# Accumulation of different visual feature descriptors in a coherent framework

Jeppe Barsøe Jessen[1], Florian Pilz[2], Dirk Kraft[1], Nicolas Pugeault[3], and Norbert Krüger[1]

[1] Mærsk Mc-Kinney Møller Institute,
University of Southern Denmark, Odense, Denmark
http://www.mmmi.sdu.dk/covig/
[2] Department of Architecture, Design & Media Technology,
Aalborg University, Denmark
[3] Centre for Vision, Speech and Signal Processing,
University of Surrey, United Kingdom

**Abstract.** We present a temporal accumulation scheme which disambiguates different kinds of visual 3D descriptors within one coherent framework. The accumulation consists of a twofold process: First, by means of a Bayesian filtering outliers become eliminated and second, the precision of the extracted information becomes enhanced by means of an unscented Kalman filtering process. It is a particular property of our algorithm to be able to deal with different kinds of visual descriptors by the very same mechanism. We show quantitative and qualitative results.

## 1 Introduction

This article proposes a novel on-line method for learning representations of objects' shape based on probabilistic tracking of a family of heterogeneous local descriptors over time, in 2D and 3D. We present a unified method that allows the temporal filtering of such different visual descriptors using a common approach. This approach allows a robotic system to learn autonomously representations of objects by manipulating them. Having internal representations of objects shape is required by state-of-the-art robotic grasping and manipulation approaches, and it is often provided as prior knowledge (e.g., as CAD models). The capacity for a robotic system to learn on-line an internal representation enabling object interaction and manipulation is an important goal for cognitive robotics [9]. In this work, we describe an object using a combination of local descriptors that are accumulated over time while the robot manipulates the object. Here, we will describe objects using a combination of edges, junctions and texture patches. The object representation is based on an Early Cognitive Vision (ECV) framework that has been presented in [21].

Visual descriptions of objects and scenes can be constituted from a variety of feature types, like point features [12, 14], edge-like features ([21, 1]) or texture descriptors in terms of patchlets [15]) carrying complementary information. Some feature types can be shown to have different relevance for different tasks,

and previous work has outlined these limitations and the benefits of using a combination of descriptors to alleviate these limitations (e.g., for the case of motion estimation, [17]). In the early cognitive vision system described in [21], these different image structures are distinguished and represented by different kinds of symbolic descriptors which parameterize the content of the local patches according to the semantic content of the local patch (see figure 1). In addition to geometric properties such as position and orientation, these features also possess appearance information. It has been shown (see, e.g., [17]) that it is advantageous to make use of these different aspects of visual information depending on the task and the actual context.

When using 3D information in visual representations, we face three problems: Firstly, wrong correspondences in (stereo) matching result in outliers in the representation. Secondly, occlusion lead to incomplete representations. Thirdly, 3D information is subject to uncertainties evolving in the reconstruction process. All three problems can be reduced by merging information across different object views. For this purpose, a number of methods have been developed (SFM, SLAM, bundle adjustment). These methods have been designed mainly for point features (see, e.g., [2, 16]), although some work also exists on line features [5, 19].

In this paper, we describe an algorithm that is designed such that it can be applied to different feature types *jointly* allowing for the accumulation of rich and disambiguated scene and object representations. This flexibility is achieved through a generic three stage scheme, which 1) makes use of Bayesian filtering for outlier removal based on confidences associated to the different feature types, 2) extends the representation by novel scene or object aspects and 3) reduces the uncertainties by an Unscented Kalman Filtering (UKF) approach. A particular property of our approach is that all three stages of our scheme can deal with the different kinds of descriptors by the very same machinery.

The algorithm was introduced in [19] and includes the use of Unscented Kalman Filtering (UKF) [6] to track the distribution in the whole feature space, instead of only considering the feature's position. This includes the semantic interpretation of each individual descriptor allowing us to keep track of the relative reliability of different components of the feature vector by their altered cross modality variance. Furthermore the algorithm incorporates probabilistic matching of features based on both geometric and appearance information. Moreover it uses temporal re-evaluation of a feature's confidence according to tracking success, including a mechanism for deletion and preservation of descriptors over time. This work extends the described approach to be able to cope seamlessly with different feature descriptors. Appropriate parameterizations for the different feature types are discussed.

