Motion Guided Learning of Object Models on the fly *

Johann Prankl, Michael Zillich, Markus Vincze

Automation and Control Institute
Vienna University of Technology, Austria
{*prankl,zillich,vincze*}*@acin.tuwien.ac.at*

Abstract. Motivated by psychologists' findings that infants already at the age of 4 months build a spatio-temporal representation of objects and perceive objects as a single entity because of coherent motion, we present a system which uses similar motion of interest points to guide the focus of attention for learning object models on the fly. The novelty of our system is to learn object models due to motion despite complex interactions of multiple objects. Consistently moving interest points are clustered, thus building the initial model of an object hypothesis. In the subsequent frames similar clusters confirm the evidence of an object and the model is extended by adding new interest points. In this way the system handles changes of appearances and rotating objects to previously unseen views. We represent objects in a star-shaped geometrical model of interest points using a codebook. A graph based spatio-temporal representation of multiple object hypotheses is maintained and thus the system is able to explain the scene even if objects are totally occluded. This representation is used for a consistent scene interpretation and to reason about possible object locations to compute a prior for object recognition.

Key words: Scene interpretation, Motion segmentation, Object recognition

* The work described in this article has been funded by the European Commission's Sixth and Seventh Framework Programmes under contract no. 6029427 (XPERO), no. 215821 (GRASP) and no. 215181 (CogX). The work was also supported by the Austrian Science Foundation under the grant #S9101 ("Cognitive Vision").

1 Introduction

One of the rising challenges is to endow our environment with capabilities to be sensitive and responsive to the presence of people. A necessary basic ability for such an artificial cognitive system (be it an ambient intelligent system or an autonomous robot) is to focus on the foreground and perceive objects as unity in contrast to the background. We aim to build a cognitive agent, which observes the environment and builds a spatio-temporal representation of object hypotheses. Our approach is motivated by psychologists' findings about infants that have shown that besides Gestalt principles and occlusion, motion is one of the most important cues to perceive object unity [1]. Gredebäck [2] and Spelke [3] have shown that infants already at the age of four months build a spatio-temporal representation of objects and accurately predict their reappearance after full occlusion.

Typical vision systems integrate low-level visual cues in a hierarchical fashion and extract relevant output from this bottom-up processing. Recent approaches try to establish feedback loops and combine different vision methods at various levels, but these methods also reach their limitations if dynamical scenes get crowded and objects get partly or even totally occluded. Our system fuses bottom-up visual processing with top-down reasoning to keep track of occluded objects and to learn appearances of objects that continuously change due to rotation or lighting. The system reasons about occlusion and hiding events and maintains an object hypothesis graph that is updated according to the visual input. We represent objects in a star-shaped geometrical model of interest points using a codebook. In case of a plausible object hypothesis from motion segmentation a learning event is triggered and the interest points of an existing object are updated or a new model is created, respectively.

We tested our system with a scenario where a human moves different objects, which interact several times, i.e., get occluded and reappear again. The goal is that the system learns object hypotheses because of coherent motion of interest points and keeps track of them even if they rotate to views which have never been seen before, or during periods of full occlusion.

The paper is structured as follows: After an overview of related work in the next section, the overall system is presented in Sec. 2. In Sec. 3, the object representation including the object hypothesis graph and the star-shaped codebook model are described. Then, the motion segmenter and the recognition component are described in Sec. 4 and Sec. 5. Finally, reasoning and hypotheses selection is described in Sec. 6 and results are shown in Sec. 7.

1.1 Related work

We present a system which integrates four basic functionalities, namely motion segmentation, tracking, object recognition and reasoning. For further readings about the first three we refer to the most relevant approaches described in [4], [5] and [6]. In what follows, we will review the state of the art regarding systems which include high level reasoning and occlusion handling.

There exist some occlusion reasoning systems for tracking or segmenting objects, mostly for traffic scenes or persons. The approaches in [7] and [8] use image regions for occlusion reasoning. A region may consist of one or more objects, the relative depth between objects is not considered. If occlusion happens, the system merges the affected regions into a new region. On the other hand a region is split, if the system is able to discriminate objects within this region. Elgammal and Davis [9] use a maximum likelihood estimation to estimate the best arrangement for people. To cope with the occlusion problem, Wu [10] proposes a dynamic Bayesian network with an extra hidden layer and in [11] tracking of multiple objects in dynamic scenes with long periods of occlusion is handled by detecting the visibility state of the objects. In case of occlusion, the whole assemble of objects is tracked.

Huang and Essa [12] present an approach for tracking a varying number of objects through temporally and spatially significant occlusions. The method is built on the idea of object permanence. They assume that a simple colour model is sufficient to describe each object in a video sequence, therefore they do not have to update their object models.

