Temporal Accumulation of Oriented Visual Features

Nicolas Pugeault[a], Norbert Krüger[b]

[a] *Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, United Kingdom.*
email: n.pugeault@surrey.ac.uk
[b] *Mærsk Mc-Kinney Møller Institute, Syddansk Universitet, DK-5230 Odense, Denmark.*
email: norbert@mmmi.sdu.dk

Abstract

In this paper we present a framework for accumulating on-line a model of a moving object (e.g., when manipulated by a robot). The proposed scheme is based on Bayesian filtering of local features, filtering jointly position, orientation and appearance information. The work presented here is novel in two aspects: First, we use an estimation mechanism that updates iteratively not only geometrical information, but also appearance information. Second, we propose a probabilistic version of the classical n-scan criterion that allows us to select which features are preserved and which are discarded, while making use of the available uncertainty model.

The accumulated representations have been used in three different contexts: pose estimation, robotic grasping, and driver assistance scenario.

Keywords: Object model building; visual representation; feature tracking; temporal filtering.

1. Introduction

This article presents a framework for on-line generation of an internal representation of unknown objects or scenes, that are observed by the system while subjected to motion. The proposed method is generic and can be applied to any feature. Also, it allows the correction over time of not only feature location, but also appearance information. In contrast, the state-of-the-art focuses on the accumulation of feature position only, while assuming the invariance of the feature's appearance; this invariance does not hold when objects are fully rotated. Moreover, this framework provides a complete representation of objects' edges structure, that makes it useful for a variety of visual as well as robotic tasks—as illustrated in section 4.

In a first step, local contour descriptors are extracted from the image and reconstructed in 3D using stereopsis.[1] The model itself encodes the object's contours directly in 3D, and associate to them appearance information such as colour. The scene's contours are encoded in this representation as strings of local features called 3D–primitives, that provide a first representation of the 3D shapes in the scene, enriched with appearance information. The appearance information has the quality of being robust under viewpoint changes, and therefore is used to improve matching reliability. At this stage, the representation is merely a collection of 3D–primitives, objects and background are not segmented in any way. By using the motion knowledge

[1] Alternatively, shape-from-motion could be used to obtain 3D–primitives. One additional complexity with this alternative is that the reconstruction uncertainty and the motion uncertainty are then related. In this work we focus on stereopsis as it allows for a simpler formulation.

provided either by a robot or a separate motion estimation[2], we segment the object from the scene (by selecting primitives that move according to the robot arm motion) and accumulate the representation. Having control over the object provides a very accurate knowledge of its motion that can be used to track individual 3D–primitives. At each frame, new observations are used to correct the 3D–primitives' full pose and to enrich the representation with new aspects of the object (e.g., parts that were previously occluded).

The mechanism presented herein improves the 3D object model obtained from stereo reconstruction in three respects:

1. Accuracy: The representation is corrected over time using new observations.
2. Reliability: Tracking primitives over time, it is possible to re-evaluate their reliability over time, and to discard erroneous ones. Since the tracking is done in 3D space, the likelihood for erroneous primitives to be tracked successfully is vanishingly small.
3. Completeness: Through Manipulation of the object, the system witnesses it under a wide range of viewpoints, and accumulates $2\frac{1}{2}D$ representations into a full $3D$ representation.

This framework requires the capacity to track features over time, and to correct their position using several frames. This is an essential problem in

[2]In this work we will mainly show results using motion extracted from the robot arm, for simplicity, but it could also be applied to visually computed motion (as in Fig. 10C and [31]), as long as an estimate of the motion error can be computed. The rationale behind using known motion is that it simplifies the problem and allows a better interpretation of the accumulation error irrespectively of motion estimation accuracy.

computer vision, and solutions belongs to two groups:

The first group consists of the geometric analytic solutions, including multi-focal tensors [10] and bundle adjustment [40]. These approaches provide optimal solutions to the problem and are prominent for solving the batch structure from motion (SFM) scenario. They can be designed to be robust to erroneous data association (see [40] for a discussion). One major problem of these solutions stems from the fact that they are fundamentally *batch* processes: all views of the object need to be simultaneously available. This can make the problem intractable for large sequences, and implies a large delay for any active system. It has been proposed to split the problem into groups of, e.g., 3 frames, reducing both delay and computational cost [23]. Nonetheless, these approaches face the dead-reckoning problem: small motion errors accumulate over time to lead to large localisation errors. Therefore, they generally need an additional global integration stage. David Nistér [24] proposed a live SFM approach based on pre-emptive RANSAC. Although the method is real-time it enforces strong constraints on feature disparity, and is limited to the estimation of feature position.

The second group uses various flavours of the Bayesian filtering theory. This provides an on-line solution by formalising the problem as a Markov process where the state vector combines both the current pose and the visual features' bearing. This can be formalised as the general Bayesian tracking problem—see [1] for a review. This theoretical formulation allows for an optimal solution, i.e., a Kalman filter [13], if the state vector has a multivariate normal distribution and if the prediction and observation processes are linear. In the mobile robotics context, the object whose model is being incremen-

tally built is the environment itself, described as a set of landmarks. The Kalman filter and its non-linear derivatives (e.g., extended Kalman filter) have been used extensively to solve the so-called simultaneous localisation and map-building (SLAM) problem (see, e.g., [5, 42, 8, 39, 21]). Andrew Davison [3] proposed a real-time monocular SLAM approach based on EKF. Also Monte Carlo Markov Chains have been used for tracking of multiple targets [15, 16, 44]. Tao et al. proposed a Bayesian approach for 2D motion segmentation in videos [38].

Because of the on-line constraint, the approach presented in this paper belongs to the second category. One essential difference to typical SLAM systems, is the large number of local features that the system needs to be able to track, to describe the object's shape completely, and the relatively low distinctiveness of these features, whereas SLAM applications generally rely on few sparse yet very distinctive features (e.g., SIFT [22]). Because of this large number of features, we will track each feature individually instead of maintaining a large joint covariance matrix. Moreover, we will track the full pose of the features as well as their appearance properties to make use of temporal information to improve both the accuracy of these appearance cues, but also to generate an estimate over time of their reliability—i.e., how invariant they are when the object is manipulated. The knowledge of this invariance is critical for object recognition and pose estimation. For example, for pose estimation, invariant cues are important for matching, whereas pose-dependent ones are important for estimating the pose.

The novel aspects of this work are:

- full feature vector tracking: we make use of Unscented Kalman Fil-

tering (UKF) [12] to track the distribution in the whole feature space, instead of only considering the feature's position. This includes the feature's orientation in space and the observed colour on both sides of the edge. This allows us to keep track of the relative reliability of different components of the feature vector by their filtered variance. It also allows for a straightforward extension to other feature types such as, e.g., junctions (see, [37]) or surface patches.

- probabilistic matching of features based on both geometric and appearance information.

- temporal re-evaluation of a feature's confidence according to the tracking success, and probabilistic argument for deletion or preservation of features during occlusions.

The framework is described in section 2, then evaluated on different scenarios in section 3. Applications making use of these representations are described in section 4 before we conclude in section 5.

2. Methods

In this section, we present the vision framework used to accumulate objects models. First in section 2.1, we will describe the local features that we use in this work. Note that the framework is generic, and could be applied to any local feature that defines a full pose and some appearance information. Then section 2.2.5 defines the state space based on such features. Section 2.3 discusses the feature tracking and filtering scheme, based on Un-

scented Kalman Filtering (UKF). Finally, section 2.4 discusses the confidence re-evaluation and the probabilistic $n-$scan criterion.

2.1. 3D line features extraction

In this work we use sparse image descriptors called visual *primitives*, that exist both in 2D and 3D space, and were discussed in [20, 27, 33]. In the 2D space, those primitives provide a condensed representation of image information sparsely sampled along image contours. In a first stage, linear and non-linear filtering operations are applied to the image (see, e.g., [11]). These filtering operations provide local information such as the likelihood that a pixel is on an edge, the orientation of this edge, the phase (that contains the type of contrast transition, see [17]) and the colour. Primitives are first extracted at contours and form a feature vector containing the edge position with sub-pixel accuracy, the local orientation, phase (contrast transition), colour on both sides of the edge and optic flow. Positions are detected sparsely with sub-pixel accuracy at places likely to contain edges (see, e.g., [11] for a description). In the following, we refer to such features as *2D–primitives*.

Such 2D–primitives are extracted on stereo pairs of images and are matched using the epipolar line and similarity constraints (see Fig. 1B, and [30] for an assessment). Pairs of matched 2D–primitives provide enough information to reconstruct the 3–dimensional equivalent of a 2D–primitive, denoted *3D–primitive* in the following (see Fig. 1C). We direct the reader to [6, 10] for a description on classical stereo reconstruction and [27, 33] for the special case of primitives.

