Incremental Model-Based Estimation using Geometric Constraints

Cristian Sminchisescu
University of Toronto
6 King's College Road
Toronto, Ontario
Canada M5S 3G4
crismin@cs.toronto.edu

Dimitris Metaxas
Rutgers University
Busch Campus
Piscataway, NJ, USA
USA 08854
dnm@cs.rutgers.edu

Sven Dickinson
University of Toronto
6 King's College Road
Toronto, Ontario
Canada M5S 3G4
sven@cs.toronto.edu

Abstract

We present a model-based framework for incremental, adaptive object shape estimation and tracking in monocular image sequences. Parametric structure and motion estimation methods usually assume a fixed class of shape representation (splines, deformable superquadrics, etc.*) that is initialized prior to tracking. Since the model shape coverage is fixed a-priori, the incremental recovery of structure is decoupled from tracking, thereby limiting both processes in their scope and robustness. In this work, we describe a model-based framework that supports the automatic detection and integration of low-level geometric primitives (lines) incrementally. Such primitives are not explicitly captured in the initial model, but are moving consistently with its image motion. The consistency tests used to identify new structure are based on trinocular constraints between geometric primitives. The method allows not only an increase in the model scope, but also improves tracking accuracy by including the newly recovered features in its state estimation. The formulation is a step towards automatic model building, since it allows both weaker assumptions on the availability of a prior shape representation and on the number of features that would otherwise be necessary for entirely bottom-up reconstruction. We demonstrate the proposed approach on two separate image-based tracking domains, each involving complex 3D object structure and motion.*

Keywords: *shape recovery, object tracking, parametric models, geometric constraints, bundle adjustment, optimization.*

1 Introduction

Many applications, such as robust and flexible object tracking or generic object recognition, depend on the recovery of reduced part models that capture the coarse shape of an object. For tracking, it is desirable to build models on the fly, but often enough image features cannot be simultaneously detected and tracked, over sufficient frames, in order to obtain the minimum number of equations for direct, bottom-up reconstruction. To address these problems, various reduced prior shape models have been used [24, 1, 13, 40, 28, 29, 16, 44, 25, 26]. For instance, deformable superquadrics represent a powerful class of models whose recovery from range data has met with considerable success, *e.g.*, [36, 8]. However, the recovery and tracking of such models from 2-D data have been elusive, due to the weak 3-D shape constraints provided by a sparse set of 2-D features, such as contours or regions. In simple scenes, where extracted image regions (or contour groups) map one-to-one to the surfaces of qualitatively-defined parts, high-level, parametric part recovery is possible [12, 11]. However, for images of real objects, in which salient image features do not necessarily map to salient model structure [23], the recovery of the coarse shape of an object is an open problem.

Consider, for example, the situation in which only a small portion of the object can be fit, albeit crudely, with a set of qualitatively defined, parametric volumetric parts. For moving objects or cameras, the estimates of both the shape and pose of the parts could be improved by tracking them. In previous work, a similar approach was used to successfully track simple, part-based objects in image sequences [5, 11]. However, if the part coverage is poor, *i.e.*, much of the object's shape is not modeled by the recovered parts, or if the level of abstraction high, *i.e.*, there is little pixel-based correspondence between detected image features and projected model features, neither the accuracy of the recovered shape and pose, nor the scope of the model will improve with tracking. How, then, can we exploit the motion of the object to improve both the accuracy and scope of the recovered shape?

In this paper, we present a dynamic, incremental approach to shape recovery and tracking, assuming that at least part of the model can be recovered during initialization. We adopt a parametric framework, where 3-D volumetric part models can adapt to the data. While in the initial frame, the pose and shape of the parts may not be accurately recovered, as the object moves the motion is tracked and the part shape refined. We show that the motion estimates provided by the initial parts can be often sufficient to identify other object structure that is moving consistently with them. Here, we use geometric consistency constraints to verify relations between the rigid parameters of the 3D model and independently moving image lines. Consistent lines are reconstructed and included in the 3D model in order to improve its shape and motion estimation over time.[1]

Organization: Following a discussion of related work, in §2 we present our parametric model estimation framework in terms of geometry and dynamics, while in §4, we give a framework to incrementally integrate consistently moving line features into the model. In §5, we discuss two experiments involving objects with complex shape and motion and show that the method is able to recover new model structure efficiently from monocular video sequences. Our quantitative results show that the incrementally recovered structure significantly improves the accuracy and speed of the tracking process, providing important constraints, especially during difficult-to-estimate degenerate object configurations with respect to the camera.

2 Related Work

Approaches to structure and motion estimation can be broadly classified as top-down (model-based) and bottom-up (feature-based). *Model-based techniques* rely on minimizing a residual error between model feature predictions and image observations assigned to them: (*i*) CAD-based methods [24, 1, 13] assume precise, off-line constructed rigid object models and estimate their motion using non-linear least-squares. Despite their effectiveness, such methods are limited to tracking known objects having fixed, non-adaptive shapes. (*ii*) Physics-based frameworks [40, 28, 29, 16] employ either reduced d.o.f. object parameterizations, like splines or superquadrics, or point-based representations with regularized physical properties[2] . From an optimization viewpoint, the deformable approaches involve quadratic energy functions and are solved using estimation schemes based on gradient descent to find local minima. (*iii*) Parametric models [44, 25, 26] are specifically built to represent certain classes of objects, but without physical analogy. Although their formulation is different from the physics-based methods, they are similar in terms of flexibility and mathematical treatment, *i.e.*, their structural and rigid parameters are estimated iteratively using non-linear techniques.

Bottom-up, feature-based structure and motion estimation techniques differ in the types of correspondences (2-D to 2-D, 2-D to 3-D, or 3-D to 3-D) and features (lines, points, or corners) assumed available – see [21, 18] for a review. Some reconstruction algorithms are based on trilinear (or more generally multilinear) constraints between points and lines [37, 32, 17] in multiple views. Others [7, 45, 14] incrementally reconstruct features as

[1] The model state combines higher-level 3D shape parameters and low-level line features. The shape has a point discretization and image measurements are collected for points and lines (fig. 1).

[2] Regularization is based on ‘physical’ measures like stiffness or damping.

they become available, by exploiting pairwise distance or (for lines) angular invariance constraints. All these methods require a minimum number of simultaneously available features, in order to obtain enough equations to directly solve for structure and motion. *Rigid* non-linear batch approaches [38, 2, 39] can work with missing features and are based on the inversion of the forward model, a method closely related to the relative orientation algorithm [20]. *Deformable* batch methods [3] extend classical factorization schemes [42] to recover a linear deformable model representation and the camera motion, under certain rigidity assumptions.

In this work, we rely on flexible parametric models [44, 40, 28, 29, 16, 25] as basic representational primitives. Nevertheless, as powerful as these techniques are, they have two important limitations:

(*i*) *Fixed Representation*: A common assumption is that the model representation is fixed and known a-priori, sometimes imposing a heavy burden on initialization [12, 11]. Furthermore, a representational gap exists between the coarse, high-level parametric shapes used to model the objects, and low-level features like points, lines, corners or curved contours that can be detected in the image [23]. It is not obvious how to bridge this gap to represent object markings, discontinuities, or other fine surface detail through the inclusion of other basic geometric primitives, *e.g.*, lines or planes, *etc*. In fact, the diversity of parameterizations corresponding to different features at different abstraction levels usually leads to difficulties when integrating them within a single representation or optimization procedure [31]. Our method aims for a representation that is flexible, can be estimated jointly (in a single optimization problem), provides higher-level abstraction and low-level image coverage, and can be refined and augmented during tracking.

(*ii*) *Estimation and Dimensionality*: The flexibility of an object representation that can adapt comes at the expense of more parameters to estimate. To avoid singularities or ill-conditioning, it is important to use complementary image cues that can induce local minima with large, stable basins of attraction in parameter space. (For good modeling, these correspond to true object localization in the image.) Methods for constraint (cue) integration in a model-based framework have successfully used contours and stereo [41], shading and stereo [16], contours and optical flow [9], and shading [30]. Beyond the particular choice of sources of information they use, these approaches differ in the way they fuse them. Some combine information in a symmetric manner, weighting it statistically (*e.g.*, *soft* constraints). Others favor a particular hierarchical constraint satisfaction order with an exact policy, such that inconsistent contributions to the solution from constraints further down in the hierarchy are pruned away by constraints higher up (*hard* constraints).