Bennett et al. [13] enhances tracking results of moving objects by reasoning about spatio-temporal constraints. The reasoning engine resolves error, ambiguity and occlusion to produce a most likely hypothesis, which is consistent with global spatio-temporal continuity constraints. However, the whole system does only bottom-up processing.

A way to incorporate knowledge into vision systems is to use a knowledge-based approach, e.g. [14] for an aerial image understanding system, and [15] for traffic monitoring applications. Matsuyama and Hwang [14] identified two types of knowledge in an image understanding system, that is, knowledge about objects and about analysis tools, and built a modular framework which integrates top-down and bottom-up reasoning. The system extracts various scene descriptions, and an evaluation function selects the most complex scene description from the database. The evaluation function is trivial and trusts the low-level vision output more than the reasoning output.

While reasoning in [15] is based on image regions using trajectories and velocity information, our reasoning is based on more abstract object behaviour to achieve a consistent scene interpretation even if objects are totally occluded.

2 System overview

Our system consists of four main parts (Fig. 1): the motion segmenter, the object detector, the reasoning component and the knowledge-base. The central role plays the reasoning component, which creates new object hypotheses triggered by the motion segmenter, maintains the hypothesis graph, predicts object locations to compute priors for the object detector and selects object hypotheses for a consistent interpretation of the current image. For each object of a specific image frame there exist several object hypotheses. Each hypothesis is linked to an object of the previous as well as to an object of the next frame. The reasoner

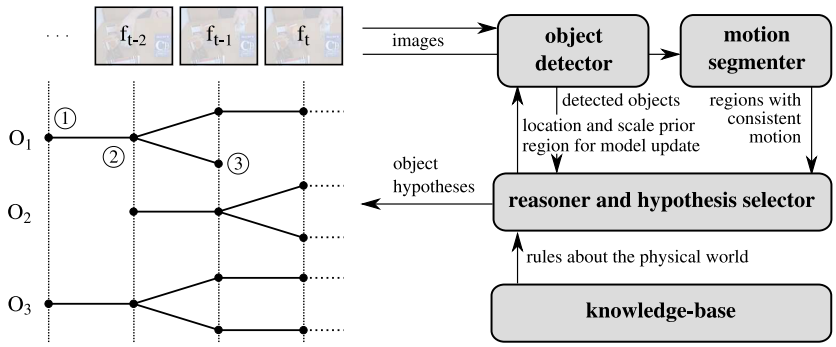


Fig. 1. System overview, including the hypothesis graph (left), the structure of the system and the communication between the different components (right).

either creates new object hypotheses of unseen motion clusters, or it creates object hypotheses including a copy of the object models of the last frames, or it creates object hypotheses with an updated model (see different nodes on the left in Fig. 1).

The result until here is an over-complete set of object hypotheses, that explains the same area of the image. To get a consistent interpretation of a particular frame a minimum description length (MDL) based selection framework is used and the best hypotheses mask weaker ones.

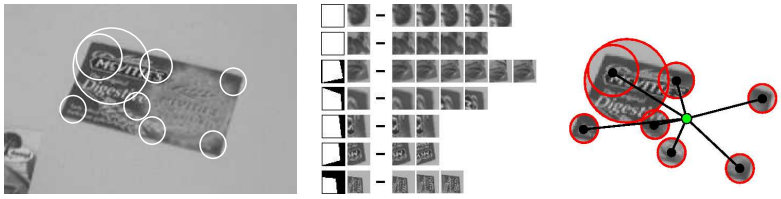
Before going into details with the different components, we describe the graph based representation of the object hypotheses.

3 Object representation

In contrast to classical object recognisers, which have an optimised model to recognise an object in one image, our approach works on image sequences and uses the history of the object hypotheses for modelling the object as well as for predicting the location in the next image. The object model is generated online and stored in the object hypothesis graph in a distributed manner.

3.1 Object hypothesis graph

The object hypothesis graph (see Fig. 1, left) is maintained by the reasoning component and stores all object locations and the models of the according views of all previous images up to the current frame. We use a star-shaped geometrical representation depicted in Fig. 2(c). Depending on the results of the object detector and the object segmenter the reasoner creates a new object hypothesis, i.e., it stores the interest points within a segmented region with respect to the centre of the region or the interest points are aligned with the stored model of the previous frame using the current detection result. Thus for each frame we have object hypotheses which are linked to the parent hypotheses of the



(a) (b) (c)

Fig. 2. Fig. 2(a) shows detected interest points and Fig. 2(b) the codebook representation, cluster means and occurrences (interest points). In Fig. 2(c) the star-shaped object representation is sketched.

previous frame and if there is a supporting segmentation for a detection result the current occurrences, i.e., the interest points within the segmented region, are stored. These occurrences are then used to generate a "small" model using the immediate previous frames optimised for tracking or – in case the object is lost – all previously seen occurrences are used to create a compact model optimised for object recognition. As proposed by Lowe [16] the Difference-of-Gaussian (DoG) operator and SIFT are used for detection and description of interest points.