A 3D–primitive encodes a scene contour's local position and orientation

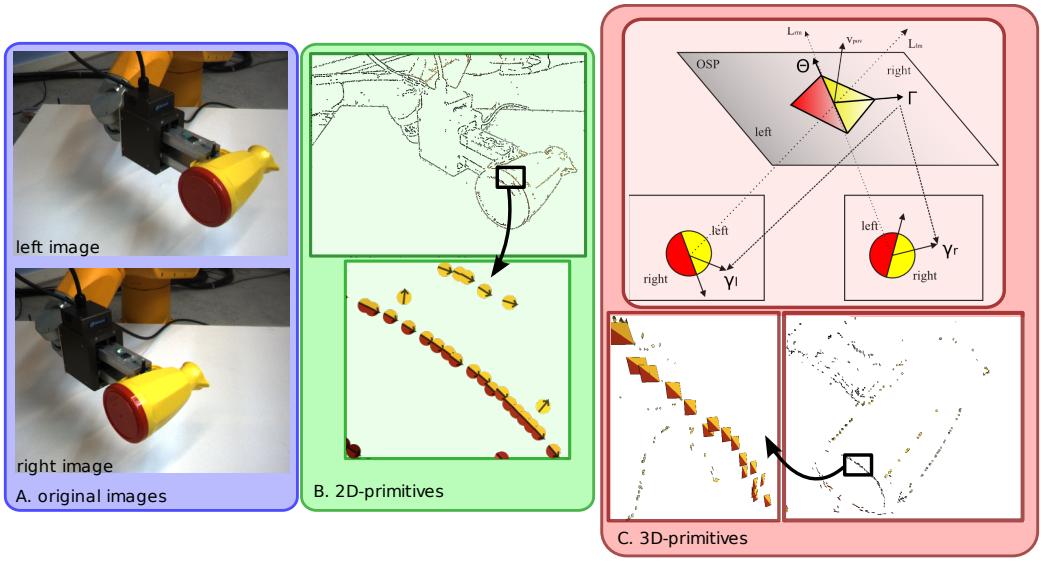


Figure 1: Illustration of the primitive extraction stage. A) original images (left and right); B) extraction of 2D–primitives; and C) stereo reconstruction of 3D–primitives.

along with the local contrast and colour on each side.

$$s = (\mathbf{p}, \omega, \mathbf{c}) \tag{1}$$

where $\mathbf{p}$ is the full 6D pose in space ; ω is the local phase ; $\mathbf{c}$ is a 6–dimensional vector encoding the RGB colour values on both sides of the contour. As a consequence, a 3D–primitive is encoded as a 13–dimensional feature vector.

A 3D–primitive's covariance is encoded as the 13×13 block-diagonal matrix Σ.

$$\Sigma_i = \begin{pmatrix} \Sigma_{G,i} & \\ & \Sigma_{A,i} \end{pmatrix} \tag{2}$$

Where $\Sigma_{G,i}$ is the uncertainty in the feature's 3D pose (geometric uncertainty) and $\Sigma_{A,i}$ is the uncertainty in the feature's appearance (appearance uncertainty). Geometric uncertainty is propagated from the 2D primitives'

position and orientation uncertainty, using geometric reconstruction; the 2D–primitives uncertainty is calculated from the uncertainty principle applied to the local filters used to extract and locate the primitives [33]. The derivation of the Jacobians for the 3D reconstruction of position and orientation is too long to be reproduced here, but it can be found in a technical report [28]. Appearance reconstruction uncertainty has been estimated using Monte-Carlo simulation on the available data.

Object shapes are described in this framework as a collection of 3D–primitives. Each feature will be represented by the triplet $\boldsymbol{Z}_i = \{s_i, \Sigma_i, B_i\}$ where s_i is the expected feature vector, Σ_i its covariance and $B_i \in [0, 1]$ the current belief that this feature belongs to the object.

2.2. Definition of the feature state space

The 3D–primitives described in the previous section contains two very different type of information. First a 3D–primitive's geometric information is encoded in the form of its location and its orientation, forming a full 6D pose. Note that because we are considering edge features, there is no exact definition of the rotation angle around the edge itself, as in the general case there is not one single plane where the edge is fully embedded. Indeed, in most cases edges will signify places in the image where either occlusions occur, or where two distinct surfaces meet. For this reason this last dimension is only estimated from the available viewpoints. This means that this value is expected to have high uncertainty, and is only used to preserve consistency between the appearance cues on both sides of the 3D-edge.

The space of all poses and all Rigid Body Motions (RBM) is called the Special Euclidean space of dimension 3 ($SE(3)$). This work uses dual-

quaternions to represent poses and motions in this space. Dual-quaternions have several advantages when representing $SE(3)$, as they allow for a compact formulation of a pose transformation under a RBM, combination of two RBMs, and RBM interpolation and blending. Moreover, they allow for a simple conversion to 6–dimensional representation of poses. We will first describe quaternions and their use to represent 3D rotations $SO(3)$ in section 2.2.1, then expose how this formulation can be extended to $SE(3)$ in section 2.2.2. The quaternion, and dual-quaternion representation of rotation and poses (respectively) provide a compact representation, allowing for efficient transformations. In contrast, Euler angles suffer from the gimbal lock problem, and transformation of 3×3 matrices require an additional step to ensure that the resulting matrix indeed defines a rotation. Section 2.2.3 then discusses how concretely we will encode a 3D–primitive's pose into a state vector, and section 2.2.4 how the appearance cues are treated. Section 2.2.5 formalises the complete state vector we use in this work.

2.2.1. Quaternions

All 3D rotations lie on the surface of a sphere called the Spatial Orthogonal Space of dimension 3, written $SO(3)$. Because of this unusual topology, this space differs from Euclidean spaces in several properties, and therefore rotations are not aptly represented by mere vectors: for example, the average between two vectors designating points on the surface of a sphere will not lie on the sphere's surface itself. Rotations could also be represented using Euler angles, which face the *Gimbal lock* problem: certain angles will lead to a degenerate state where certain motions become impossible. It is also possible to represent 3D rotations by 3×3 matrices, but this implies the use of 9

parameters instead of the required 3, and has the drawback that operations between rotations such as blending can lead to matrices representing invalid 3D rotations. Quaternions [9], on the other hand allow a compact and efficient representation of 3D rotations, and allow for an efficient blending and combination of 3D rotations without the risk of generating degenerate states.

Quaternions are an extension of complex numbers, composed of one real part and three imaginary parts:

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k = (q_0, \mathbf{v}) \tag{3}$$

for convenience, we will write

$$\Re[\mathbf{q}] = q_0 \tag{4}$$

$$\Im[\mathbf{q}] = \mathbf{v} = [q_1, q_2, q_3]^\top \tag{5}$$

where $\Re[\mathbf{q}]$ is called the real part and $\Im[\mathbf{q}]$ the imaginary part of the quaternion $\mathbf{q}$. The conjugate of quaternion $\mathbf{q}$ is given by:

$$\mathbf{q}^* = (q_0, -\mathbf{v}) \tag{6}$$

The unit-quaternion

$$\mathbf{q} = (\cos(\alpha/2), \sin(\alpha/2)\mathbf{r}) \tag{7}$$

describe a rotation of angle α around a rotation axis $\mathbf{r}$ (with $\|\mathbf{r} = 1\|$). Note that all unit quaternions, such that $\mathbf{q}\mathbf{q}^* = 1$ represent valid rotations in SO(3), The similarity with Euler axis-angle notation allows for straightforward transformation between the two notations.

Similarly, the combination of two rotations encoded by $\mathbf{q}_1$ and $\mathbf{q}_2$ is given by:

$$\mathbf{q}_{12} = \mathbf{q}_1 \mathbf{q}_2 \tag{8}$$

Also, efficient algorithms exist for interpolating and blending in $SO(3)$ using quaternions (SLERP [36]).

2.2.2. Dual-quaternions

Dual-quaternions [43] are an extension of quaternions that extend quaternion properties and formulations to the Special Euclidian space of dimension 3 ($SE(3)$), the space of all RBMs:

$$SE(3) \cong \mathbb{R}^3 \times SO(3) \tag{9}$$

A dual-quaternion consists two quaternions, a real part and a dual part:

$$\check{\mathbf{q}} = \mathbf{q}_R + \epsilon \mathbf{q}_\epsilon \tag{10}$$

such that $\epsilon^2 = 0$ and $\mathbf{q}_\epsilon = iq_4 + jq_5 + kq_6$. A RBM is represented by a unit dual-quaternion such as:

$$\check{\mathbf{q}}\check{\mathbf{q}}^* = 1 \tag{11}$$

where $\check{\mathbf{q}}^* = \mathbf{q}_R^* - \epsilon \mathbf{q}_\epsilon^*$ is the conjugate of $\check{\mathbf{q}}$.