We work in a robust model-based tracking framework, where we assume an incomplete initial model and recover additional structure using geometric consistency tests. The tests are based on those used in separate, bottom-up structure and motion estimation for 2-D to 2-D line correspondences in the Euclidean calibrated case [21, 27, 46]. Unlike these approaches, we assume an incomplete adaptive model, and we do not solve for the rigid parameters in a bottom-up fashion. Instead, given the model's estimated rigid parameters and independently tracked lines in the image, we test only if these motions are consistent, and reconstruct the lines that pass the test. Their image contribution is fused together with point-based contour and intensity observations into an augmented model-based cost function for tracking (§4.5).

3 Model Representation and Estimation

The next two sections review the geometric modeling and optimization used in our framework (see [33] for details). We describe how new image features, initially not modeled, are detected and reconstructed, and how we design modified cost functions that include them. Model parameters are computed using a robust MAP estimator. For increased reliability, this can be embedded in a multiple hypothesis framework [19, 4, 34]. Given the second-order continuity of our cost surface, direct multiple minima search methods [35] are also applicable.

3.1 Model Geometry

The reference shape of the model is defined over a domain Ω as $\mathbf{p} = \mathbf{G}(\mathbf{x_d}, \mathbf{u})$, where $\mathbf{G}$ defines a global deformation based on parameters $\mathbf{x_d}$, and $\mathbf{u} \in \Omega$ is an element of the model discretization (a point on its surface mesh). The model is represented with (possibly) multiple deformable superquadric ellipsoid parts having global tapering and bending deformations [36, 40]. The prediction of an element $\mathbf{u}$ in the image is computed as $\mathbf{r} = \mathbf{P}(\mathbf{T}(\mathbf{G}(\mathbf{x}_r, \mathbf{p})))$, where $\mathbf{T}$ is a rigid displacement represented by $\mathbf{x}_r$, and $\mathbf{P}$ is a perspective image transform. The rigid and non-rigid parameters are assembled in a model state vector, $(\mathbf{x}_r, \mathbf{x}_d)$. The state is also augmented with incrementally recovered 3D line parameters (see fig. 1).

3.2 Cost Function and Optimization

During tracking, robust prediction-to-image matching cost metrics, and their gradient and Hessians, $\mathbf{g}_i, \mathbf{H}_i$, are evaluated for each predicted model feature $\mathbf{r}_i$ (for model feature $\mathbf{u}_i$), and the results are summed over all features to produce the image contribution to the overall parameter space cost function. We use image-based cost metrics, such as robust normalized edge energy, intensity-based cost metrics, and feature-based cost metrics (for new lines that are incrementally recovered). Thus, we (implicitly or explicitly) associate the predictions $\mathbf{r}_i$ with one or more nearby image features $\bar{\mathbf{r}}_i$. The cost is a robust function ρ of the prediction error $\Delta\mathbf{r}_i(\mathbf{x}) = \bar{\mathbf{r}}_i - \mathbf{r}_i(\mathbf{x})$, where $\rho(s)$ can be any increasing function with $\rho(0) = 0$ and $\frac{d}{ds}\rho(0) = \frac{\nu}{\sigma^2}$. This models error distributions corresponding to a central peak with scale σ, and a widely spread background of outliers ν.

The overall parameter space cost function consists of terms from contour f_C, intensity f_I, and incrementally reconstructed lines f_L: $f = f_C + f_I + f_L$. Model state estimation is based on local cost optimization. We use a second order trust region method, where a descent direction is chosen by solving the regularized subproblem [15]: $(\mathbf{H} + \lambda\mathbf{W})\,\delta\mathbf{x} = -\mathbf{g}$, where $\mathbf{W}$ is a symmetric positive definite damping matrix, λ is a dynamically chosen weighting factor, $\mathbf{g} = \frac{df}{\mathbf{dx}}$, and $\mathbf{H} = \frac{d^2 f}{\mathbf{dx}^2}$. In our case, $\mathbf{H} = \mathbf{H}_C + \mathbf{H}_I + \mathbf{H}_L$ and $\mathbf{g} = \mathbf{g}_C + \mathbf{g}_I + \mathbf{g}_L$. Specific forms for individual feature costs are given next.

3.2.1 Contour Cost

Simple image preprocessing operations are used during feature extraction for the contour likelihood, as follows: 1) the images are smoothed with a Gaussian kernel; 2) they are contrast normalized; 3) a Canny edge detection is applied; and 4) an edge distance Chamfer image is computed. Given the distance image, we build a 2-dimensional continuous potential surface, $f_C = \frac{1}{2}\int ||\mathbf{P_c}||^2\mathbf{du}$, by fitting local quadric surfaces to 3x3 image patches (this is a windowed parabolic fitting method, also known as Savitsky-Golay filtering). The gradient and Hessian matrices of the corresponding contour cost term (applied to model features that lie on occluding contours or on high surface curvature) can be derived from the model-image Jacobian and the corresponding $\mathbf{P_c}$ quadric terms:

$$\mathbf{g}_C = \int \frac{\mathbf{dP_c}(\mathbf{r}(\mathbf{x}))}{\mathbf{dx}}\,\mathbf{du} = \int \mathbf{J}^\top \frac{\mathbf{dP_c}}{\mathbf{dr}}\,\mathbf{du} \tag{1}$$

$$\mathbf{H}_C = \int \frac{\mathbf{d^2P_c}}{\mathbf{dx}^2}\,\mathbf{du} \approx \int \mathbf{J}^\top \frac{\mathbf{d}^2\mathbf{P_c}}{\mathbf{dr}^2}\mathbf{J}\,\mathbf{du} \tag{2}$$

3.2.2 Intensity Cost

To model not only geometric image features, like edges, but also image intensity variations, we use cost models based on intensity residuals, $\Delta\mathbf{I}$. The observables are image gray values or colors, $\mathbf{I}$, rather than feature coordinates $\mathbf{r}$. To transfer from a point projection model, $\mathbf{r} = \mathbf{r}(\mathbf{x})$, to an intensity-based one, we compose with the assumed local intensity model, $\mathbf{I} = \mathbf{I}(\mathbf{r})$, and premultiply point Jacobians by point-to-intensity Jacobians, $\frac{\mathbf{dI}}{\mathbf{dr}}$. Given the intensity cost: $f_I = \frac{1}{2}\int \|\Delta\mathbf{I}\|^2 \mathbf{du}$, the gradient is:

$$\mathbf{g}_I = \int \rho' \mathbf{J}^\top \Delta\mathbf{I}^\top \frac{\mathbf{dI}}{\mathbf{dr}}\,\mathbf{du} \tag{3}$$

Similarly, the cost Hessian in a Gauss-Newton approximation is:

$$\mathbf{H}_I \approx \int \rho'' \mathbf{J}^\top (\frac{\mathbf{dI}}{\mathbf{dr}})^\top \frac{\mathbf{dI}}{\mathbf{dr}}\,\mathbf{J}\,\mathbf{du} \tag{4}$$

Used in optimization, the intensity cost provides ‘soft’ model-based optical flow constraints. Implicitly, the image patches under the 3D model prediction during initialization (or the previous image during tracking) are registered against the current image. The inter-frame flow is explained by the 2D variations allowed by the shape and motion parameters of the 3D model. We collect measurements at visible model nodes inside the predicted convex-hull and avoid the occluding contours because of likely optical flow constraint boundary violations.

4 Line Feature Formulation

In §3, tracking is based on robust estimation, where model parameters are constrained by contour and intensity observations. These are localized in the neighborhood of predictions for the already-known model parts. In this section, we show how new lines can be integrated into the model – these are not part of the initial model, but evidence for them exists in the image. Tracking starts with a minimal model, and incrementally over time, we: 1) identify line features moving consistently with the model, and 2) augment the model with those features to improve its tracking.

Our incremental tracking method is based on image-level and model-level processing. We use contour and intensity observations to estimate the rigid and non-rigid parameters of the model. Independently, we use image-based techniques to detect and track lines (This involves interest point tracking and line fitting, see §5). These are *image-tracked lines* **(ITLs)**. We decide if an ITL (not present in the model) belongs to the object using two geometric consistency tests, derived from ITL’s in at least three frames. The lines that pass the test are *consistent image-tracked lines* **(CITLs)**. We recover CITL structure in a model-centered coordinate system, and predict their appearance in subsequent images based on the current estimate of the model's rigid motion. These predictions are the *model-predicted lines* **(MPLs)**. The error between a CITL and a MPL is used to define additional image alignment cost terms. The reconstructed lines are then re-estimated jointly with other model parameters, to improve robustness and remove bias. The tracking pipeline is shown in fig. 1.