The accumulation of the symbolic representation is an important disambiguation mechanism of the ECV system and has been applied for object learning and recognition [9] in the context of line features. The work introduced in this paper will allow for the extension of such work to richer representations realizing even more efficient and stable pose estimation, recognition and grasping.

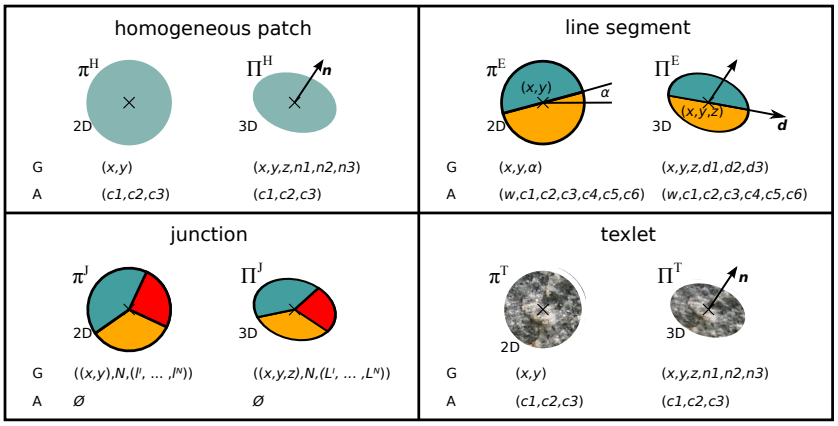



**Fig. 1.** Four different image structures in 2D and 3D and their parameterization. We distinguish between four different kinds of local image structures: *Homogeneous patches*, *edges*, *junctions* and *texture*. All four structures need to be represented by local symbolic descriptors covering appearance as well as geometric information, both in 2D and 3D. The semantic content is very different for the different kinds of structure.

## 2 Feature Descriptors

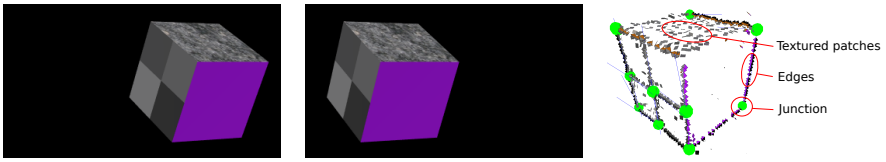
The differentiation in feature type is achieved by making use of the concept of intrinsic dimensionality of the local image signal [3]. When we talk about a specific kind of descriptor $\Pi^K$, as shown in figure 1, we indicate this by a superscript $K \in \{H, E, J, T\}$ denoting homogeneous patches, edges, junctions and texture patches respectively.

The 3D feature descriptors $\Pi^K = (G^K, A^K, \Sigma_G^K, \Sigma_A^K, B)$ illustrated in figure 1 together with their 2D equivalent $\pi^K$, are parameterized by five terms representing geometric information $G^K$, appearance information $A^K$, corresponding uncertainty estimates $\Sigma_G^K$ and $\Sigma_A^K$ and a confidence $B \in [0, 1]$. The confidence $B$ represent the systems current belief that the given descriptor is a correctly extracted primitive representing a feature in the physical scene. The first four terms depend on the feature type itself and will be defined below.

Tracking the different feature types is done using an Unscented Kalman filter. This requires a state vector representing the current state of a primitive. Thus, for each primitive type $K$ we define the state vector: $S^K = \mathbf{state}(\Pi^K) = (G^K, A^K)$ which also allows for a straightforward update of the primitive when a new state has been estimated.

The exact parameterization of the descriptors as well as the associated initial covariance matrices are defined in the following subsections. Note that we do not discuss homogeneous primitives here since our system is based on stereo processing which can not be initialized at homogeneous areas.[4]

[4] However, note that the accumulation scheme could also be used on data extracted by sensors not having this problem, such as, e.g., laser sensors.




(a) Left (b) Right (c) Stereo model

**Fig. 2.** (a) and (b) shows a stereo pair from the sequence. (c) is the extracted stereo primitives projected onto an image. Some features are labeled to illustrate the difference in visualization.