The equivalent translation and rotation Euler axis-angle representation can be recovered from a dual-quaternion $\check{\mathbf{q}}$ by the equations:

$$\alpha = 2 \arccos \Re[\mathbf{q}_R] \tag{12}$$

$$\mathbf{r} = \frac{1}{\sin \frac{\alpha}{2}} \Im[\mathbf{q}_R] \tag{13}$$

$$\mathbf{t} = \Im[2\mathbf{q}_\epsilon \mathbf{q}_R^*] \tag{14}$$

designing a rotation of angle α around an axis $\mathbf{r}$ and a translation of $\mathbf{t}$.

The advantage of this notation is that it allows for convenient formulation of points and lines [7, 35] transformation under a RBM:

$$\mathbf{p}' = \check{\mathbf{q}}\mathbf{p}\check{\mathbf{q}}^* \tag{15}$$

The combination of two RBMs $\check{\mathbf{q}}_1$ and $\check{\mathbf{q}}_2$ is given by:

$$\check{\mathbf{q}}_{12} = \check{\mathbf{q}}_1 \check{\mathbf{q}}_2 \tag{16}$$

Some studies have shown that RBM can be efficiently interpolated in dual-quaternion notation (e.g., [14]).

2.2.3. Feature pose encoding

In this work we encode a feature's pose as the dual-quaternion $\check{p}_t$, that aligns the coordinate system's z–axis to the feature's orientation, and the y–axis to its normal.

One advantage of this formulation is that we can use Eq. 16 to model a feature's pose transformation according to the RBM described by the dual-quaternion $\check{\mathbf{m}}_{t,t+1}$ as:

$$\check{\mathbf{q}}_{t+1} = \check{\mathbf{m}}_{t,t+1} \cdot \check{\mathbf{q}}_t \tag{17}$$

Therefore, a RBM transformation can be computed as a simple product of two dual-quaternions. The geometric covariance is computed from classical line reconstruction formulae Jacobians (see, e.g., [6]).

2.2.4. Apperance cues

The geometric information captures the feature's pose information. For the purpose of matching and tracking features, it is required to also have pose-independent information that we will call *appearance* information. In our case, we will encode the contrast transition across the edge as the local phase ω (see [17]), and the colour information on both sides of the edge. The phase encodes the type of contrast transition across the edge in a 2π periodic continuum between $(-\pi, +\pi]$, where a bright line on a dark background is

encoded by a phase of 0, a dark line on a bright background by π; a bright to dark edge by $-\pi/2$ and a dark to bright edge by $+\pi/2$. For the colour information, the CIE $L^*a^*b^*$ colour space is chosen as it offers two qualities for our purpose. First, it is designed to be a perceptually uniform colour space, and therefore, distances in this space correspond to the distances perceived by humans. Second, the lightness is encoded by the L component, allowing to reduce illumination effects on the two other components. Therefore, the colour information on both sides of the edge is encoded as two Lab vectors $\mathbf{c} = (L_1, a_1, b_1, L_2, a_2, b_2)$. It follows that the appearance vector is 7–dimensional.

$$\mathbf{A} = (\omega, L_1, a_1, b_1, L_2, a_2, b_2) \tag{18}$$

Alternatively, other type of appearance cues could be used (e.g., texture). The initial covariance of a reconstructed 3D-primitive's appearance vector depends on the camera properties and the illumination conditions, and was therefore evaluated a priori by tracking thousands of features. Figure 2 shows the standard deviation of features' appearance over time when the object was moving. Each bar in the graph shows the standard deviation of one appearance component. For example, phase has a low deviation whereas luminance has a strong deviation—due to illumination.

2.2.5. State vector

The resulting state vector s is a 13–dimensional vector describing jointly geometric and appearance information:

$$s \;=\; (\mathbf{t}, \alpha\mathbf{r}, \mathbf{A}) \tag{19}$$

In this equation, $\mathbf{t}$ is the translation vector, $\mathbf{r}$ the rotation axis and α the rotation angle (see Section 2.2.2). The covariance Σ is 13×13 and block di-

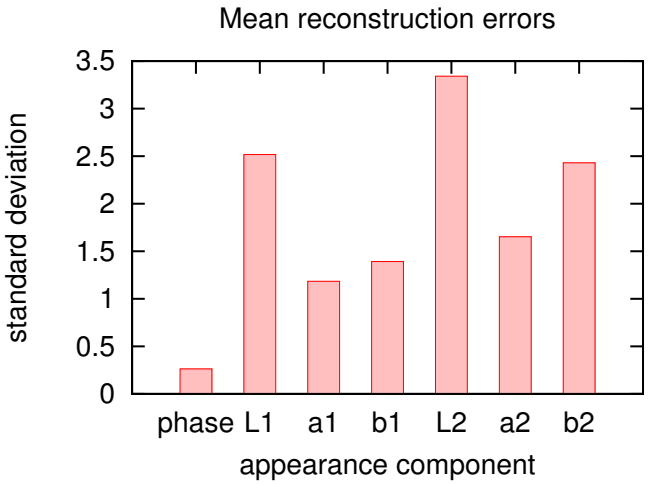


Figure 2: Appearance attribute uncertainties recorded as standard deviations. This was obtained by tracking all features over time and evaluating the mean standard deviation over all features.

agonal, composed of the geometric and appearance covariance matrices. The geometric covariance is computed from the covariance of the 2D primitives and the projection matrices (as detailed in [29]), the appearance covariance is estimated by Monte Carlo simulation processed by adding a small amount of noise to the original images.

2.3. Feature tracking and filtering

In this work we are interested in filtering full 6D poses. Because RBMs in dual-quaternion representations are non-linear operations, we need a non-linear implementation of Bayesian filtering. For these reasons we chose to use an Unscented Kalman Filter (UKF) [12] to track and filter the features' state vectors over time. The UKF is an extension of the classical Kalman Filter [13] to non-linear operations, that estimates posterior distributions by transforming a small set of specifically chosen points according to a non-

linear function $f(s)$. Because the distribution is assumed to be normal, only few points are required to estimate it, and this makes the UKF one order of magnitude faster than particle filtering. The UKF has been shown to be more consistent than the EKF when estimating non-linear functions [12]—this is especially the case for functions with strong non-linearities like the orientation prediction. Moreover, it does not require the computation of the prediction function's Jacobians.

2.3.1. Prediction

In our case, the non-linear prediction function $f(s)$ is fully defined by the RBM imposed by the robot.

$$s_{t+1} = f(s_t) = \begin{cases} \check{\mathbf{p}}_{t+1} &= \check{\mathbf{q}}_{t,t+1}\check{\mathbf{p}}_t \\ \mathbf{A}_{t+1} &= \mathbf{A}_t \end{cases} \tag{20}$$

The motion operates on the geometric part of the state vector, while leaving the appearance part unchanged. This is consistent with the assumption of Lambertian surfaces that is widely used in multiple-view computer vision. Note that a more complex illumination model could be handled by this framework, as it allows for non-linear prediction models.

Because this transformation is non-linear, there is no simple analytic formulation to transform the multivariate distribution formed by the feature's estimate and uncertainty. One classical way to circumvent this difficulty would be to use the first order Taylor expansion of the function f as a linear approximation—as is done, for example, in the Extended Kalman Filter (EKF). The problem with this simplification is that linearising a non-linear function can lead to large approximation errors, especially when the the function has strong non-linearities (like is the case for 3D rotation, for example).

This leads to inconsistent estimations of the posterior covariance that may lead to slow convergence (or even no convergence at all) of the EKF approaches. This sensitivity would also reduce the generality of the approach to locally nearly linear features. Consequently, we make use of another, more accurate method that is called the Scaled Unscented Transform, and has the triple advantage of being generic, accurate, and not requiring the derivation of Jacobians—see [12] for a comparison with the EKF.

2.3.2. Scaled Unscented transform

We make use of the Scaled Unscented Transform (SUT) to estimate the predicted state's covariance. The SUT allows to predict the transformation of a normal distribution by a non-linear process f. This is done by selecting a specific set of sample points from the distribution, and transforming them according to $f(s)$. This is different from particle filtering in the sense that only a small number of specially chosen samples are necessary to estimate a normal distribution, whereas particle filtering requires a large number of samples to estimate generic distributions.

In practice, the unscented transform requires $2n + 1$ samples $\mathcal{S}_i$ with associated weights W_i, for predicting a state $\hat{\Pi}$ of dimension n.

$$\begin{array}{rcll} \mathcal{S}_0 & = & s_{t-1} & W_0^m = \frac{\lambda}{n+\lambda} \\ & & & W_0^c = W_0^m + (1 - \alpha^2 + \beta) \\ \mathcal{S}_i & = & s_{t-1} + \left[\sqrt{(n+\lambda)\Sigma_{t-1}}\right]_i & W_i^m = W_i^c = \frac{1}{2(n+\lambda)} \\ \mathcal{S}_{i+n} & = & s_{t-1} - \left[\sqrt{(n+\lambda)\Sigma_{t-1}}\right]_i & W_{i+n}^m = W_{i+n}^c = \frac{1}{2(n+\lambda)} \end{array} \tag{21}$$

In these formulae, $[A]_i$ refers to the i^{th} column of the matrix A, and the covariance matrix square root is calculated using Cholesky decomposition.