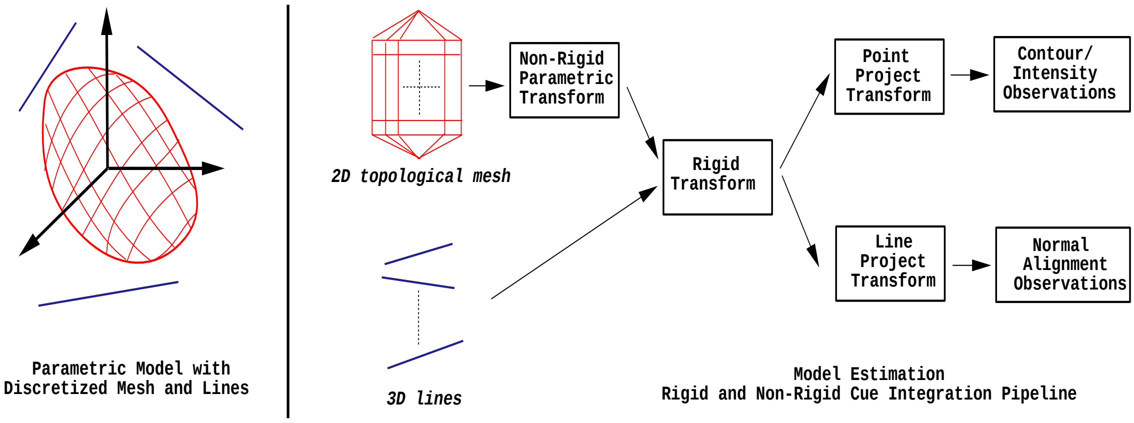

Parametric Model with
Discretized Mesh and Lines
2D topological mesh
3D lines
Non-Rigid Parametric Transform
Rigid Transform
Point Project Transform
Contour/ Intensity Observations
Line Project Transform
Normal Alignment Observations
Model Estimation
Rigid and Non-Rigid Cue Integration Pipeline

Figure 1: Estimation pipeline for incremental tracking. On the left, we show the parametric model and incrementally recovered lines. On the right, we give the predictive pipeline that relates model parameters to image measurements during tracking. The recovered line features are included in the model, and all parameters are jointly estimated in a non-linear refinement loop as follows: the coarse shape parameters, the new reconstructed lines, and the common rigid parameters. This gives robust and unbiased results.

4.1 Line parameterization

We denote 3-D lines by l_i $(i = 1 \ldots n)$, and their projected image lines (or segments) by L_i $(i = 1 \ldots n)$. A 3-D line is parameterized by a unit vector **v**, representing one of its two possible directions, and a vector **d** perpendicular to l (see fig. 2). This is a 6-dimensional over-parameterization with only 4 intrinsic d.o.f. The constraints among variables determine a 4-dimensional manifold in the 6-dimensional representation space, and any line can be identified with two points on this manifold [22, 39].

The line l and the optical center of the camera determine a plane (the line's interpretation plane) with normal, $\mathbf{N} = (N_x, N_y, N_z)^\top$. The interpretation plane intersects the image plane, defined by $z = f$ (with f the camera focal length), at L. The equation of L is:

$$N_x X + N_y Y + N_z f = 0 \qquad (5)$$

The relation allows the plane normal containing a 3-D line to be recovered from the

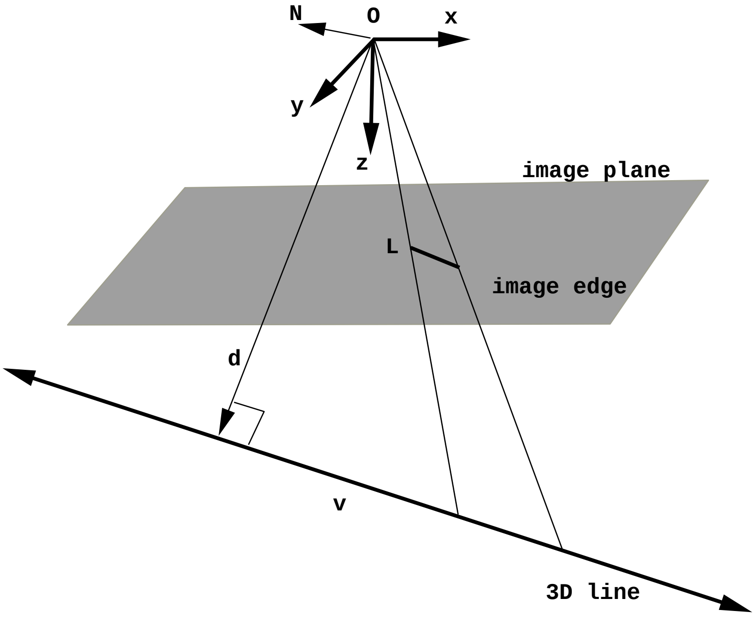


Figure 2: Line parameterization using distance and unit direction vectors. The representation is 6-dimensional, but there are only 4 d.o.f., due to unit direction and orthogonality constraints.

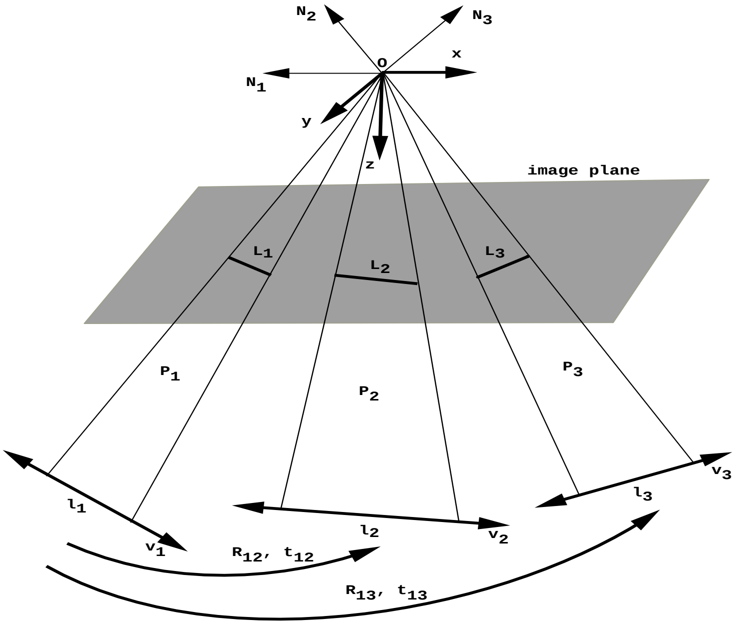


Figure 3: Line motion in three frames.

equation of its projection. Given a plane, $\mathbf{P} = (N_x, N_y, N_z, 0)^\top = (\mathbf{N}^\top, 0)^\top$, and any point $\mathbf{X} = (X, Y, Z, 1)^\top$ belonging to the interpretation plane of a line, the plane equation is: $\mathbf{P}^\top \mathbf{X} = \mathbf{0}$.

4.2 Model-Based Consistency Tests

Standard formulations for structure and motion estimation using image line correspondences (*e.g.*, [21, 46, 27]) rely on three frames and at least six line correspondences (although no formal proof is yet available [21]) to uniquely recover the structure and motion of a rigid object[3]. For the two view case, the resulting system of equations does not constrain the motion at all. There is a consistent structure for any set of image lines and any motion.

Consider the motion of a line, l, in three successive frames ($l_i, i = 1, 2, 3$, with direction $\mathbf{v_i}, i = 1, 2, 3$) and image projections L ($L_i, i = 1, 2, 3$). The motion between frames 1 and 2 has translation $\mathbf{t}_{12}$, and rotation $\mathbf{R}_{12}$, whereas for frames 1 and 3, these are $\mathbf{t}_{13}$ and $\mathbf{R}_{13}$. The normals of the interpretation planes, P_1, P_2, P_3, determined by L_i and the center of projection, are $\mathbf{N_1}, \mathbf{N_2}, \mathbf{N_3}$, respectively (see fig. 3).