**Edge Primitives** have an orientation that can be reliably computed as well as a position on a one-dimensional manifold (aperture problem). The local structure can also be determined from local filter responses [8], allowing to differentiate between step edges (e.g., transition from dark to bright) and line structures (e.g., bright line on darker background). This structure is taken into consideration when extracting and encoding the color information [21]. An appropriate geometric representation of the edge or line segment primitive carries a full 6D pose in 3D. First we have the 3D position and second there is a vector $\mathbf{d}$ pointing in the direction of the line. The appearance information of the line segment consist of two color triplets defining the color on the left and right side of the edge (and one possibly on the edge for a line structure), and a phase $\omega$ defining the color transition. Formally, we have $G^E = (\mathbf{t}, \mathbf{r}) = (x, y, z, d_1, d_2, d_3)$, $A^E = (\omega, c_1^l, c_2^l, c_3^l, c_1^r, c_2^r, c_3^r)$. The covariance of the edge primitive is $\Sigma_{G,0}^E \in \mathbb{R}^6 \times \mathbb{R}^6$ for the geometry and $\Sigma_{A,0}^E = I_6$ for the appearance (where $I_n$ is the identity matrix of dimension $n \times n$).

**Junction Primitives** are intersections of edges and have a complex 2D geometry covering the intersection point as well as half–lines extending from it. Because of this complexity, a large degree of ambiguity can be expected in the computation of the junction parameters and appearance information is not reliable enough for matching. The complex geometry extends to the 3D domain where an important distinction is whether the lines intersecting in 2D also intersect in 3D. We represent the geometric information of the junction primitive as the 3D position where the lines intersect. A list of the intersecting lines of the junction is also maintained as a list of links $L$ to line segment primitives. These line segments are accumulated using the normal procedure for lines with the added constraint that they can only be matched with line segments that belongs to matching junctions. Color information is contained in the line segments. Formally, we have $G^J = \{(x, y, z), n, (L_1^E, ..., L_n^E)\}$. Appearance information is disregarded for junctions. The covariance for the junction primitive is $\Sigma_{G,0}^J \in \mathbb{R}^3 \times \mathbb{R}^3$ for the geometry.

**Texture Primitives** are characterized by an intrinsic complexity which is difficult to characterize in 2D [18]. This complexity however allows in general for the computation of reliable correspondences for stereo and optic flow processing. A reasonable 3D interpretation is a 3D surface patch, which in contrast to homogeneous patches, can be computed reliably by stereo matching. How-

ever, also irregular structures (e.g., trees) in 3D create 2D textures. Hence a 3D representation of the geometric information probably also requires at least two different descriptors for surface patches and irregular structures as outlined in [7] to which we also refer to for further details. The texture primitive, also denoted texlet because of the similarities with the patchlet introduced in [15], is defined by a full 6D pose in 3D. For now the only appearance information computed for the texlet is the mean color. Formally, we have $G^T = (x, y, z, n_1, n_2, n_3)$, $A^T = (c_1, c_2, c_3)$. The covariance for the texture primitive is $\Sigma^T_{G,0} \in \mathbb{R}^6 \times \mathbb{R}^6$ for the geometry and $\Sigma^T_{A,0} = I_3$ for the appearance.

## 3 The Accumulation Algorithm

In this chapter we describe a framework which makes use of the spatial representation by computing the different image descriptors at a given frame (see [21]) and (already available or estimated) rigid body motion information (in the following called 'RBM') in order to predict a representation for the next frame, compare it with the actual representation extracted in the next frame and finally merge the two representations.

Some notation must be introduced to describe the generic use of Unscented Kalman filtering of motion and Bayesian confidence update. Every primitive that has been extracted from the image is an *observed* primitive in the Kalman filtering domain and we denote this $\Pi$. An accumulated primitive $\tilde{\Pi}$ on the other hand is an abstract entity, which most likely has never been observed in its exact form in any image. It is the result of interpolation between matches over multiple frames. From the abstract primitives in the accumulated representation we compute predictions $\hat{\Pi}$ by applying an RBM. These predictions can then be matched with the extracted primitives of the next frame.

When a 3D primitive is represented by its corresponding state vector we denote this accordingly, meaning that $S$ is the extracted state, $\tilde{S}$ is the accumulated state and $\hat{S}$ is the predicted state. Uncertainties can also be denoted according to the primitive it belongs to, i.e., $\Sigma, \tilde{\Sigma}, \hat{\Sigma}$. We use the notation $\Pi^K_{i,t}$ to indicate the $i$–th primitive in a set of primitives of type $K$ belonging to the $t$–th frame. Similarly we have $S^K_{i,t}$ for the state vectors.