Note that different weights are used for computing the mean (W^m) and the covariance (W^c). The weighting of the points is used to sample very nearby points and thereby avoid non-local effects. The parameter λ is defined as

$$\lambda = \alpha^2(n + \kappa) - n \tag{22}$$

where $\kappa = 0$, $\alpha = 0.01$ and $\beta = 2$ (as we have a Normally distributed prior). We refer to [41] for a discussion of those parameters.

From these weighted samples, it is possible to evaluate the mean ($\hat{s}_t$) and covariance ($\hat{\Sigma}_t$) of the predicted normal distribution:

$$\hat{s}_t = \sum_{i=0}^{2n} W_i^m f(\mathcal{S}_i) \tag{23}$$

$$\hat{\Sigma}_t = \Sigma_P + \sum_{i=0}^{2n} W_i^c \left[f(\mathcal{S}_i) - \hat{s}_t\right] \cdot \left[f(\mathcal{S}_i) - \hat{s}_t\right]^\top \tag{24}$$

where $\Sigma_P = \sigma_P \mathbf{I}$ is an estimate of the RBM prediction error, set to a diagonal matrix with a small value σ_P ($\sigma_P = 0.01$). This value need to be set according to the precision of the motion estimates. This step allows us to predict the state vector of a primitive after the RBM $f(s)$.

2.3.3. Matching

The SUT allows to predict a model of the object. The next step of the filtering consists in comparing this predicted model with the observed features. To this end, the predicted features $\hat{\boldsymbol{Z}}_i$ are compared with the newly observed ones $\tilde{\boldsymbol{Z}}_j$ by reprojecting them in both image planes and matching their projection using the Mahalanobis distance. Compared to the classical Euclidean distance, the Mahalanobis distance has the advantages that it takes into

account feature uncertainty and correlations between feature components; therefore, it is a more robust measure in cases where feature uncertainty is strongly anisotripic, like for reconstructed 3D features. Moreover, the Mahalanobis distance of a multivariate normal distribution is itself distributed according to a χ^2 distribution, which provides a theoretically sound distance threshold for matching. A newly observed 3D–primitive $\tilde{\boldsymbol{Z}}_j$ is matched with a predicted 3D–primitive $\hat{\boldsymbol{Z}}_i$ if both of their projections in both frames are matched according to a χ^2 criterion applied onto their Mahalanobis distance.

$$(\hat{s}_i - \tilde{s}_j)^\top (\hat{\Sigma}_i + \tilde{\Sigma}_j)^{-1} (\hat{s}_i - \tilde{s}_j) < \chi^2_{k=13,p=0.05} \tag{25}$$

In this equation $\chi^2_{k=13,p=0.05}$ indicates the $p = 0.05$ value in the χ^2 distribution of dimension 13; because the Mahalanobis distance has a χ^2 distribution, that guarantees that Eq. 25 will select 95% of the correct matches.

In this case, likelihood of the match μ_t in each projected frame is evaluated using a normal distribution centred on the predicted primitive.

$$p\left[\mu_t(\hat{\boldsymbol{Z}}_i, \tilde{\boldsymbol{Z}}_j)\right] = \frac{\exp\left[-\frac{1}{2}(\hat{s}_i - \tilde{s}_j)^\top (\hat{\Sigma}_i + \tilde{\Sigma}_j)^{-1} (\hat{s}_i - \tilde{s}_j)\right]}{(2\pi)^{n/2}\sqrt{|(\hat{\Sigma}_i + \tilde{\Sigma}_j)|}} \tag{26}$$

If the χ^2 criterion is not met, we define that
$p\left[\mu_t(\hat{\boldsymbol{Z}}_i, \tilde{\boldsymbol{Z}}_j)\right] = 0$.

It may happen that several observed features match an accumulated one, notably when the accumulated feature's covariance is large. This will happen for example when an object is moved closer to the camera: the predicted covariance will be large, and cover several newly observed features. In this case, the most likely match (according to Eq. (26)) one is preserved in a

winner-take-all fashion:

$$p\left[\mu_{i,t}\right] = \max_j p\left[\mu_t(\hat{\boldsymbol{Z}}_i, \tilde{\boldsymbol{Z}}_j)\right] \tag{27}$$

In this case, the other observed features are still considered as matched, and will not be added to the representation.

2.3.4. Correction

Once the matching is done, the set of model features $\hat{\Pi}_t$ can be corrected from the newly observed features $\tilde{\Pi}_t$ using a straightforward Kalman filtering approach. In this case, the equations for the Kalman filter's correction stage are simplified by the fact that both predicted and the observed states lie in the same feature space. Therefore, the error vector between predicted and observed states is:

$$\Delta s = \tilde{s} - \hat{s} \tag{28}$$

The innovation matrix is given by:

$$D = \hat{\Sigma} + \tilde{\Sigma} \tag{29}$$

From those, the classical Kalman equations provide us with the optimal gain:

$$K = \hat{\Sigma} \cdot D^{-1} \tag{30}$$

Finally the posterior distribution's mean is:

$$s = \hat{s} + K \cdot \Delta s \tag{31}$$

and its covariance:

$$\Sigma = (I - K) \cdot \hat{\Sigma} \tag{32}$$

2.4. Confidence re-evaluation

The model is updated in a final step that is the combination of three mechanisms: the first one updates confidence in each individual feature according to how well it has been matched at this frame; the second adds newly observed features that were not matched in the model; and the third discards model features that have not been sufficiently confirmed by observations.

The rationale between this scheme is that a number of erroneous features are expected to be observed at each time-step, mainly due to incorrect stereo-correspondences. Also, it is expected that correct features may occasionally fail to be observed, or even become occluded for extended periods of time. Therefore we need a mechanism to weed out erroneous features while preserving correct ones. This is classically achieved by the so-called $n-scan$ *criterion* in the Multi Hypotheses Tracking literature (see, e.g., [34]). In this work we will make use of the probability $p\left[\mu_{i,t}\right]$ to replace this criterion by a statistical argument.

Effectively, the confidence in a feature's correctness is evaluated as a measure of how consistently it has been observed over a length of time. The complete matching history of an accumulated primitive is denoted as: $\mu_{i,t}^{*} = \{\mu_{i,1}, \ldots, \mu_{i,t}\}$

We propose a recursive formulation that allows us to update features' confidence $p\left[\boldsymbol{Z}_i\right]$ at each time-step according to how well the predicted feature matched the observation (see Fig. 3). A straightforward application of Bayes

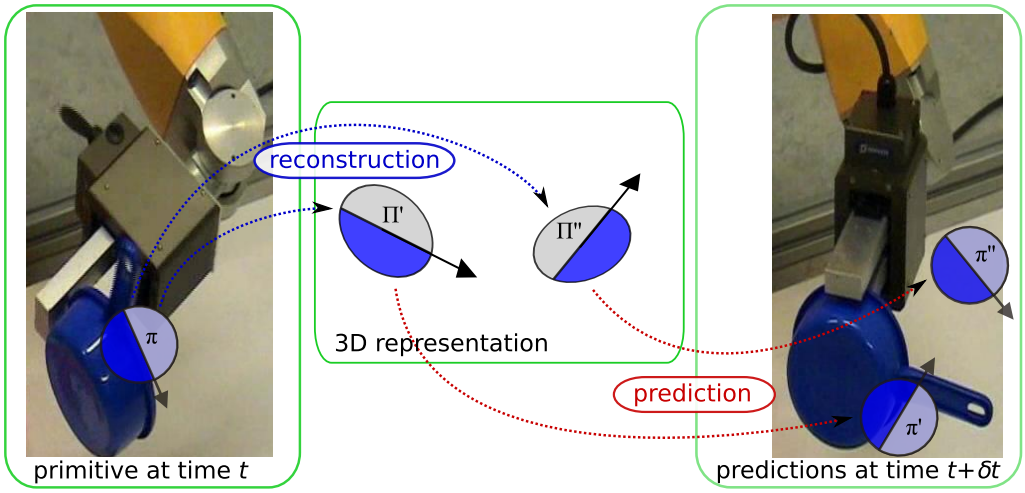


Figure 3: Illustration of the primitives temporal matching.