The constraints between line normals and the rigid motion in 3 frames can be derived either geometrically or algebraically as [21, 27, 46]:

$$\mathbf{N_1} \cdot (\mathbf{R}_{12}^{-1}\mathbf{N_2} \times \mathbf{R}_{13}^{-1}\mathbf{N_3}) = 0 \qquad (6)$$

$$-\mathbf{t}_{12} \cdot (\mathbf{R}_{12}\mathbf{N_1}) = \frac{\|\mathbf{N_2} \times \mathbf{R}_{12}\mathbf{N_1}\|}{\|\mathbf{N_2} \times \mathbf{R}_{23}^{-1}\mathbf{N_3}\|} \cdot \mathbf{R}_{23}^{-1}\mathbf{t}_{23} \cdot \mathbf{R}_{23}^{-1}\mathbf{N_3} \qquad (7)$$

In this model-based approach, we do not directly solve for rotation and translation. Instead, given a model with *known* motion and *independent* ITLs, we *verify* if the lines move consistently with the model (CITLs). Given an ITL *in three frames* (*i.e.*, knowing $\mathbf{N_1}$, $\mathbf{N_2}$ and $\mathbf{N_3}$) and the rigid motion of the model (*i.e.*, $\mathbf{R}_{12}$, $\mathbf{t}_{12}$, $\mathbf{R}_{13}$, $\mathbf{t}_{13}$), the relations (6) and (7) are used to test if the 2-D line motion is consistent with the 3-D rigid motion. It can be shown that (6), (7) are necessary and sufficient conditions for consistency[4]. If the test is verified, we hypothesize that the line is part of the object and we include it in its model. Notice that this model-based test is flexible. It can apply to individual ITLs and does not require a minimal set of three-frame ITLs, as in bottom-up structure from motion algorithms.

[3] Within a scale factor for translation and structure parameters.

[4] A 3D line has 4 intrinsic degrees of freedom while a projected image line has only 2. Measurements are collected in 3 frames, so this will determine 3x2-4=2 independent relations.

4.3 Model-Based Structure Recovery

Once a moving image line has been assigned to the model, we reconstruct its 3-D structure, *i.e.*, $(\mathbf{v}, \mathbf{d})$, in a model-centered coordinate system. To increase robustness, one can use as many line correspondences in as many frames (at least two) as are available. This is done as follows: all interpretation planes for the lines in the camera coordinate system are transformed to a common, model-centered coordinate system. Each line, l_i, with interpretation plane, $\mathbf{P_i}$, is displaced by $\mathbf{T_c^{-1}T_i^{-1}}$, where:

$$\mathbf{T_c} = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ 0 & 1 \end{bmatrix} \tag{8}$$

is the displacement of the camera, and $\mathbf{T_i}$ is the displacement of the model (in the world coordinate system) in image frame i. The equation of the plane in the object-centered coordinate system is: $\mathbf{P_i}^\top \cdot \mathbf{T_c^{-1}T_i^{-1}} \cdot \mathbf{X} = \mathbf{0}$. By stacking together the equations for all lines, we obtain:

$$\mathbf{A} \cdot \mathbf{X} = \begin{bmatrix} \mathbf{P_1^\top} \cdot \mathbf{T_c^{-1}T_1^{-1}} \\ \mathbf{P_2^\top} \cdot \mathbf{T_c^{-1}T_2^{-1}} \\ \cdots \\ \cdots \\ \cdots \\ \mathbf{P_k^\top} \cdot \mathbf{T_c^{-1}T_k^{-1}} \end{bmatrix} \cdot \mathbf{X} = \mathbf{0} \tag{9}$$

All the planes have to intersect at a common line, so the [k x 4] matrix $\mathbf{A}$ should have rank 2. Any point $\mathbf{p}$ on the intersecting line can be written as a linear combination of the singular vectors corresponding to the 2 smallest singular values of $\mathbf{A}$:

$$\mathbf{p} = a \cdot \mathbf{X_{s1}} + b \cdot \mathbf{X_{s2}} \tag{10}$$

The line can be reconstructed as:

$$\mathbf{v} = \mathbf{X_{s1}} - \mathbf{X_{s2}} \qquad \mathbf{d} = (\mathbf{I} - \frac{\mathbf{v} \cdot \mathbf{v}^\top}{\| \mathbf{v} \|^{\mathbf{2}}}) \cdot \mathbf{X_{s1}} \tag{11}$$

The stability of the reconstruction can be verified in terms of the ratio of the 2nd and 3rd singular values of $\mathbf{A}$ (in the noise-free case, the last two singular values should be zero), being satisfactory when this ratio is high. This linear method, although robust, is prone to bias in the initial line parameter estimates, due to fixed rigid displacements. To remove bias, we work in a non-linear estimation framework, that jointly re-estimates all the model parameters (including new reconstructed lines), based on robust, statistically meaningful error norms.

4.4 Forward Model Line Prediction

Once consistent lines (CITL) have been identified, reconstructed, and included in the model, they are used to improve tracking by providing additional constraints on the alignment with the object.

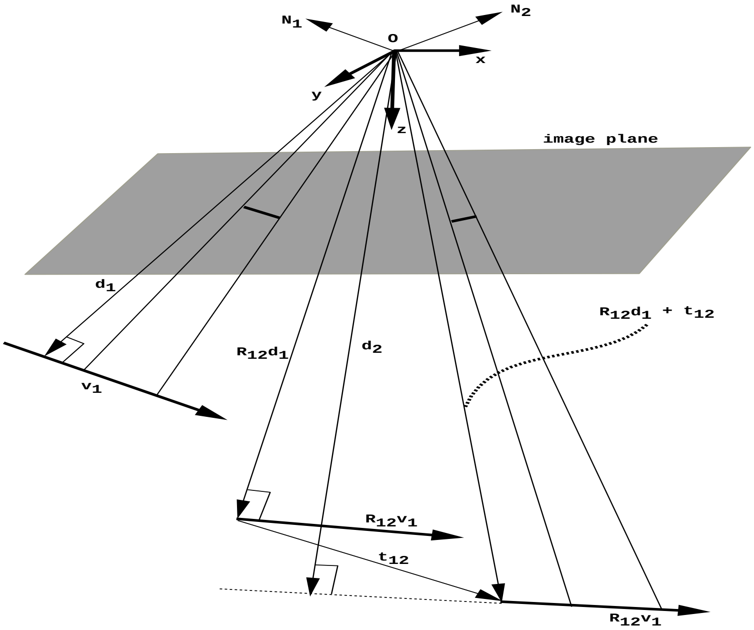


Figure 4: Forward line transfer under the action of the Euclidean group.

Consider the two frame case, as shown in fig. 4. Given $(\mathbf{N_1}, \mathbf{v_1}, \mathbf{d_1})$, and the model rigid motion $\mathbf{t}_{12}, \mathbf{R}_{12}$, we can obtain $(\mathbf{N_2}, \mathbf{v_2}, \mathbf{d_2})$ as:

$$\mathbf{v_2} = \mathbf{R}_{12}\mathbf{v_1} \tag{12}$$

$$\mathbf{d_2} = (\mathbf{R}_{12}\mathbf{d_1} + \mathbf{t_{12}}) - \mathbf{v_2}((\mathbf{R}_{12}\mathbf{d_1} + \mathbf{t_{12}}) \cdot \mathbf{v_2}) \tag{13}$$

$$\mathbf{N_2} = \frac{\mathbf{v_2} \times \mathbf{d_2}}{\parallel \mathbf{v_2} \times \mathbf{d_2} \parallel} \tag{14}$$

The Jacobian of the 3D line transform w.r.t. structure and motion parameters is complex but straightforward to derive analytically (we used Maple for automatic differentiation). Given a line represented as: $\mathbf{k}_l = (\mathbf{v}^\top, \mathbf{d}^\top)^\top$, and the rigid model parameters, $\mathbf{x}_r$, we

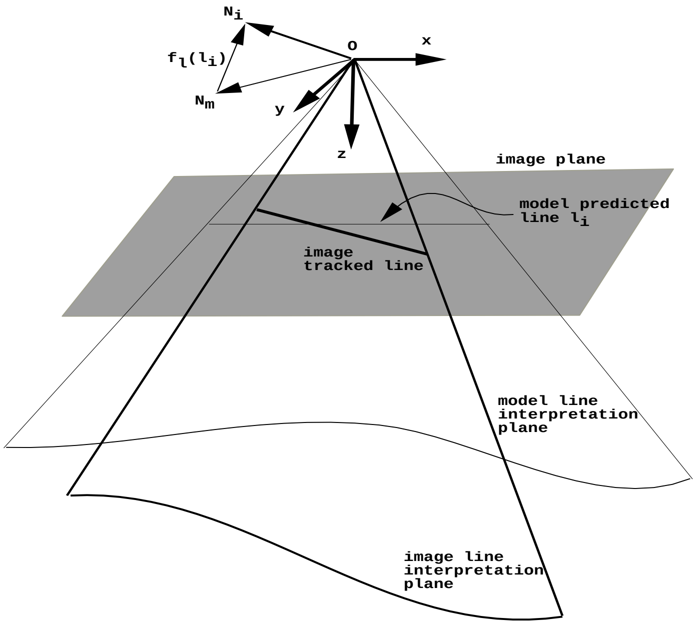


Figure 5: Line alignment error is based on the alignment of the their interpretation planes.

concatenate structure and motion parameters[5] as: $\mathbf{x}_l = (\mathbf{k}_l^\top, \mathbf{x}_r^\top)^\top$. The Jacobian of the line parameters with respect to the model parameters is a [6x13] matrix: $\mathbf{J}_{\mathbf{k}_l} = \frac{\mathbf{dk}_l}{\mathbf{dx}_l}$.