All primitives have an associated confidence. The confidence $B^K_{i,t}$ indicates the system's belief at time $t$ whether this descriptor corresponds to an object structure. For the newly extracted ones we use a prior confidence depending on the type of primitive, the confidence of an accumulated abstract primitive is estimated using Bayesian filtering and the predicted primitive will have a confidence identical to the originating abstract primitive.

In the following subsections the individual parts of the accumulation algorithm will be described in further detail based on the state vector representation of the primitive. The first three steps are basically Kalman filtering involving a prediction, matching and correction step. The final step is Bayesian filtering, which updates the confidences according to primitive state and matching history.

### 3.1 Kalman filtering

The RBM can be formulated in generic terms applying for all primitives defined as state vectors. It will affect geometric information only as the appearance ideally would stay constant.

$$\hat{S}^K_{t+1} = \mathbf{state}(\hat{\Pi}^K_{t+1}) = \mathbf{state}(\mathbf{RBM}^{(T,R)}(\Pi^K_t)) \tag{1}$$

We make use of the Scaled Unscented Transform (SUT) to estimate the new covariance $\hat{\Sigma}^K_{t+1}$ of the predicted state $\hat{S}^K_{t+1}$. The SUT allows for the prediction of 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 $f(S)$ as described in [19].

The transformation of the primitive under an RBM includes a transition of the geometric information in the current state to the Special Euclidean group of dimension 3, $SE(3)$. In this work we use dual-quaternions when representing $SE(3)$. We will not go into further detail on the theory of dual quaternions here but instead guide the interested reader to [19]. Both the feature pose and Rigid Body Motions (RBMs) are well described by dual-quaternions, which then allows for a compact formulation of a pose transformation under a RBM (with $T, R$ being the translation and rotation parameters of the RBM):

$$\mathbf{RBM}^{(T,R)}(\tilde{\Pi}^K_t) = \hat{\Pi}^K_{t+1} \tag{2}$$

We indicate that an accumulated primitive $\tilde{\Pi}$ has been matched with an observed primitive $\Pi$ at time $t$ by $\mu_t(\tilde{\Pi}, \Pi)$, which will be defined below.

Having computed the predicted representation, the next step of the filtering is to compare this model with the observed features. A newly observed 3D–primitive $\Pi_j$ is matched with a predicted 3D–primitive $\hat{\Pi}_i$ if their associated states are matched according to a $\chi^2$ criterion applied to their Mahalanobis distance.

$$(\hat{S}^K_{i,t+1} - S^K_{j,t+1})^\top (\hat{\Sigma}^K_{i,t+1} + \Sigma^K_{j,t+1})^{-1} (\hat{S}^K_{i,t+1} - S^K_{j,t+1}) < \chi^2_{k=N^K, p=0.05} \tag{3}$$

In this equation $\chi^2_{k=N^K, p=0.05}$ indicates the $p = .95$ value in the $\chi^2$ distribution of dimension $N^K$. By definition of the Mahalanobis distance, this implies that 95% of the correct matches will satisfy this criterion. In this case, likelihood of the match $\mu_t$ in each projected frame is evaluated using a normal distribution centered on the predicted primitive. By that we define the binary match function $\mu_t(\tilde{\Pi}_i)$ which is 1 when an abstract primitive was matched at time $t$ or 0 elsewise.

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. (4)) is preserved in a winner–take–all fashion.

$$p\left[\mu_t(\hat{\Pi}_i, \Pi_j)\right] = \frac{\exp\left[-\frac{1}{2}(\hat{S}_i - S_j)^\top \hat{\Sigma}_t^{-1}(\hat{S}_i - S_j)\right]}{(2\pi)^{n/2}\sqrt{|\hat{\Sigma}_t|}} \tag{4}$$

If the $\chi^2$ criterion is not met, we define that $p\left[\mu_t(\hat{\Pi}_i, \Pi_j)\right] = 0$. Once the matching is done, the set of model features $\hat{\Pi}_t$ can be corrected from the newly observed features $\Pi_t$ using a straightforward Kalman filtering approach as outlined in [19] for line features.