theorem provides us with the following formulation for a feature's probability:

$$p\left[\boldsymbol{Z}_i|\mu^*_{i,t}\right] = \frac{p\left[\mu^*_{i,t}|\boldsymbol{Z}_i\right]p\left[\boldsymbol{Z}_i\right]}{p\left[\mu^*_{i,t}|\boldsymbol{Z}_i\right]p\left[\boldsymbol{Z}_i\right]+p\left[\mu^*_{i,t}|\bar{\boldsymbol{Z}}_i\right]p\left[\bar{\boldsymbol{Z}}_i\right]} \tag{33}$$

$$p\left[\boldsymbol{Z}_i|\mu^*_{i,t}\right] = \left(1+\frac{p\left[\mu^*_{i,t}|\bar{\boldsymbol{Z}}_i\right]p\left[\bar{\boldsymbol{Z}}_i\right]}{p\left[\mu^*_{i,t}|\boldsymbol{Z}_i\right]p\left[\boldsymbol{Z}_i\right]}\right)^{-1} \tag{34}$$

If we assume the independence of the successive observations, we can write the probability of their joint occurence as the product

$$p\left[\mu^*_{i,t}|\bar{\boldsymbol{Z}}_i\right] = \prod_t p\left[\mu_{i,t}|\bar{\boldsymbol{Z}}_i\right] \tag{35}$$

$$p\left[\mu^*_{i,t}|\boldsymbol{Z}_i\right] = \prod_t p\left[\mu_{i,t}|\boldsymbol{Z}_i\right] \tag{36}$$

From this, we can rewrite Eq. 33 as:

$$p\left[\boldsymbol{Z}_i|\mu^*_{i,t}\right] = \left(1+\frac{\prod_t p\left[\mu_{i,t}|\bar{\boldsymbol{Z}}_i\right]p\left[\bar{\boldsymbol{Z}}_i\right]}{\prod_t p\left[\mu_{i,t}|\boldsymbol{Z}_i\right]p\left[\boldsymbol{Z}_i\right]}\right)^{-1} \tag{37}$$

This formulation has the inconvenience that it requires to record all primitives' complete matching history $\mu^*_{i,t}$. This is impractical for an on-line al-

gorithm, and therefore we derive a recursive formulation. From Eq. 34, we reformulate the confidence computation as:

$$p\left[\boldsymbol{Z}_{i,t}|\mu^*_{i,t}\right] = (1+\zeta_{i,t})^{-1} \ , \tag{38}$$

where ζ is evaluated recursively by

$$\zeta_{i,0} = 1/p\left[\boldsymbol{Z}\right] - 1 \tag{39}$$

$$\zeta_{i,t} = \frac{p\left[\mu|\bar{\boldsymbol{Z}}_i\right]}{p\left[\mu_{i,t}|\boldsymbol{Z}_i\right]}\zeta_{i,t-1} \tag{40}$$

with $p\left[\boldsymbol{Z}\right]$ is the prior probability that an observed 3D–primitive is correct, and $p\left[\mu|\bar{\boldsymbol{Z}}_i\right]$ is the prior probability for an erroneous observation. Although the exact value of $p\left[\boldsymbol{Z}\right]$ is essentially object dependent and cannot be estimated or calculated for an unknown object, we found experimentally that changes in this value impact only slightly on convergence speed. Therefore, we set it to a small value ($p\left[\boldsymbol{Z}\right] = 0.2$ for all experiments). The value of $p\left[\mu|\bar{\boldsymbol{Z}}_i\right]$ depends directly on the quality of the matching process, and can be theoretically set to the p–value of χ^2 criterion used for matching (in our case: $p\left[\mu|\bar{\boldsymbol{Z}}_i\right] = 0.05$). This parameter influence convergence speed and the impact of weak matches, and can potentially be tuned to reduce outliers or improve completeness. The value is set to $p\left[\mu|\bar{\boldsymbol{Z}}_i\right] = 0.05$ for all experiments.

.

2.4.1. Hypotheses acceptance and discarding: Probabilistic N-scan criterion

If an hypothesis' confidence $p\left[\boldsymbol{Z}_{i,t}|\mu^*_{i,t}\right]$ falls below a threshold τ^-, then it is deemed erroneous and discarded; if it raises above a threshold τ^+, then it is deemed verified up to certainty, and its confidence is not updated any more. This allows for the preservation of features during occlusion. This

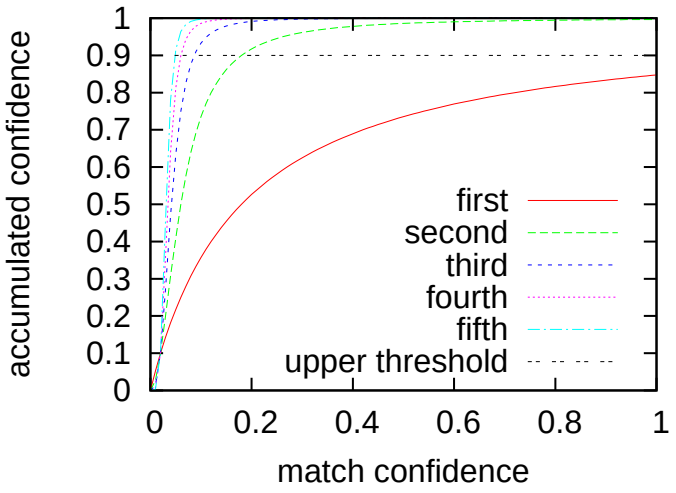


Figure 4: Confidence update for repeated observations of the same match quality. The $x-$axis shows the quality of the match and the $y-$axis the resulting updated confidence, for the five first updates. The dashed line shows the threshold we used for these experiments, i.e. $\tau_{\max} = 0.9$. Above this value, the hypotheses were considered as confirmed.

is effectively a soft version of the classical $n-$scan strategy in tracking [34]. The $n-$scan criterion is used to limit the tracking complexity when allowing for multiple data associations. It states that a feature should be kept if it has been matched successfully n times, discarded otherwise. In our case, the equivalent n value depends on how well the feature is matched in each frame. If the feature is roughly matched, and the uncertainty is high, it will require more frames to validate the association (i.e., a larger n). Conversely, if the feature is accurately matched, it will be validated quickly (i.e., small n). Fig. 4 shows the confidences accumulated after repeated observations of a

primitive. In this graph, the $x-$axis records the match quality.[3], the $y-$axis shows the corresponding accumulated confidence, and the different curves illustrate the confidences from first to the fifth observations. The dashed line shows the level of the upper threshold $\tau^+ = 0.9$ that we chose for those experiments. Any hypothesis whose accumulated confidence rises above this threshold is deemed to be correct and preserved in memory.

2.4.2. Adding new features to the model

Observed primitives $\tilde{\boldsymbol{Z}}_j$ that were not matched with any of the prediction

$$p\left[\mu_t(\hat{\boldsymbol{Z}}_i, \tilde{\boldsymbol{Z}}_j)\right] = 0, \ \forall \hat{\boldsymbol{Z}}_i \tag{41}$$

are then added to the representation to enrich it. Thereby the representation becomes progressively more complete, whereas erroneous hypotheses are discarded. This is illustrated in Fig. 5.

3. Results

In this section we demonstrate the system's performance using an artificial OpenGL sequence and real sequences of a robotic system manipulating a variety of toy kitchen utensils. The results are illustrated in terms of accuracy improvement and model building.

3.1. Pose correction

We evaluated the pose correction on an OpenGL rendered artificial sequence. The sequence features a rotating cube, which position and shape is

[3]For this figure (only) it is assumed that features are repeatedly matched with the same quality.

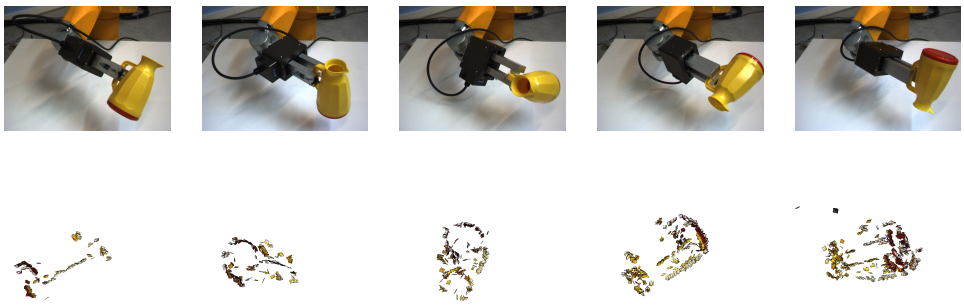


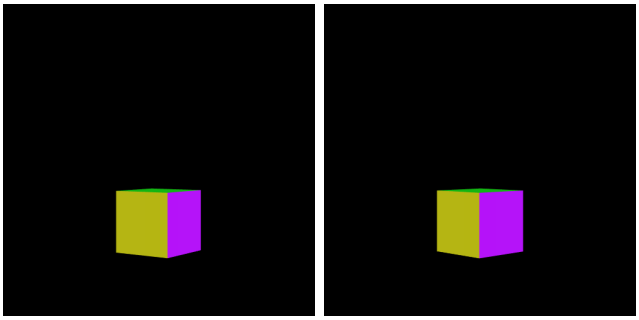
Figure 5: Accumulation of object model over time. Over 72 frames the object is rotated a full 360 degrees (5 degrees per frame) by the robot arm, offering a full 3D view of the object to the cameras. Features are progressively added to the representation as new aspects of the object become visible, while visible features are corrected and occluded features are preserved.

perfectly known. Therefore, each primitive is associated to the closest edge, and its position and orientation is compared to this edge's. The purpose of the artificial cube experiment is to assess whether in an idealized scenario the scheme can 1) improve reconstructed feature accuracy beyond what is allowed by stereo performed on sub-pixel features, despite the sparsity of the extracted 2D–primitives and the aperture problem.