3.2 Building compact object models

Our object model for recognition is inspired by the work of Leibe et al. [5], who proposed to build up a vocabulary ("codebook", see Fig. 2(a) and 2(b)) of interest points and to compose a geometric structure out of this vocabulary. We extended this approach with a geometric pruning algorithm to get a more compact model and thus speed up object detection.

The first step is to create a codebook for each object. Therefore links of current object hypotheses are traced back to the parent objects and the according descriptors of the occurrences are clustered following the RNN-algorithm as described in [5]. The RNN-algorithm is an agglomerative clustering algorithm which successively merges local descriptors until a cut-off threshold is reached. Thus this algorithm automatically determines the number of clusters while ensuring the cluster compactness. The next step is to assign the geometric locations to each cluster mean. Hence each codebook entry can vote for several object locations described in detail in Sec. 5. We use sequences of images thus a lot of similar occurrences build a codebook entry which offers the possibility for a statistical analysis to prune unreliable occurrences. In a post processing step the codebook is optimised to speed up the object detection, therefore we apply a geometric hashing for each codebook entry, in which hash bins must have at least two entries otherwise the according occurrence is deleted. Then the codebook is examined and all entries with less than two occurrences are deleted.

Summarised, separate codebooks are created for each object including occurrences of at least 3 frames for tracking and occurrences of all previous frames if an object gets lost. Clustering and geometric pruning is used to build a compact

object model including only reliable occurrences. These object models are then used for recognition described in Sec. 5.

4 Motion segmentation

The whole system is triggered by the motion segmentation component. We do not rely on a perfect segmentation of moving objects, but rather take care to achieve robustness later due to the cognitive component described in Sec. 6. Consequently, we just use a fast clustering of interest points depending on their affine motion. Our approach is inspired by the work of Pundlik et al. [4], who presented a real-time incremental approach to motion segmentation operating on sparse feature points. In contrast to Pundlik, who randomly selects interest points and uses an incremental growing algorithm, we use a 2-dimensional histogram of the length of the motion vectors and the motion direction to obtain good initial pre-clusters. Then a splitting algorithm and an outlier detection follows and theses clusters are merged depending on similar affine motion.

(a) (b) (c)

Fig. 3. Grouped motion vectors of interest points are shown with identical colours; different colours mean different clusters. Each subfigure depicts the result of different processing steps. Fig. 3(a) shows the result of grouping according to similar length and direction of the motion vectors using a 2-dimensional histogram. Fig. 3(b) is the result after examination of the neighbouring motion vectors using a delaunay triangulation and after affine outlier detection. In Fig. 3(c) the result of Fig. 3(b) is used to initialise a merging algorithm which combines clusters depending on their affine motion.

In detail: the first step is to examine the 2D-motion histogram. Therefore we search for all local maxima, that is we look for histogram bins which are surrounded by bins with a lower number of entries. Starting from the local maxima all neighbouring bins are clustered until a saddle bin is found. The result can be seen in Fig. 3(a). The next step is to split large clusters in case of intersecting convex hulls of other clusters. Therefore we use a delaunay tree to create a location neighbourhood graph of all interest points detected in the image. The splitting criterion prohibits intersections of two clusters and thus

substitutes a cluster with two new ones if they have no connection within the delaunay tree. After an affine outlier detection using a Least Median of Squares implementation, publicly available at FORTH [17] (see Fig. 3(b)), the clusters are again merged if the affine error is lower than the maximal error would be if they stayed separated. Thus the merging criterion results in

$$C_m = C_i \cup C_j \text{ for } e_m < \max(e_i, e_j) \tag{1}$$

In (1), C_i and C_j are two clusters, which are tested for similar affine motion and C_m denotes the merged cluster. e stands for the affine errors of the clusters. Additionally we can adjust a chaining parameter to cluster only features which are tracked for more than two frames and are thus considered as more stable.

Fig. 3 shows the results of all three main steps. It can be seen that the three outliers on the hand are filtered as well as the mismatch on the keypad of the telephone. The final motion clusters are handed over to the reasoning component which initialises a new object hypothesis or adds the features to an existing object. This is explained in detail in Sec. 6, after the object recogniser is described.

5 Object detection

The same interest points (DoG-operator and SIFT-descriptor [16]) used for the motion segmentation just described, are also used for object recognition. The detected interest points are matched with the codebook and activated codebook entries vote for an object centre.