4.5 Line Cost

Given a MPL, as in (5), and a CITL, we define 2-D residual errors for their misalignment (see fig. 5). The transform in (14) maps a 3D line to the interpretation plane normal used for alignment. This involves the computation of a [3x6] Jacobian: $\mathbf{J}_N = \frac{\mathbf{dN}}{\mathbf{dk}_l}$. The Jacobian $\mathbf{J}_l = \frac{\mathbf{dN}}{\mathbf{dx}_l}$ for the transform that takes the line representation in the model frame, through the model rigid motion, into an interpretation plane normal, is computed via the chain rule using Jacobians $\mathbf{J}_{\mathbf{k}_l}$ and $\mathbf{J}_N$.

Given $\mathbf{N}_i$, the normal of the CITL interpretation plane, and $\mathbf{N}_m$, the normal of the MPL interpretation plane, the cost $f_{l_i}(\mathbf{x})$ corresponding to errors $\Delta\mathbf{N}_i(\mathbf{x}) = \mathbf{N}_i - \mathbf{N}_m$, over an ensemble of lines, is:

$$f_L = \sum_i f_{l_i}(\mathbf{x}) = \frac{1}{2}\sum_i \rho_i(\Delta\mathbf{N}_i(\mathbf{x})\mathbf{W}_i\Delta\mathbf{N}_i(\mathbf{x})^\top) \qquad (15)$$

The overall cost gradient and Hessian uses contour, intensity, and line observations *c.f.*

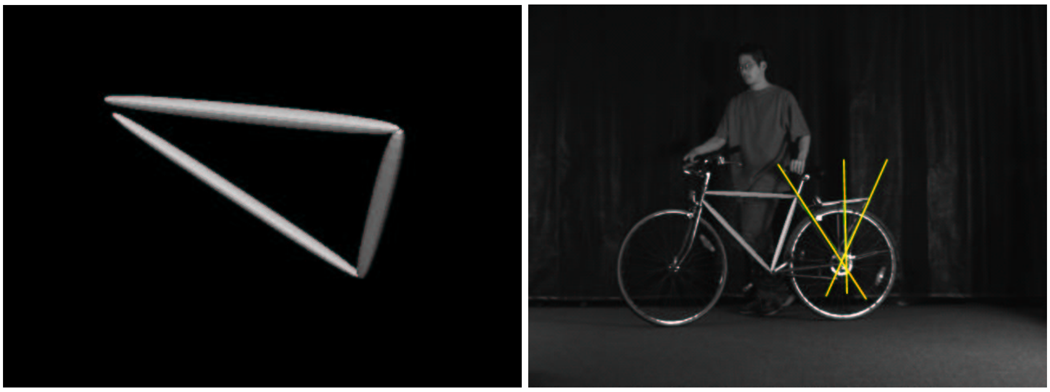
[5] These are parameters estimated with contributions from image line observations, but do not include the shape parameters $\mathbf{x}_d$ (see fig. 1). The vector thus has 7 parameters for the rigid motion (represented with quaternions for rotations) and 6 parameters for the 3D line.

(1), (2), (3), (4) and (15), respectively:

$$\mathbf{g} = \mathbf{g}_c + \mathbf{g}_I + \sum_i \mathbf{J}_{l_i}^{\top} \rho_i' \mathbf{W}_i \Delta \mathbf{N}_i \tag{16}$$

$$\mathbf{H} \approx \mathbf{H}_c + \mathbf{H}_I + \sum_i \mathbf{J}_{l_i}^{\top} (\rho_i' \mathbf{W}_i + 2\rho_i'' (\mathbf{W}_i \Delta \mathbf{N}_i)(\mathbf{W}_i \Delta \mathbf{N}_i)^{\top}) \mathbf{J}_{l_i} \tag{17}$$

The use of a robust error norm tolerates incorrect line hypotheses. For example, although initially detected as consistent (and included in the model), a line can be removed from the model if it persists being an outlier for a long period of time. Detecting such situations involves verifying whether the outlier's cost influence was suppressed by the robust norm during large time periods.



(a)

(b)

Figure 6: (a) Initial model. (b) Reconstructed model includes the initial chain of superquadric parts, modeling the central frame of the bike, and the incrementally reconstructed lines (in yellow), overlaid on the object in the image.

5 Experiments

The experiments we show consist of two *monocular* sequences, each containing 4 seconds of video (200 frames recorded at 50 fps) of a moving bike (fig. 9) and of a space robotics end-effector grapple fixture (fig. 11). Both sequences involve significant translational and rotational motion in the camera frame. Part of the bike structure is modeled and tracked using a model made of 3 pieces (fig. 6a), whereas the grapple fixture is modeled using a single superquadric (fig. 7a). The initial model shape and pose was provided manually in