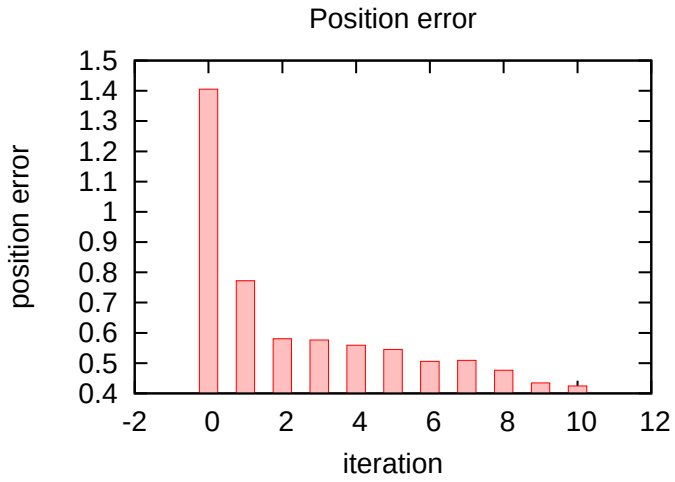
In Fig. 6 we can see the position and orientation errors recorded after several iterations of the process. This figure shows that the error in both position and orientation decreases quickly through UKF filtering.

3.2. Model building

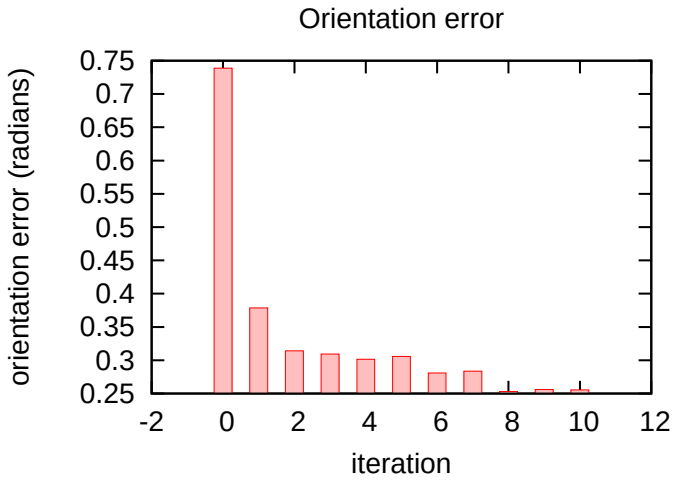
We evaluate our approach on a selection of objects, manipulated by a robot arm. Note that these objects were chosen to evaluate best the perfor-



(a) Two images from the sequence



(b) position correction



(c) orientation correction

Figure 6: Correction of the pose error after several iterations of the UKF filtering (the cube has an edge size of 10 units).

mance and limitations of the accumulation of edge features that we describe in this work. For this reason, they were chosen to have little texture, strongly curved surface and edges, occluding edges, and cast shadows.

Figs. 7 and 8 show the result of the accumulation scheme on several objects. The first row show one image of the objects. In the second row, the accumulated models are shown. It can be seen in these figures that the accumulated features describe well the contours of the object, including the occluded ones. In order to estimate the quality of the accumulated shape, we estimated the three eigenvalues of the accumulated primitives' position. These are recorded for each object in the last row. These graphs show that the object's shape become more accurately modelled with iterations of the process. For example, the pan generates two large eigenvalues and one small, indicating a flat object with little depth. The vase and the knife, on the other hand, have one main eigenvalue, indicating a flat object.

In Fig. 9 the convergence of the Kalman gain is illustrated by plotting the mean trace of the gain matrices for all accumulated primitives in a model against the number of iterations. The figure shows a clear convergence of the gain for most objects, with the notable exception of the 'pot' and 'pan' sequences. Convergence of the gain implies that the estimated pose and appearance of the primitives is converging over time. For this reason, the aperture problem prevents the gain from converging for objects that are very circular. In effect, for the 'pan' and 'pot' sequences, that contain mostly circular structures, the primitives follow a perpetual drift along the circular contour which however does not disturb the actual shape reconstruction process.

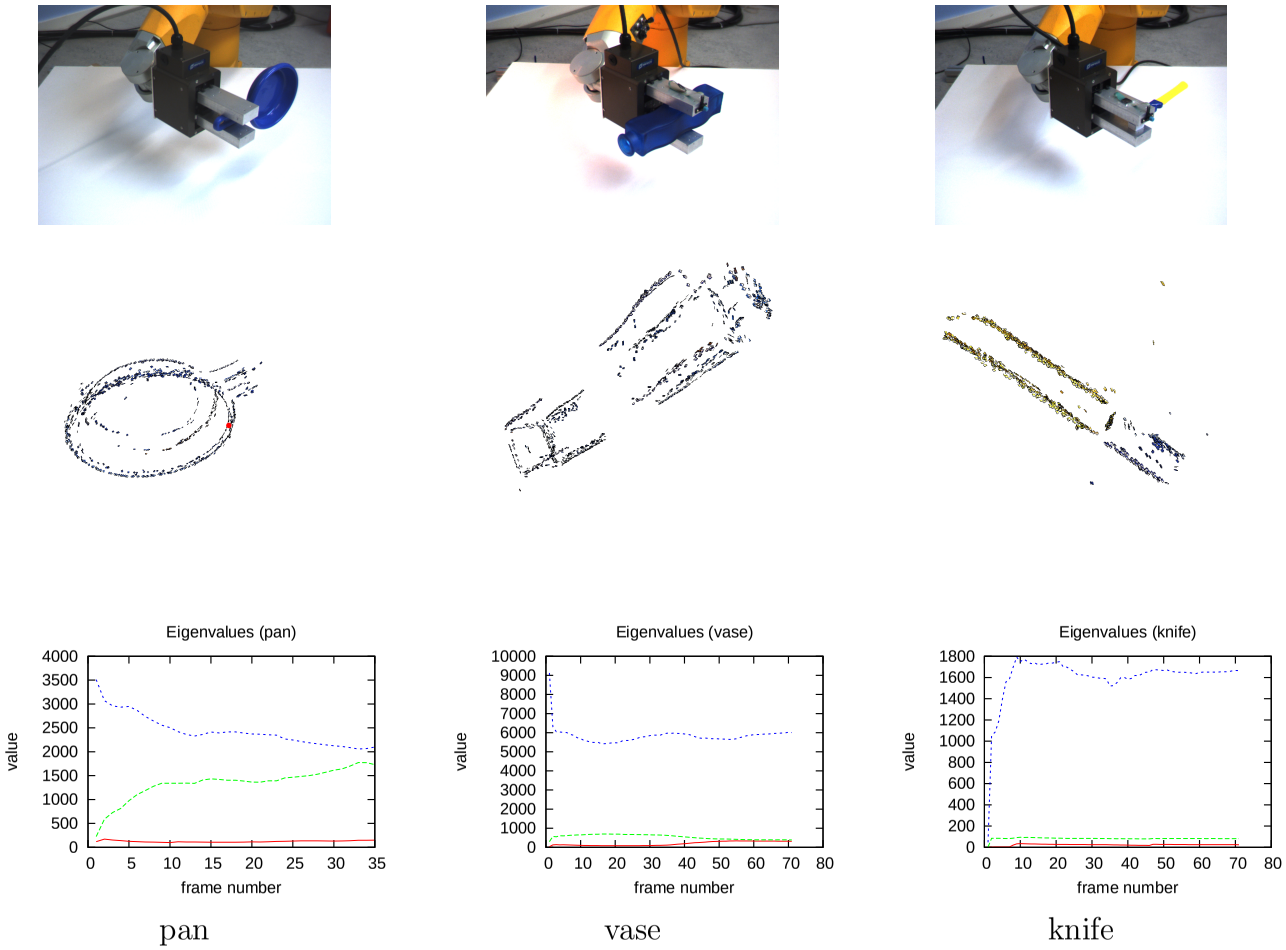


Figure 7: Accumulation results of several objects.