### 3.2 Accumulation of confidence

We define the tracking history of an abstract primitive $\tilde{\Pi}_i$ from its emergence at time 0 until time $t$ as:

$$\boldsymbol{\mu}_t(\tilde{\Pi}_i) = \left(\mu_t(\tilde{\Pi}_i), \mu_{t-1}(\tilde{\Pi}_i), \cdots, \mu_0(\tilde{\Pi}_i)\right)^T \tag{5}$$

thus, applying Bayes formula:

$$p\left[\Pi_i|\boldsymbol{\mu}_t(\tilde{\Pi}_i)\right] = \frac{p\left[\boldsymbol{\mu}_t(\tilde{\Pi}_i)|\Pi_i\right] p\left[\Pi_i\right]}{p\left[\boldsymbol{\mu}_t(\tilde{\Pi}_i)|\Pi_i\right] p\left[\Pi_i\right] + p\left[\boldsymbol{\mu}_t(\tilde{\Pi}_i)|\neg\Pi_i\right] p\left[\neg\Pi_i\right]} \tag{6}$$

where $p\left[\Pi\right]$ is the prior likelihood that a primitive of a specific type has been correctly extracted and $p\left[\neg\Pi\right]$ is the prior likelihood that it has been erroneously extracted. $p\left[\boldsymbol{\mu}_t(\tilde{\Pi}_i)|\Pi\right]$ is the likelihood of a primitive tracking history $\boldsymbol{\mu}(\tilde{\Pi}_i)$ given that the primitive $\Pi$ is correctly extracted.

According to [19], if we rewrite Eq. 6 and assume independence between successive observations we have:[5]

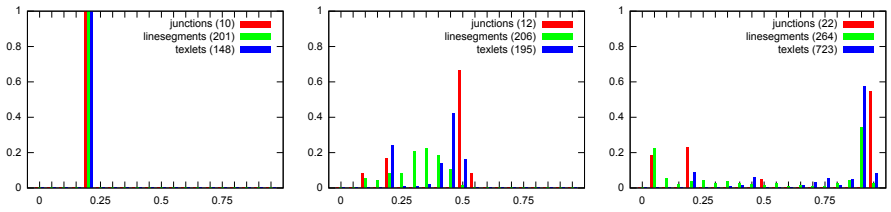
$$p\left[\Pi_i|\boldsymbol{\mu}_t(\tilde{\Pi}_i)\right] = \left(1 + \frac{\prod_t p\left[\mu_t(\tilde{\Pi}_i)|\neg\Pi_i\right] p\left[\neg\Pi_i\right]}{\prod_t p\left[\mu_t(\tilde{\Pi}_i)|\Pi_i\right] p\left[\Pi_i\right]}\right)^{-1} \tag{7}$$

The computed likelihood is used as feature confidence $B$. This allows both for elimination of entities with confidence below a minimum threshold and to freeze entities with confidence above an acceptance threshold. Eliminated features are removed from the representation as a result of poor matching or matching quality. Frozen features have their confidence locked, but are still updated with the Kalman filter when matching is possible.

## 4 Results

To evaluate the accumulation framework we apply the system in two different scenarios. First, an artificial image sequence is generated using a simple cube rendered in OpenGL for perfectly known motion, shape and pose. This is ideal

[5] Note that a particular issue of this formulation is the requirement to record the entire matching history $\boldsymbol{\mu}_t(\tilde{\Pi}_i)$. Therefore we use a recursive formulation derived from equation (7) introduced in [19], which is more practical for an on-line algorithm.




(a) frame 0 (b) frame 1 (c) frame 10

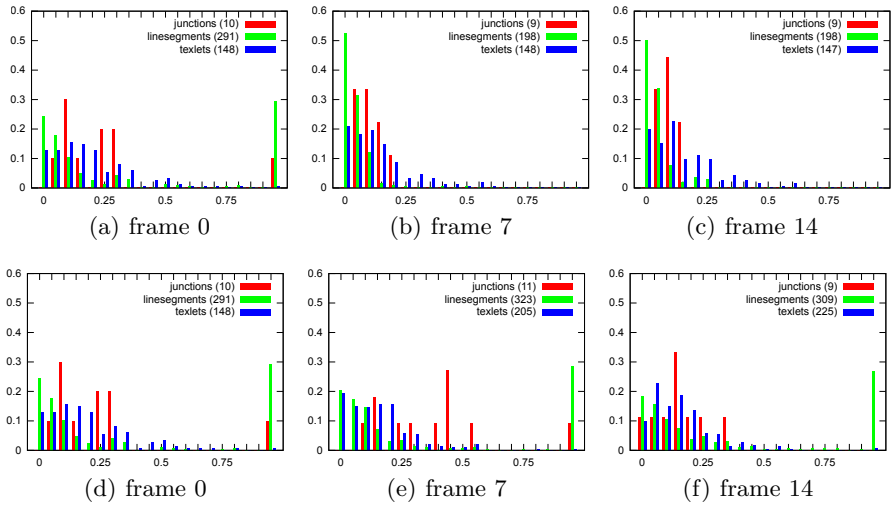
**Fig. 3.** Confidence histograms including eliminated features.

for quantitative verification of pose correction and Bayesian confidence update based on matching quality and history. Second, we grasp a series of objects using a robot and record natural scene images while the objects are subject to motion. This is used for qualitative evaluation of the ability to build object models.