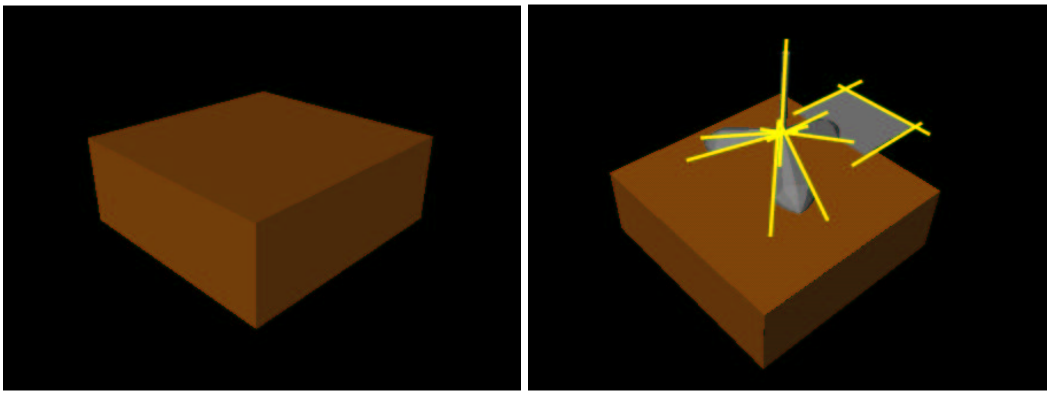


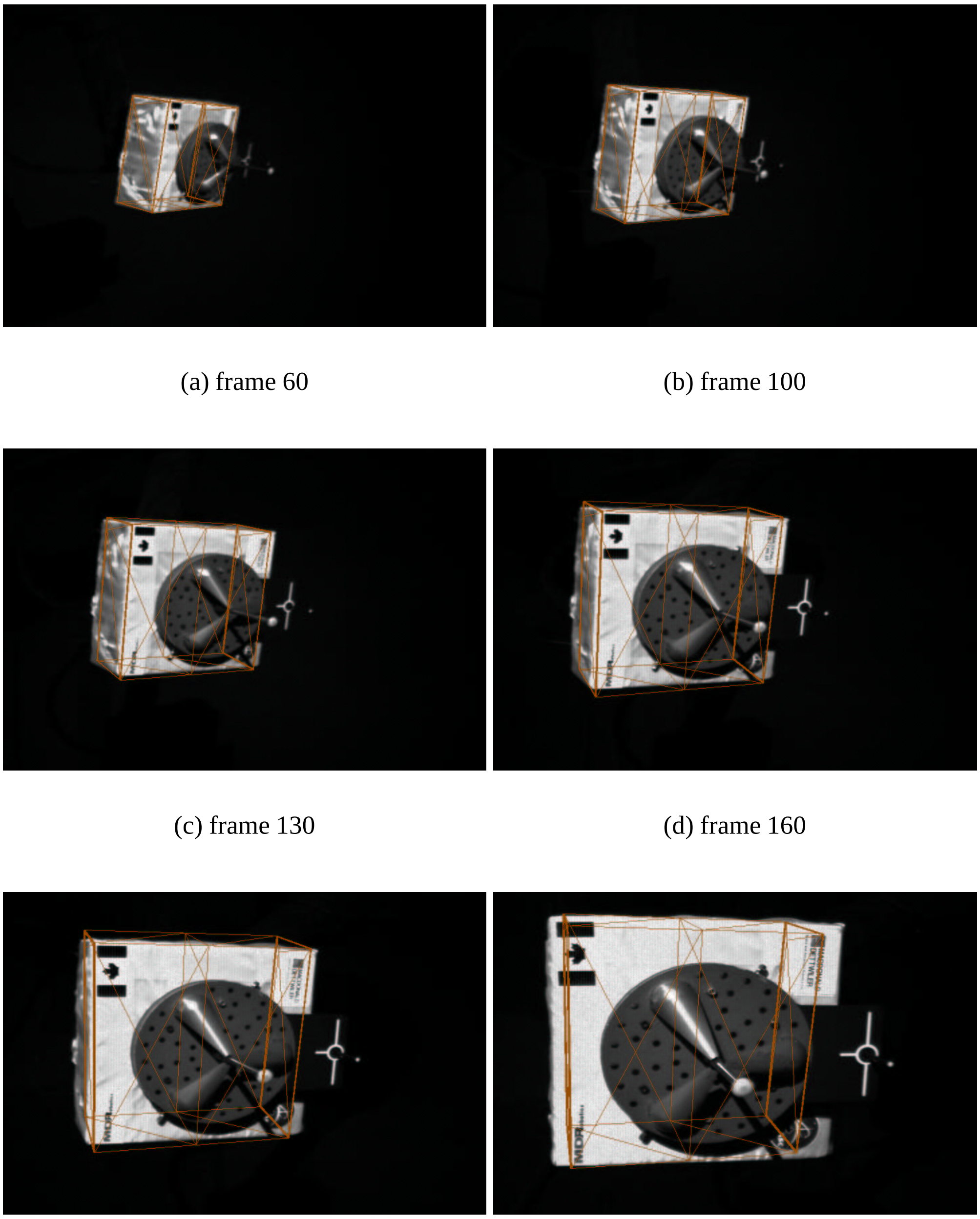
Figure 7: (a) Initial model. (b) Reconstructed model, with additional structures (in gray) shown in good alignment with the incrementally recovered lines.

all experiments. An initialization method based on aspect graphs [12, 11] could be used to automate the process, but we haven't pursued this here. The models are displayed overlaid and rendered flat-shaded gray (the bike sequence) and brown wireframe (the grapple fixture sequence), respectively[6]. The recovered models are shown in fig. 6b and fig. 7b, where the incrementally reconstructed lines appear to be in good alignment with the ridges of other non-modeled surfaces of the object.

In a first experiment, we try to track the grapple fixture using a non-adaptive model made of a superquadric with fixed parameters (fig. 8). Tracking fails due to the incorrect shape and pose initialization. The model appears well fitted in frame 60, but as the object moves, it becomes clear that the initialization was inaccurate. The model gradually drifts and ultimately fails to track in frame 208. Such situations can easily occur in many applications due to uncertain monocular initialization (*e.g.* due to viewpoint degeneracy), incomplete shape coverage, or partial occlusion.

We have also run tracking experiments using adaptive, incrementally growing models, on the bike and the grapple fixture sequences. In both of them, prior to new line reconstruction, we track using the initial part model and use contour and intensity measurements. The rest of the sequence (once CITLs have been identified) is tracked using the enhanced model, with new incrementally reconstructed lines. They augment the parameter space and provide additional alignment residuals between CITLs and MPLs. Lines not part of

[6] We use a fine, uniform superquadric re-tessellation for tracking[8], but display the curvature based one for better image visualization. The mesh appears sparse because its points are densely concentrated at sharp corners on the surface.



(a) frame 60

(b) frame 100

(c) frame 130

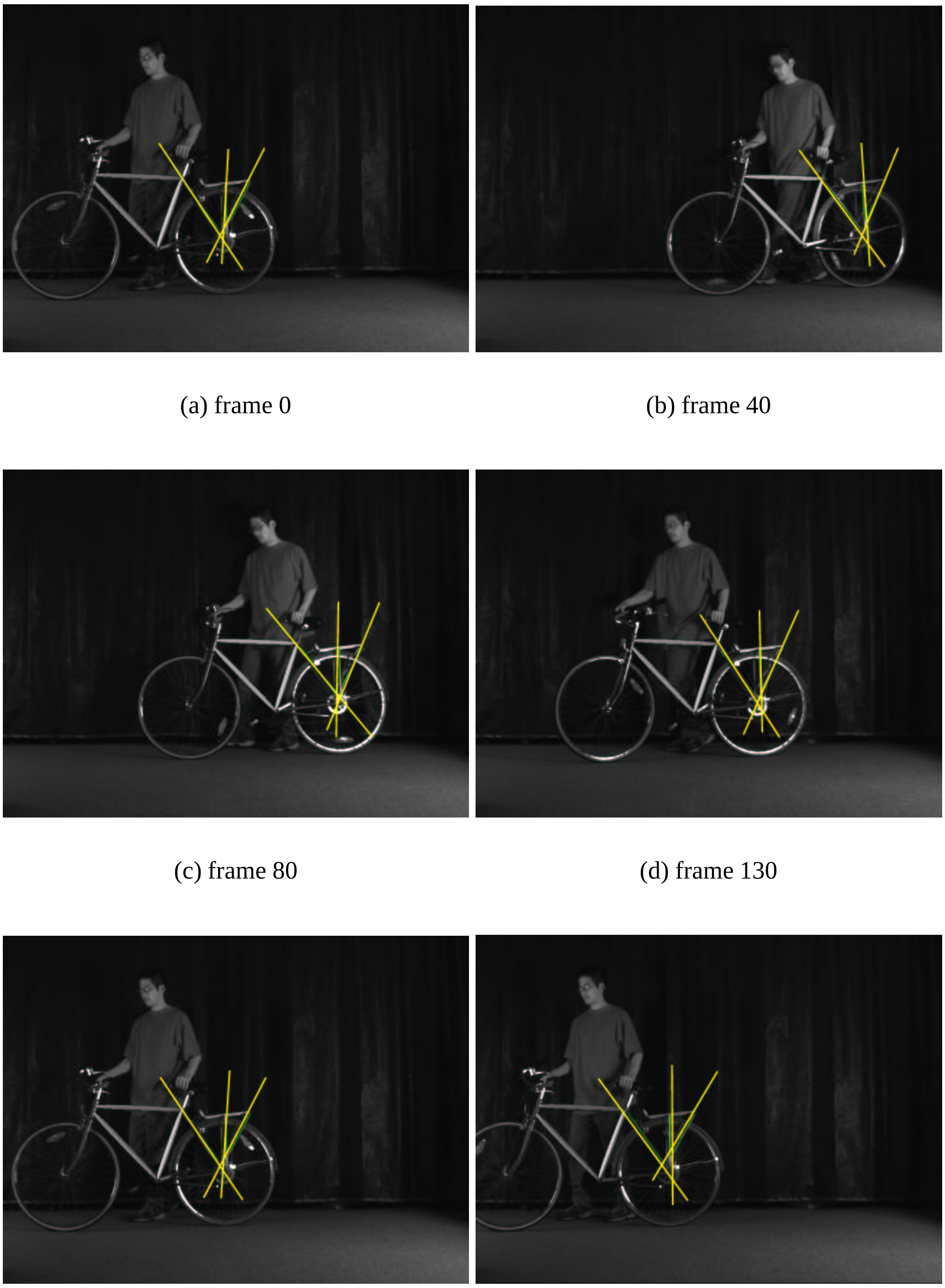
(d) frame 160

(e) frame 180

(f) frame 208

Figure 8: Using an uncertain, manually initialized non-adaptive model (brown wireframe) leads to tracking failure. Although initially the model appeared well fitted, the motion revealed that the shape was incorrect. The imperfect initialization and the use of a model that cannot dynamically adapt eventually lead to tracking failure.

the initial models are tracked using an independent line tracker, based on interest points with line fitting in each frame. The lines are detected using a method described in [6]. This



(a) frame 0

(b) frame 40

(c) frame 80

(d) frame 130

(e) frame 170

(f) frame 200

Figure 9: Tracking and augmenting a bike model (bike triangular frame, gray flat shaded) with additional lines. The MPLs are shown in yellow, the CITLs (green) are also visible on color plates. We show all the reconstructed lines (their MPL) in all frames, but lines are reconstructed incrementally (see text).

identifies edgels and their orientations, hypothesizes lines through them, and selects those having strongly supported edgels using RANSAC (see [6] for details).