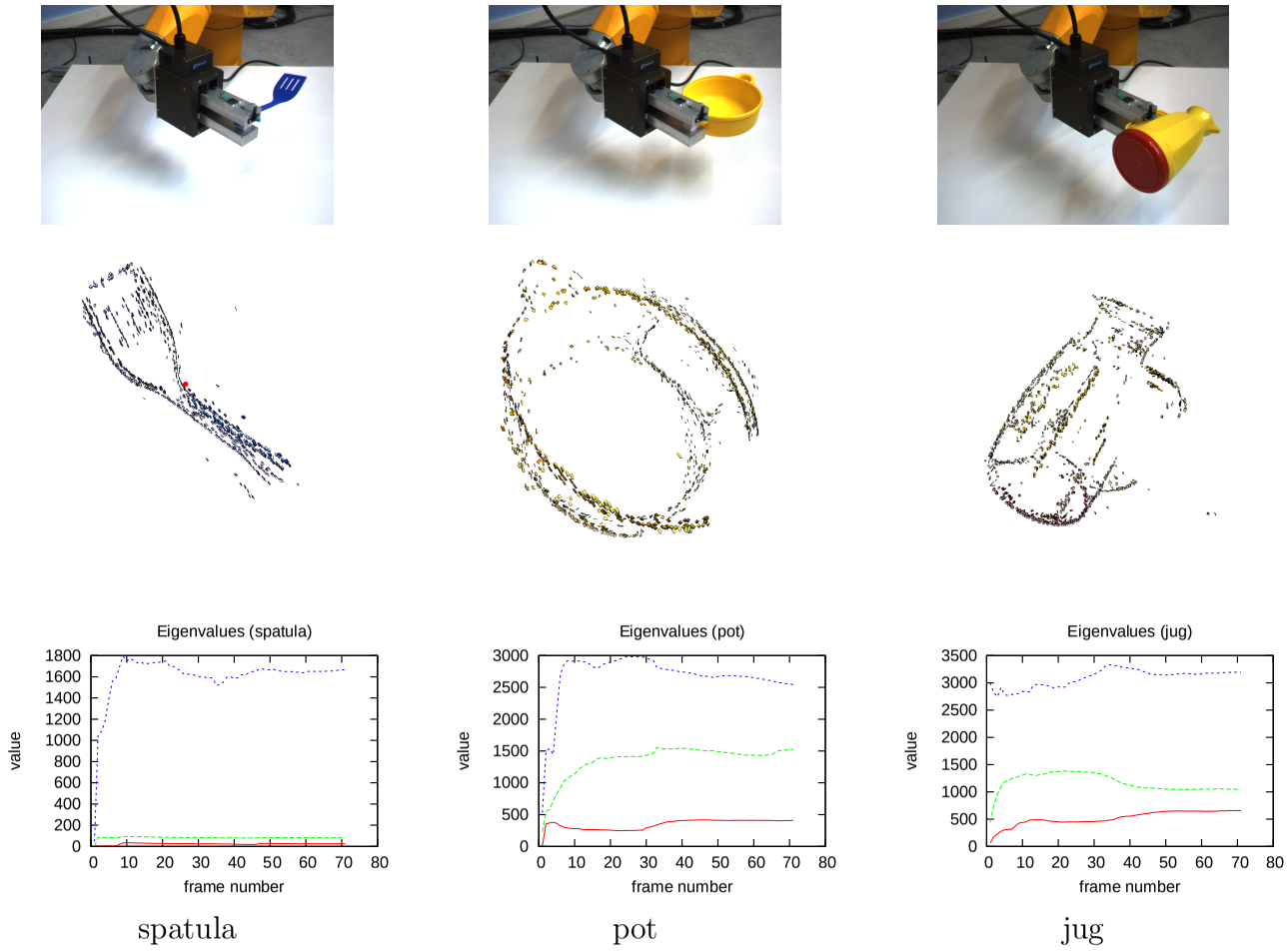


Figure 8: Accumulation results of several objects.

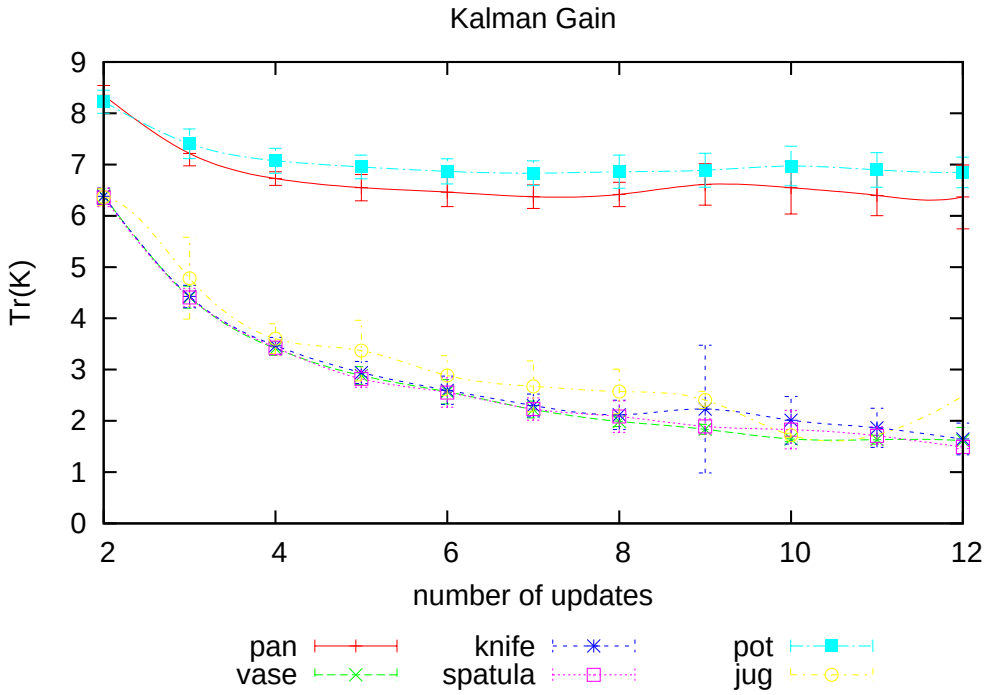


Figure 9: Convergence of the Kalman Gain for all sequences.

4. Applications

The algorithm described in this paper has been used in three different contexts. In the first application (briefly described in section 4.1), unknown objects become segmented by a cognitive robot-vision system and a multi-modal representation becomes computed. This learned representation has then been used for pose estimation and grasp learning as described in section 4.2. Finally, the algorithm has been used for the computation of large scale maps in outdoor environments: We demonstrate the flexibility of the extended Kalman filtering by accumulating additional information associated to primitives as for example their association to lane markers. This is described in section 4.3.

4.1. Birth of the Object

In the cognitive system [19], the accumulation algorithm has been used by a robot-vision system to acquire world knowledge in terms of the objectness of things and object shape. For this a premature grasping mechanism is used to get physical control over the object allowing it to move it in a controlled way. Then objects become constituted by the primitives accumulated according to the robot motion (subtracting the robot hand performing the very same motion). Since the motion is performed by the robot itself, based on the robot movements predictions of the change of visual features can be made and the objects become constituted by these features (see figure 5 and 10A).

4.2. Pose Estimation and Grasp Learning

The accumulated representations have been used for pose estimation in [4] in a hierarchical Bayesian net (see figure 10B i, ii). The fact that different