**Artificial sequence:** Figure 2 shows an example of a stereo pair of images. We use a simple object with the shape of a cube. Our cube has 6 faces of uniform colors, one face of marble texture and one face with a simple pattern of four colors. Inside the faces of uniform colors we do not expect to extract (homogeneous) features since stereo cannot generate correspondences at such structures. At the edges of the cube we expect to extract lines in two categories. Either the faces of the cube are visible on both sides of the edge and the extracted edge feature will contain colors of the object only, or the edge represents a depth discontinuity with the color of the face of the cube on one side and the background color on the other side. The face with a simple pattern of four colors, on the other hand, provides unambiguous edges, with stable appearance. At the corners of the cube the edges meet and we expect to extract junctions. At the textured face we primarily expect textured patches, but could also encounter areas that will be classified as edges or junctions.

In figure 3 we show the distribution of confidences in some early iterations. which after some iterations lead to the characteristic three–modal distribution seen in 3(c). The leftmost peak represents the eliminated features which we will be removed from the set and no longer be updated; they correspond in general to wrong stereo matches. The second peak represents newly observed features that could not be matched with existing ones and thus are added to the representation with a confidence that equals the prior probability of a correctly observed feature. The rightmost peak represents the permanently accepted features. It is an important observation that the majority of the primitives are distributed by confidence into one of the three groups. This indicates that after a few frames most primitives are either discarded or accepted as a result of the matching quality.

A ground truth model of the cube geometry is necessary to compute the distribution of errors in position and orientation for the set of features. It is straightforward to obtain for the outer geometry such as the edges, corners and faces. The internal structure of the texture is more complicated in terms of ground truth, as it contains edges and junctions depending on scale. To avoid




(a) frame 0 (b) frame 7 (c) frame 14

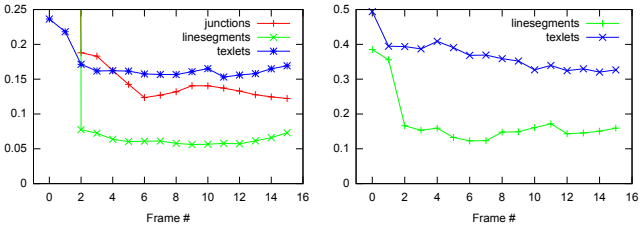
(d) frame 0 (e) frame 7 (f) frame 14

**Fig. 4.** (a)-(c): Position error histogram of corrected features, where no new hypotheses are added after first frame. (d)-(f): Position error of features extracted at corresponding single frames.

complications, we use a special extraction procedure when we want to compute the error distribution. First we extract edges and junctions on a cube without texture. Then we add the texture and extract textured patches. In this way we avoid the need for hand labeling features extracted within the textured surface. For each extracted feature we compute the shortest distance in 3D to a corresponding geometric element in the ground truth model, which is then the error in position. Having found the nearest element in the ground truth model we can also compute the error in orientation.

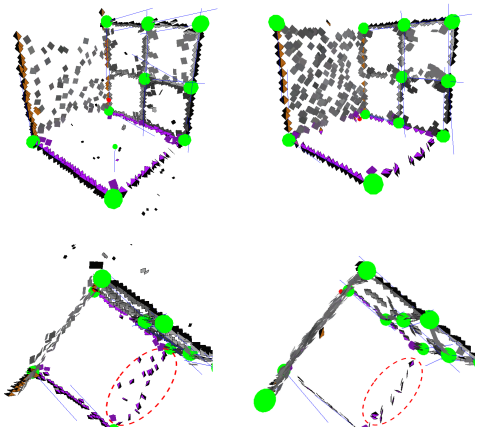
In figure 4 we show the development of position error on an accumulated model. The features are extracted at the first frame and will afterwards only be updated and not extended with new hypotheses during accumulation in order to compare the corrected representation to the stereo representation extracted at the single frames. In figures 4(a) to 4(c) we show the distribution of position errors for the accumulated representation at different stages. The single frame stereo representation extracted at the actual frame are shown in figures 4(d) to 4(f) for comparison. Features with an error out of range on the x-axis are gathered in the rightmost bar of the histogram. These high-error features disappear during accumulation when outliers are removed and the total number of features decreases. In general the small corrections lead to a shift towards smaller errors for the entire representation. Note that figures 4(a) and 4(d) are identical, as the accumulated representation after the first frame is exactly the extracted stereo.