It is important that the rigid motion estimation is accurate, because it affects both the validity of the consistency tests and the quality of line reconstruction. In practice, the models are initialized manually and their parameters are usually uncertain. Therefore, we neither test consistency nor reconstruct lines immediately after initialization, but allow a delay (about 20 frames) so that the models lock onto the data. Results for the runs are shown in fig. 9 and fig. 11. Lines determined as CITLs are plotted green while the model reconstructed and predicted lines are plotted yellow.

In the bike sequence, the linear reconstruction is based on 12 frames within the interval, 20-60. Although no lines are reconstructed until frame 20 in the bike sequence, their reprojection is displayed over the entire sequence. Because reconstruction uses a model-centered coordinate frame, the lines can be predicted backward in the initial image frames once the model motion has been estimated. In the grapple fixture sequence, lines are shown as they are incrementally tracked and recovered (thus, some are not visible initially). The consistency tests are performed several times for different groups of lines, in frames (20, 40, 60), (120, 140, 160) and (160, 175, 190). The tests are evaluated using a threshold $\tau = 0.05$ that worked well across the sequences we tried. The stability of the reconstruction is checked by the ratio of the 2nd and 3rd singular values associated with the matrix $\mathbf{A}$. This gives a principled criteria for deciding if a set of lines and their displacements support accurate reconstruction. Potential bias in the linear reconstruction is eliminated by re-estimating the lines jointly with all the rigid and non-rigid model parameters in a non-linear loop. Indeed, in both sequences (fig. 9 and fig. 11), the predictions from lines added to the model correctly align with the lines of the object in the image.

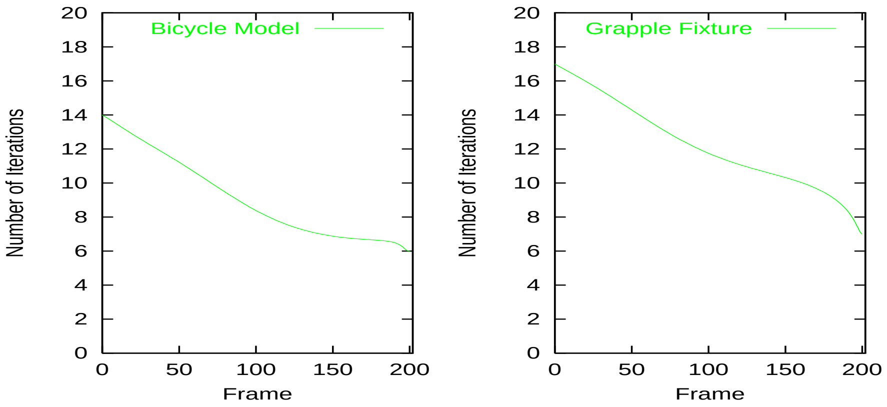


Figure 10: The number of iterations per frame (Bezier interpolation) decreases due to the addition of line constraints.

Tracking is more stable and accurate as more line features are reconstructed and used. In fig. 10 we give quantitative plots that show the decrease in the number of iterations of our non-linear optimizer as more line residuals are added to the model cost function. The average per-node model error decreases from 0.9 (initially) to 0.4 pixels (frame 60) in the bike tracking sequence, and from 1.2 pixels (initially) to 0.8 (frame 60), 0.6 (frame 160) and 0.2 (frame 190) in the grapple fixture sequence, respectively. Clear improvements are noticeable *e.g.*, for the difficult-to-track towards-camera motion at the end of the grapple fixture sequence.

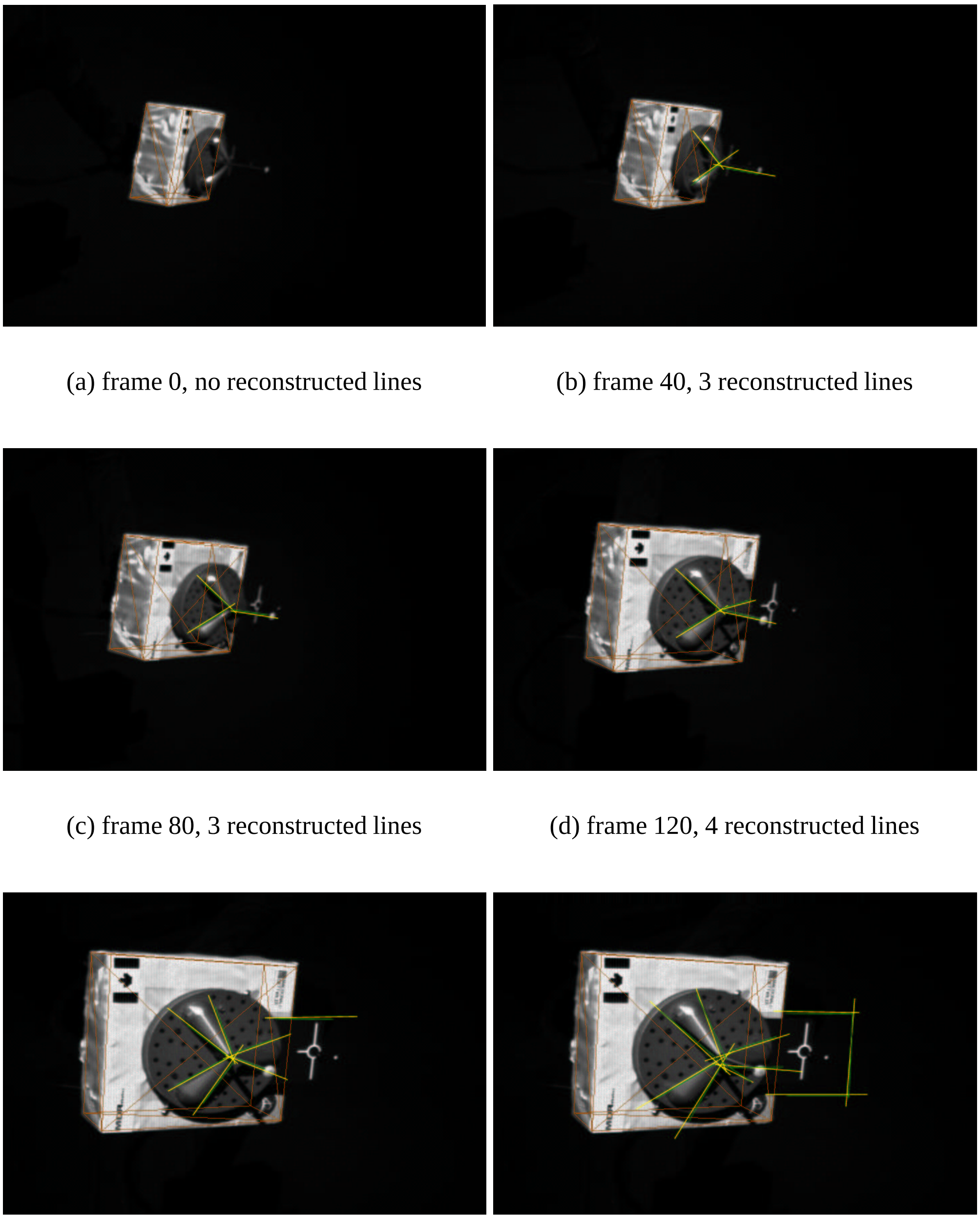
For the grapple fixture sequence, we also analyze the tracker failure modes with respect to inaccuracies in line detection and different cost error distributions. We performed tracking runs using costs based on Gaussian and Lorentzian error distributions [33]. We also simulated two noise levels in the line feature extraction, where we perturbed the tracked points by 2 and 4 pixels before line fitting[7]. These sets of experiments are identified as G2, G4, R2, R4 (G is for Gaussian, R for robust, and the digit gives the noise level in the feature extraction). We decide tracking failure by visual inspection at the frame where the model starts drifting from the object: *e.g.*, in fig. 8, this occurs around frame 135. We found G2=163, G4=147, R2=198, R4=175, with a clear performance advantage for the robust optimizer. The accuracy of line detection becomes critical towards the end of the sequence (frame 180), where small errors in the 3D rigid motion estimation can make the model width parameters highly uncertain (close to unobservable). This is caused by the incidental alignment of the model's depth axis with the camera ray of sight. In such degenerate cases, the estimated rigid motion is significantly improved by including new lines, provided these can be detected accurately.