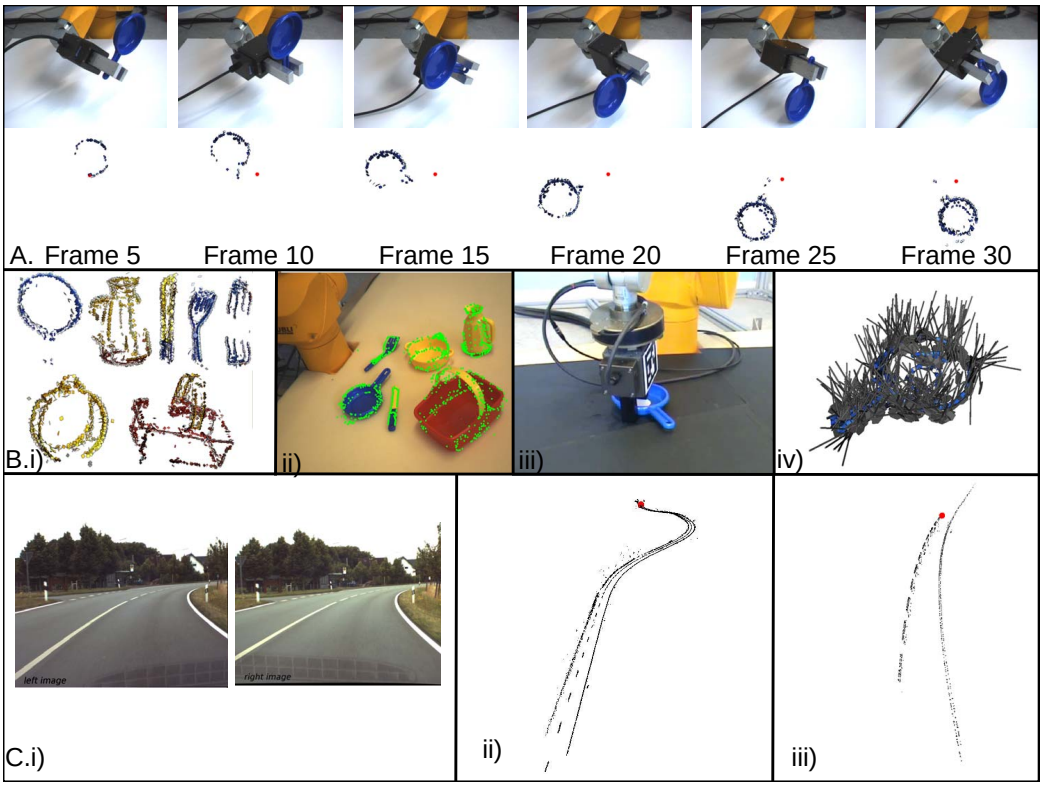


Figure 10: Applications. A) BoO: Show emerging object over different frames and the subtraction of gripper. B) Pose Estimation and Grasp Learning: i. Accumulated objects used for ii. pose estimation; iii. object grasping; and iv. storing of successful grasps. C) Driver assistance scenario: i. Stereo Image; ii. Outdoor map; and iii. accumulated lane.

aspects of visual information (position, orientation, and colour) have been accumulated makes the learned representation in particular powerful for such matching tasks. Once the matching has been performed successfully, further learning of grasping can be performed by letting the robot play with the object (try to grasp and let drop) and successful grasps can become associated to the accumulated representations (see [18] and figure 10B iii,iv).

4.3. Constructing Maps in Outdoor Environments

The accumulation algorithm has also been used in outdoor environments in a driver assistance scenario (see figure 10C,i). Here the motion has been computed based on image correspondences (for details, see, e.g., [25]). In addition to the confidence described in this paper (see Eq. 38), we accumulate confidences for a primitive to become associated to a lane extracted from individual stereo frames according to [2]. In figure 10C)-iii, primitives with high associated confidences for them to be part of lane markers.

5. Discussion

We presented in this article a framework for the on-line acquisition of models of objects and scenes subjected to a known motion. The framework presented makes use of a probabilistic formulation for matching, tracking, correcting and selecting features for the model. Moreover, we did filter and correct not only feature position but also orientation and appearance information. This provides us with a model of object structure based on a collection of edge descriptors covering geometric and appearance information. These extracted models have been used for tasks as diverse as grasping [26], object learning [19], motion estimation [32] and pose estimation [4].

Characteristic for our approach is that we accumulate edge information in terms of an abstracted representation by local multi-modal descriptors covering geometric and appearance information. In contrast to SLAM application, we are interested in accumulating aspect information of a large number of landmarks, and therefore cannot keep correlations between landmarks. This allowed to maintain a complete representation of the object's shape, rather than a subset of selected landmarks. In contrast to SFM approaches, the system is incremental and on-line, and does not require to memorise the full trajectory of features. The result is a powerful and rich representation ensuring its general applicability. Another aspect allowing for a flexible use of the algorithm is the ability to learn models on-line which is required in particular for robotic applications in which novel views appear in unpredicted ways.

In this work, we assumed that appearance information is unaffected by motion. Future research may attempt to include in the model the illumination impact on appearance information. Also the extension of our algorithm to other feature types (e.g., junctions or patchlets) is straightforward due to the flexibility of the unscented Kalman filter. For junction structures this has already been shown in [37].

Acknowledgements

This research was funded by the European project PACOPLUS (IST-FP6-IP-027657).

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non–Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[2] B. Boesman, L. Baunegaard With Jensen, E. Başeski, N. Pugeault, and N. Kruger. Bayesian reasoning using 3D relations for lane marker detection. *Proceedings of VMV 2009*, pages 127–134, 2009.

[3] A.J. Davison. Active search for real-time vision. In *Proceedings of the ICCV*, volume 1, pages 66–73, 2005.

[4] R. Detry, N. Pugeault, and J. Piater. A probabilistic framework for 3D visual object representation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 31(10):1790–1803, 2009.

[5] P. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Transactions in Robotics and Automation*, 17(3):229–241, 2001.

[6] O.D. Faugeras. *Three–Dimensional Computer Vision*. MIT Press, 1993.

[7] Y. Gu and J. Luh. Dual–number transformation and its application to robotics. *IEEE Journal of Robotics and Automation*, 3(6):615–623, 1987.

[8] J.E. Guivant and E.M. Nebot. Optimization of the Simultaneous Localization and Map–Building Algorithm for Real–Time Implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[9] W.R. Hamilton. *Lectures on Quaternions*. Royal Irish Academy, 1853.

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

[11] B. Jähne. *Digital Image Processing*. Springer, 2002.

[12] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A new approach for the nonlinear transformation of means and covariances in linear filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.

[13] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[14] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39 – 46, 2007.

[15] Z. Khan, T. Balch, and F. Dallaeat. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, 2005.

[16] Z. Khan, T. Balch, and F. Dallaeat. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 28(12):1960–1972, 2006.

[17] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.

[18] D. Kraft, R. Detry, N. Pugeault, E. Başeski, J. Piater, and N. Krüger. Learning objects and grasp affordances through autonomous exploration. In *International Conference on Computer Vision Systems (ICVS)*, pages 235–244, 2009.

[19] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger. Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. *Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics*, 5:247–265, 2009.

[20] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multimodal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour (AISB)*, 1(5):417–427, 2004.

[21] T. Lemaire, C. Berger, I-K. Jung, and S. Lacroix. Vision–Based SLAM: Stereo and Monocular Approaches. *International Journal of Computer Vision*, 74(3):343–364, 2007.

[22] D.G. Lowe. Distinctive Image Features from Scale–Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[23] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. 3D reconstruction of complex structures with bundle adjustment: An incremental approach. In *IEEE International Conference on Robotics and Automation*, pages 3055–3057, 2006.

[24] D. Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.

[25] F. Pilz, N. Pugeault, and N. Krüger. Comparison of point and line features and their combination for rigid body motion estimation. In *Sta-*

tistical and Geometrical Approaches to Visual Motion Analysis, LNCS 5604, pages 280–304. Springer, 2009.

[26] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. A strategy for grasping unknown objects based on co–planarity and colour information. *Robotic and Autonomous Systems*, 58(5):551–565, 2010.

[27] N. Pugeault. *Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation*. PhD thesis, Georg-August-Universität Göttingen, 2008.

[28] N. Pugeault, S. Kalkan, E. Başeski, F. Wörgötter, and N. Krüger. Reconstruction uncertainty and 3d relations. Technical Report 2007 - 6, Robotics Group, Maersk Institute, University of Southern Denmark, 2007.

[29] N. Pugeault, S. Kalkan, E. Baseski, F. Wörgötter, and N. Krüger. Reconstruction uncertainty and 3D relations. In *Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08)*, 2008.

[30] N. Pugeault and N. Krüger. Multi–modal matching applied to stereo. *Proceedings of the BMVC 2003*, pages 271–280, 2003.

[31] N. Pugeault, K. Pauwels, M. Van Hulle, F. Pilz, and N. Krüger. Multi–level architecture for detection and tracking of independently moving objects. In *Proceedings of the Int. Conf. on Computer Vision Theory and Applications (VISAPP'10)*, 2010.

[32] N. Pugeault, F. Wörgötter, , and N. Krüger. Rigid body motion in an early cognitive vision framework. In *Proceedings of the IEEE Systems, Man and Cybernetics Society Conference on Advances in Cybernetic Systems*, 2006.

[33] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, semantically rich visual descriptors and their application in robotics. *International Journal of Humanoid Robotics*, 7(3):1–27, 2010.

[34] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6), 1979.

[35] F. Shevlin. Analysis of orientation problems using Plücker lines. *International Conference on Pattern Recognition, Bisbane*, 1:65–689, 1998.

[36] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[37] K. Simonsen, M. Nielsen, F. Pilz, N. Krüger, and N. Pugeault. Spatial-temporal junction extraction and semantic interpretation. *5.th International Symposium on Visual Computing. Lecture Notes for Computer Science (LNCS), Springer Verlag 2009*, 2009.

[38] H. Tao, H.S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 24(1):75–89, 2002.

[39] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H.F. Durrant-Whyte. Simultaneous Localization and Mapping with Sparse Extended

Information Filters. *International Journal of Robotics Research*, 23(7–8):693–716, 2004.

[40] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[41] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, Nov 2001.

[42] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[43] A. Yang and F. Freudenstein. Application of dual–number quaternion algebra to the analysis of spatial mechanisms. *ASME Journal of Applied Mechanics*, pages 300–308, 1964.

[44] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 30(7):1198–1211, 2008.