Figure 5 shows the mean error for each of the feature types in each frame. We notice how it is reduced by outlier removal and pose correction. Also here, no new features are added after the first frame, which makes it easy to see this development.




(a) Mean position error (b) Mean orientation error

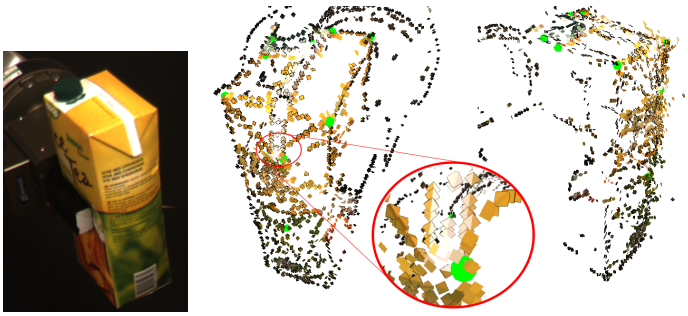
**Fig. 5.** Correction of features added in first frame. Junctions have no orientation in this context.



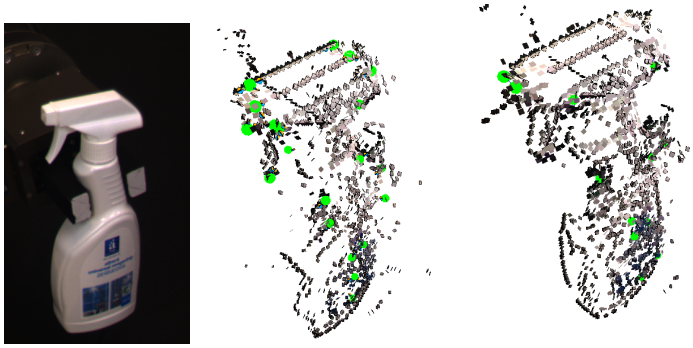
(a) 1 iteration (b) 15 iterations

**Fig. 6.** Cube represented by accumulated set of heterogeneous 3D features.

In figure 6 we compare the accumulated heterogeneous feature model of the cube from two viewpoints at different stages in the process. The green blobs indicate junctions, single colored squares represents textured patches and dual colored squares represents line segments. Multiple viewpoints are chosen to give an idea of the 3D information. Figure 6(a) shows the model after one iteration and here we see a number of outliers and inaccurate reconstructions caused by the noise we apply to the artificial image and wrong stereo matches. After fifteen iterations we observe two significant changes to the model, which is shown in figure 6(b). First, the outliers has been removed or corrected in position, as seen e.g. in the 'difficult' lines marked with a red ellipse at the bottom. This side of the cube are all the time close to horizontal in the image sequence and hence the reconstructed primitives are very noisy. Note also how the noisy line appear very nice from one viewpoint (in the top) but a lot worse from another (in the bottom). Second, the model is now more complete, seen e.g. in the number of textured patches on the textured face of the cube, the good descriptions of correctly positioned lines and the occurrence of two additional junctions.



(a) Ice tea (b) Multiple viewpoints of accumulated model



(c) Spray (d) Single stereo model (e) Accumulated model

**Fig. 7.** Real scene objects rotated by robot. The object models are accumulated over 10 frames.

**Real world scene:** Figure 7 shows objects in a real scene and their accumulated model after 10 frames. Rotational motion is applied to the object using the robot between each frame. We use motion estimation [17] to capture the motion information required for prediction. We see that the two objects are described by line segments at the contours. Junctions are extracted where the lines meet and the surface in between contour lines is nicely represented by texlets and also some line segments. The robot gripper is shown as a part of the object model as it of course shares the same motion as the object. The gripper can easily be removed because of known geometry. Figure 7(b) shows the accumulated model of an ice tea from two different viewpoints. The surface facing the camera is represented by numerous texlets and line segments and only few outliers exist. Figure 7(d) and 7(e) compares a single stereo model with the confidence thresholded accumulated model. As expected, we see that the accumulated model is indeed more complete and has fewer outliers.

