}^T$.

¹²For small θ_{xy} , $\gamma_{xy} \approx 2/(3 - \theta_{xy}^2/2)$.

is, with ϕ_j from (9), R_j, X_j from (10), $\frac{\partial \mathbf{r}}{\partial R}$ from Appendix II,

$$\frac{\partial [\mathbf{r}_c^T \mathbf{t}_c^T]^T}{\partial [\mathbf{r}_j^T \mathbf{t}_j^T]^T} = \begin{bmatrix} \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}_j} & 0 \\ \frac{\partial \mathbf{t}_c}{\partial \mathbf{r}_j} & \frac{\partial \mathbf{t}_c}{\partial \mathbf{t}_j} \end{bmatrix}, \frac{\partial \mathbf{r}_c}{\partial \mathbf{r}_j} = \frac{\partial \mathbf{r}_c}{\partial R_c} R_l \frac{\partial R_j}{\partial \mathbf{r}_j} R_r \quad (43)$$

if $\phi_j = +1$ if $\phi_j = -1$

$$\frac{\partial \mathbf{t}_c}{\partial \mathbf{r}_j} = R_l \frac{\partial R_j}{\partial \mathbf{r}_j} \mathbf{t}_r \quad \frac{\partial \mathbf{t}_c}{\partial \mathbf{r}_j} = R_l \frac{\partial R_j}{\partial \mathbf{r}_j} (\mathbf{t}_r - \mathbf{t}_j)$$

$$\frac{\partial \mathbf{t}_c}{\partial \mathbf{t}_j} = R_l \quad \frac{\partial \mathbf{t}_c}{\partial \mathbf{t}_j} = -R_l R_j$$

$$R_l \triangleq R_n \cdots R_{j+1}, R_r \triangleq R_{j-1} \cdots R_1, R_c \triangleq R_n \cdots R_1$$

$$\mathbf{t}_r \triangleq (X_{j-1} \circ \cdots \circ X_1)(\mathbf{0}).$$

APPENDIX II

Due to numerical issues with other equations we found in the literature, we developed the following numerically stable algorithm to calculate the log map

$$\mathbf{r}(R), R \triangleq \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix}$$

and its $3 \times [3 \times 3]$ column tensor Jacobian $\partial \mathbf{r} / \partial R$. Explanation omitted for space; the algorithm is tested in our software [3].

$$\mathbf{v} = [R_{zy} - R_{yz} \quad R_{xz} - R_{zx} \quad R_{yx} - R_{xy}]^T$$

$$c = (\text{tr}(R) - 1)/2, s = \|\mathbf{v}\|/2, \theta = \text{atan2}(s, c)$$

choose $ijk \in \{xyz, yzx, zxy\}$ s.t. $R_{ii} = \max(R_{xx}, R_{yy}, R_{zz})$

$$\delta = 1 + R_{ii} - R_{jj} - R_{kk}$$

if $\delta > \epsilon_\delta$ then $\triangleright R$ not identity, θ not small

$$\gamma = \theta(3 - \text{tr}(R))^{-\frac{1}{2}}, d = \sqrt{\delta}$$

$$r_i = d\gamma, r_j = \gamma(R_{ji} + R_{ij})/d, r_k = \gamma(R_{ki} + R_{ik})/d$$

$$\mathbf{r} = [r_x \quad r_y \quad r_z]^T \triangleright \text{solution up to sign}$$

if $\theta < (\pi - \epsilon_\theta) \triangleright$ resolve sign by testing action of R

$$\mathbf{p} = \mathbf{r} \times [0 \ 0 \ 1]^T, \text{ if } \mathbf{p}^T \mathbf{p} < 1/4 \text{ then } \mathbf{p} = \mathbf{r} \times [0 \ 1 \ 0]^T$$

$$\text{if } (R\mathbf{p})^T (\mathbf{r} \times \mathbf{p}) < 0 \text{ then } \mathbf{r} \leftarrow -\mathbf{r}, d \leftarrow -d$$

\triangleright solution for $\mathbf{r}(R)$ complete, now find $\partial \mathbf{r} / \partial R$

$$\hat{\mathbf{r}} = [\hat{r}_x \quad \hat{r}_y \quad \hat{r}_z]^T = \mathbf{r} / \theta, \partial \theta / \partial R = (c[\hat{\mathbf{r}}]_\times - sI) / 2$$

$$w_i = 1, w_j = -1, w_k = -1, U = \text{diag}([w_x \quad w_y \quad w_z]^T)$$

$$\gamma \partial d / \partial R = (\gamma / (2d))U, d \partial \gamma / \partial R = \hat{r}_i \left(\frac{\partial \theta}{\partial R} + \frac{I\theta}{6 - 2 \text{tr } R} \right)$$

$$\partial r_i / \partial R = \gamma \partial d / \partial R + d \partial \gamma / \partial R$$

$$V = 0_{3 \times 3}, V_{ji} \leftarrow -1, V_{ij} \leftarrow -1, W = 0_{3 \times 3}, W_{ki} \leftarrow -1, W_{ik} \leftarrow -1$$