6 Limitations and Future Work

The approach we have presented aims at flexibly modeling and tracking objects, but still has limitations that motivate future research:

1. We assume that at least a portion of the object can be modeled using simple volumetric parts, and that at least one of them can be coarsely fit using a parametric model. Previous work addressed the direct recovery of volumetric deformable models from 2-D images [10, 11]. That work assumed that image regions mapped to volumetric part surfaces, which turned out to be a strong assumption. More recent research has investigated techniques for model-based region merging [23].

[7] The variance of the cost error distribution is 1.5 in all experiments.



(a) frame 0, no reconstructed lines

(b) frame 40, 3 reconstructed lines

(c) frame 80, 3 reconstructed lines

(d) frame 120, 4 reconstructed lines

(e) frame 160, 7 reconstructed lines

(f) frame 190, 10 reconstructed lines

Figure 11: Grapple fixture model tracking (the grapple fixture model in wireframe, in brown) with MPLs (yellow). CITLs (green) are also visible on color plates. The model increases its scope during tracking and 10 additional lines, initially not modeled, are incrementally reconstructed and integrated into estimation by frame 190. The shape and the motion of all features (superquadric parameters, additional lines, rigid motion) are jointly estimated in a non-linear loop to avoid bias.

2. The consistency tests and the incremental structure added to the model is currently restricted to lines, whereas estimation applies to a model discretized with points and lines. The framework can be extended to planar surface patches and curves.

3. As discussed in §1, a long-term computer vision goal is flexible object modeling and qualitative shape recovery for recognition. The framework presented here improves the shape and motion estimation of some initially recovered volumetric model parts, and adds additional structure to them. Still, a representational gap exists between this structure, in the form of lines, and the volumetric parts, useful for qualitative modeling and recognition. It would be interesting to group the incrementally recovered features and recover volumetric part abstractions from them.

7 Conclusions

We have presented a framework for incremental model acquisition and tracking using parametric adaptive models. We relax the constraint that the model has to be entirely known a-priori, and enhance its basic discretization structure in terms of points and lines, but preserve a higher-level representation, in terms of parametric shapes. This allows a flexible use of model-based geometric consistency tests for incremental 3D line feature recovery and eliminates the need for minimal sets of feature correspondences that may often not be available for direct, bottom-up reconstruction. Tracking robustly combines linear and non-linear estimation techniques and augments the initial model-based cost function with new line measurements. We have experimentally demonstrated good reconstruction and tracking accuracy in two separate image domains, both involving objects with complex structure and motion.

Acknowledgments

The authors want to thank Bill Triggs for discussions and feedback, Doug DeCarlo for explanations on deformable models, Alexandru Telea for implementation assistance, and Piotr Jasiobedzki of MDRobotics, for the grapple fixture sequence. The authors gratefully acknowledge the support of NSERC, MDRobotics, CITO, NSF, and PREA.

References

[1] M. Armstrong and A. Zisserman. Robust Object Tracking. In *Asian Conference on Computer Vision*, 1995.

[2] A. Azarbayejani and A. Pentland. Recursive Estimation of Motion, Structure and Focal Length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.

[4] T. Cham and J. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 239–245, 1999.

[5] M. Chan, D. Metaxas, and S. Dickinson. Physics-based tracking of 3-D objects in 2-D image sequences. In *Proceedings, 12 International Conference on Pattern Recognition*, pages 326–330, Jerusalem, Israel, October 1994.

[6] J. Clarke, S. Carlsson, and A Zisserman. Detecting and tracking linear features efficiently. In *Proceedings of the 7th British Machine Vision Conference, Edinburgh*, 1996.

[7] J. Crowley, P. Stelmaszyk, T. Skordas, and P.Pugget. Measurements and Integration of 3-D Structures by Tracking Edge Lines. *International Journal of Computer Vision*, 1992.

[8] D. DeCarlo and D. Metaxas. Blended deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):443–448, April 1996.

[9] D. DeCarlo and D. Metaxas. Combining Information in Deformable Models Using Hard Constraints. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1999.

[10] S. Dickinson and D. Metaxas. Integrating Qualitative and Quantitative Shape Recovery. *International Journal of Computer Vision*, 1994.

[11] S. Dickinson and D. Metaxas. Using aspect graphs to control the recovery and tracking of deformable models. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(1):115–142, 1997.

[12] S. Dickinson, A. Pentland, and A. Rosenfeld. Shape Recovery using Distributed Aspect Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[13] T. Drummond and R. Cipolla. Real-time Tracking of Multiple Articulated Structures in Multiple Views. In *European Conference on Computer Vision*, 2000.

[14] D. Dube and A. Mitiche. The Incremental Rigidity Scheme for Structure from Motion: The Line-Based Formulation. In *European Conference on Computer Vision*, pages 292–296, 1990.

[15] R. Fletcher. Practical Methods of Optimization. In *John Wiley*, 1987.

[16] P. Fua and G. Leclerc. Taking Advantage of Image-Based and Geometry-based Constraints to Recover 3-D Surfaces. *Computer Vision and Image Understanding*, 1996.

[17] R. Hartley. Lines and Points in Three Views and the Trifocal Tensor. *International Journal of Computer Vision*, 1997.

[18] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[19] T. Heap and D. Hogg. Wormholes in Shape Space: Tracking Through Discontinuities Changes in Shape. In *IEEE International Conference on Computer Vision*, pages 334–349, 1998.

[20] K. B. Horn. Relative Orientation. *International Journal of Computer Vision*, 1990.

[21] T. S. Huang and A. N. Netravali. Motion and Structure from Feature Correspondences: A Review. *Proc. of the IEEE*, 1994.

[22] K. Kanantani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, 1996.

[23] Y. Keselman and S. Dickinson. Generic Model Abstraction from Examples. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2001.

[24] D. Lowe. Three-dimensional Object Recognition from Single Two-dimensional Images. *Artificial Intelligence*, 31(3), 1987.

[25] D. Lowe. Fitting Parameterized Three-dimensional Models to Images . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[26] D. P. McReynolds and D. Lowe. Rigidity Checking of 3D point Correspondences under Perspective Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[27] A. Mitiche and J. Aggarwal. Line-based Computation of Structure and Motion using Angular Invariance. In *IEEE Workshop on Motion*, 1986.

[28] A. Pentland and B. Horowitz. Recovery of Non-rigid Motion and Structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[29] A. Pentland and S. Sclaroff. Closed Form Solutions for Physically-based Shape Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[30] D. Samaras and D. Metaxas. Incorporating Illumination Constraints in Deformable Models. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.

[31] S. Seitz. Implicit Scene Reconstruction from Probability Density Functions. In *DARPA Image Understanding Workshop*, 1998.

[32] A. Shashua. Algebraic Functions for Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[33] C. Sminchisescu. *Estimation Algorithms for Ambiguous Visual Models–Three-Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*. PhD thesis, Institute National Politechnique de Grenoble (INRIA), July 2002.

[34] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[35] C. Sminchisescu and B. Triggs. Mapping Minima and Transitions in Visual Models. *International Journal of Computer Vision*, 61(1), 2005.

[36] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The case of Superquadrics with Local and Global Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

[37] M. Spestakis and J. Aloimonos. A Multi-frame Approach to Visual Motion Perception. *International Journal of Computer Vision*, 1991.

[38] R. Szeliski and S. B. Kang. Recovery 3-D Shape and Motion from Image Streams using Non-linear Least-squares. Technical report, DEC TR, 1994.

[39] C. Taylor and D. Kriegman. Structure and Motion from Line Segments in Multiple Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.

[40] D. Terzopoulos and D. Metaxas. Dynamic 3D models with Local and Global deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[41] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on Deformable Models: Recovering 3-D Shape and Non-rigid Motion. *A.I.*, 36(1), 1988.

[42] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[43] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In Springer-Verlag, editor, *Vision Algorithms: Theory and Practice*, 2000.

[44] S. Ullman. Maximizing Rigidity: The Incremental Recovery of 3-D Structure From Rigid and Nonrigid Motion. In *Perception*, volume 13, pages 255–274, 1984.

[45] T. Vieville and O. Faugeras. Feed-forward Recovery of Motion and Structure From a Sequence of 2-D Line Matches. In *IEEE International Conference on Computer Vision*, 1990.

[46] B. Yen and T. Huang. Determining 3-D Motion and Structure of a Rigid Body Using Straight line Correspondences. In *ASI NATO Series*, 1983.