**Conclusion:** We have introduced an accumulation framework that is able to disambiguate object representations consisting of different visual descriptors in a coherent way using the same machinery for all descriptors. By this we have extended the work in [19] on line segments to generic visual descriptors which

will provide richer and more powerful representations for a variety of tasks as being addressed in the early cognitive vision system [21].

## References

1. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6) (1986)
2. Dissanayake, P., Newman, P., Durrant-Whyte, H., Clark, S., Csorba, M.: A solution to the simultaneous localisation and mapping (SLAM) problem. IEEE Transactions in Robotics and Automation 17(3), 229–241 (2001)
3. Felsberg, M., Kalkan, S., Krüger, N.: Continuous dimensionality characterization of image structures. Image and Vision Computing 27, 628–636 (2009)
4. Guivant, J., Nebot, E.: Optimization of the Simultaneous Localization and Map–Building Algorithm for Real–Time Implementation. IEEE Transactions on Robotics and Automation 17(3), 242–257 (2001)
5. Isard, M., Blake, A.: Condensation — conditional density propagation for visual tracking. IJCV 29(1), 5–28 (1998)
6. Julier, S., Uhlmann, J., Durrant-Whyte, H.: A new approach for the nonlinear transformation of means and covariances in linear filters. IEEE Transactions on Automatic Control (1996)
7. Kalkan, S., Wörgötter, F., Krüger, N.: Statistical analysis of local 3d structure in 2d images. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1114–1121 (2006)
8. Kovesi, P.: Image features from phase congruency. Videre: Journal of Computer Vision Research 1(3), 1–26 (1999)
9. Kraft, D., Pugeault, N., Başeski, E., Popović, M., Kragic, D., Kalkan, S., Wörgötter, F., Krüger, N.: 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)
10. Krger, N., Pugeault, N., Baeski, E., Jensen, L.B.W., Kalkan, S., Kraft, D., Jessen, J.B., Pilz, F., Nielsen, A.K., Popovi, M., Asfour, T., Piater, J., Kragic, D., Wrgtter., F.: Early cognitive vision as a front-end for cognitive systems. ECCV 2010 Workshop on "Vision for Cognitive Tasks" (2010)
11. Lemaire, T., Berger, C., Jung, I.K., Lacroix, S.: Vision–Based SLAM: Stereo and Monocular Approaches. International Journal of Computer Vision 74(3), 343–364 (2007)
12. Lowe, D.G.: Robust model-based motion tracking through the integration of search and estimation. Int. J. Comput. Vision 8(2), 113–122 (1992)
13. van der Merwe, R., Doucet, A., de Freitas, N., Wan, E.: The Unscented Particle Filter. Tech. Rep. CUED/F-INFENG/TR 380, Cambridge University Engineering Department (2000)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)
15. Murray, D., Little, J.: Patchlets: representing stereo vision data with surface elements (2005)
16. Nistér, D.: Preemptive ransac for live structure and motion estimation. Machine Vision and Applications 16(5), 321–329 (2005)

17. Pilz, F., Pugeault, N., Krüger, N.: Comparison of point and line features and their combination for rigid body motion estimation. Statistical and Geometrical Approaches to Visual Motion Analysis, Springer LNCS 5604 (2009)
18. Portilla, J., Simoncelli, E.: A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision 40(1), 49–71 (2000)
19. Pugeault, N., Krüger, N.: Temporal accumulation of oriented visual features. Journal of Visual Communication and Image Representation (in press)
20. Pugeault, N., Wörgötter, F., Krüger, N.: Accumulated Visual Representation for Cognitive Vision. In Proceedings of the British Machine Vision Conference (BMVC) (2008)
21. Pugeault, N., Wörgötter, F., Krüger, N.: Visual primitives: Local, condensed, semantically rich visual descriptors and their applications in robotics. Special Issue on "Cognitive Humanoid Vision" of the International Journal of Humanoid Robotics 7(3), 379–405 (2011)
22. Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., Durrant-Whyte, H.: Simultaneous Localization and Mapping with Sparse Extended Information Filters. International Journal of Robotics Research 23(7–8), 693–716 (2004)