$$\partial r_j / \partial R = (\gamma / d)V + (R_{ji} + R_{ij})(d \partial \gamma / \partial R - \gamma \partial d / \partial R) / \delta$$

$$\partial r_k / \partial R = (\gamma / d)W + (R_{ki} + R_{ik})(d \partial \gamma / \partial R - \gamma \partial d / \partial R) / \delta$$

$$\partial \mathbf{r} / \partial R = [\partial r_x / \partial R \quad \partial r_y / \partial R \quad \partial r_z / \partial R]^T$$

else \triangleright small θ

$$\text{if } \theta > \epsilon_\theta \text{ then } \alpha = s / \theta \text{ else } \alpha = 1 - \theta^2 / 6$$

$$\mathbf{r} = \mathbf{v} / (2\alpha) \triangleright \text{solution for } \mathbf{r}(R), \text{ now find } \partial \mathbf{r} / \partial R$$

$$\text{if } \theta > \epsilon_\theta \text{ then } \lambda = (s - c\theta) / (2s^2), \partial \theta / \partial R = (c[\mathbf{r} / \theta]_\times - sI) / 2$$

$$\text{else } \lambda = \theta / 12, \partial \theta / \partial R = (c(1_{3 \times 3} - I) - sI) / 2$$

\triangleright using $\partial[\mathbf{r}]_\times / \partial \mathbf{r}$ from (41) and Kronecker product \otimes

$$\partial \mathbf{r} / \partial R = (1 / (2\alpha))(\partial[\mathbf{r}]_\times / \partial \mathbf{r})^T + \lambda \mathbf{v} \otimes (\partial \theta / \partial R)$$

REFERENCES

- [1] M. T. Mason and J. K. Salisbury, Jr., *Robot hands and the mechanics of manipulation*. MIT Press, 1985.
- [2] S. Petitjean, "A survey of methods for recovering quadrics in triangle meshes," *ACM Computing Surveys*, vol. 34, pp. 211–262, 2002.
- [3] M. Vona and D. Kanoulas, "Surface Patch Library (SPL)," 2011, <http://ccis.neu.edu/research/gpc/spl>.
- [4] N. Vaskevicius, A. Birk, K. Pathak, and S. Schwertfeger, "Efficient representation in 3D environment modeling for planetary robotic exploration," *Advanced Robotics*, vol. 24, no. 8-9, 2010.
- [5] M. Dai, T. S. Newman, and C. Cao, "Least-squares-based fitting of paraboloids," *Pattern Recognition*, vol. 40, pp. 504–515, 2007.
- [6] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *J. Graphics Tools*, vol. 3, no. 3, pp. 29–48, 1998.
- [7] K. Pathak, N. Vaskevicius, and A. Birk, "Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds," in *Proceedings of the IEEE international conference on Robotics and Automation*, 2009, pp. 2035–2040.
- [8] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker, "Ambler—an autonomous rover for planetary exploration," *IEEE Computer*, vol. 22, pp. 18–26, 1989.
- [9] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "A bayesian regression approach to terrain mapping and an application to legged robot locomotion," *Journal of Field Robotics*, vol. 26, pp. 789–811, 2009.
- [10] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2006, pp. 3062–3067.
- [11] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.
- [12] D. Murray and J. J. Little, "Patchlets: Representing stereo vision data with surface elements," *IEEE Workshop on Motion and Video Computing*, vol. 1, pp. 192–199, 2005.
- [13] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673–689, 1996.
- [14] M. W. Powell, K. W. Bowyer, X. Jiang, and H. Bunke, "Comparing curved-surface range image segmenters," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1998.
- [15] C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, and K. Yamamoto, "Comparison of local plane fitting methods for range data," in *CVPR (I)*, 2001, pp. 663–669.
- [16] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*. Dover Publications, Incorporated, 2005.
- [17] G. Wang, Z. Houkes, G. Jia, B. Zhenga, and X. Li, "An estimation-based approach for range image segmentation: On the reliability of primitive extraction," *Pattern Recognition*, vol. 36, pp. 157–169, 2003.
- [18] M. E. Mortenson, *Geometric Modeling*. Industrial Press, 2006.
- [19] S. L. Meyer, *Data Analysis for Scientists and Engineers*. Peer Management Consultants, Ltd., 1992.
- [20] V. Srinivasan, *Theory of Dimensioning*. Marcell Dekker, 2003.
- [21] R. W. Brockett, "Robotic manipulators and the product of exponentials formula," pp. 120–129, 1983.
- [22] F. C. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 643–647, 1994.
- [23] L. Rocha, L. Velho, and P. C. P. Carvalho, "Image moments-based structuring and tracking of objects," in *Brazilian Symposium on Computer Graphics and Image Processing*, 2002.
- [24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing*. New York, NY, USA: Cambridge University Press, 1992.
- [25] K. Konolige and P. Mihelich, "Technical description of kinect calibration," 2010, http://www.ros.org/wiki/kinect_calibration/technical.
- [26] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, pp. 56–68, 1986.