Consistent votes are accumulated in the Hough accumulator array. We use a three dimensional space where occurrences of activated codebook entries vote for an object location $\mathbf{x}_v = (x_v, y_v)$ and a scale s_v:

$$s_v = \frac{s_i}{s_{occ}}, \tag{2}$$

$$\mathbf{x}_v = \mathbf{R}\mathbf{x}_{occ}s_v + \mathbf{x}_i. \tag{3}$$

In (2), s_i is the scale of the detected interest point in the current image and s_{occ} denotes the scale of the occurrence in the learning image, respectively. In (3) $\mathbf{x}_i$ is the location of the detected interest point in the current image, $\mathbf{x}_{occ}$ denotes the location of the object centre with respect to an occurrence of the model and $\mathbf{R}$ stands for the matrix that describes the rotation from model to image orientation of the interest point.

Once all matched interest points have voted, the Hough accumulator array is used to find the most promising object hypotheses. The probabilistic votes in each Hough bin i are summed up and – starting with the best hypothesis, i.e., the largest bin – the object location is refined. This is done in a mean shift like procedure, for which the neighbouring bins are examined for contributing votes. This handles the typical boundary effect of Hough voting schemas.

The result of the mean shift refinement is a cluster of interest points, that consistently vote for an object location. This cluster is used to compute an affine

homography H_{aff}, for which we use the Least Median of Squares implementation already mentioned in Sec. 4. H_{aff} is further used to project the model boundary to the current frame. The projected boundary is not only used for visualisation but also for interest point statistics and for computation of the confidence value

$$c(o|m, f_t) = -\kappa_1 + (1 - \kappa_2) \cdot \frac{n_{matched}}{n_{detected}} + \kappa_2 \cdot \frac{s_{matched}}{n_{detected}} \tag{4}$$

of an object o for a given frame f_t and an object model m. $n_{matched}$ are the matched interest points and $n_{detected}$ are the number of the detected interest points located within the boundary projected to the current frame. $s_{matched}$ is the sum of the weights

$$w = p_m p_{occ} = \frac{1}{n_m \cdot n_{occ}} \tag{5}$$

of all matched interest points with p_m and p_{occ} denoting the probabilities of the match and the occurrence in the model, respectively. n_{occ} is the number of occurrences of the specific object model of the activated codebook entry and n_m is the number of activated entries of the interest point. κ_1 and κ_2 are two constants which weight the different factors.

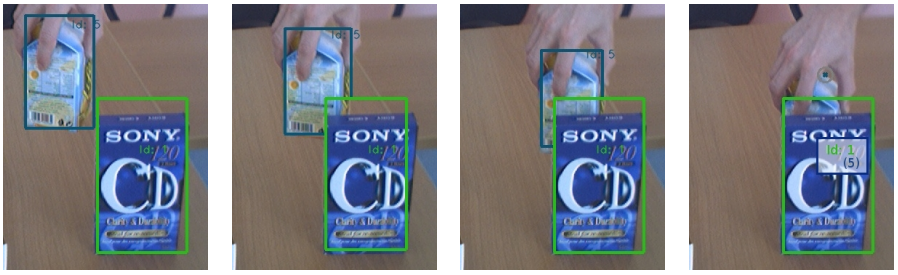
6 Reasoning and hypotheses selection

The central role plays the reasoning component, it predicts object locations both for the case of tracking and for the case of total occlusion. It creates new object models or updates existing models depending on coherent segmentation and detection results and it selects objects from an over-complete set to get a consistent scene interpretation. The following sections describe the different functionalities of the reasoner starting with the occlusion analysis.

6.1 Occlusion analysis

We aim to get a consistent interpretation of an image sequence thus it is necessary to predict objects even if they are totally occluded. Therefore we developed an event based occlusion analysis schema. If an object gets lost the past, the current and the predicted object locations in the future are examined for possible occluders. Therefore for each location the overlap of the projected object boundary with the other visible object hypotheses is computed and if they overlap the visible object gets an occlusion vote. The voting is done for all past and future object locations which are within a maximum distance of half the object size. After the visible objects accumulated the votes, the ID of the occluded object is assigned to the visible one with the most votes and to all other which got more than 80% of the maximum. It turned out that this voting schema is more reliable than only looking at the position of disappearance because in case of partial occlusion our model updating algorithm tends to shrink the estimated object boundary to the visible part of the object.

Fig. 4 shows an occlusion event, the correct depth ordering which is estimated from the confidence value (cp. Sec. 6.2), and the link of an occluded object to the occluder (indicated by an object ID within the brackets).



(a) (b) (c) (d)

Fig. 4. Occlusion event including correct depth ordering and an occluded object linked to the occluder

6.2 Confidence value for tracking using a location and a scale prior

For tracking the objects we use a constant velocity assumption therefore the affine homography H_{inc} between two frames is computed for each object. Then the assumed location and scale is computed for objects of the current frame and the confidence value is extended to

$$c_{track}(o|m, f_t) = c(o|m, f_t) + \kappa_3 \cdot \log\, p(o_{f_t}|o_{f_{t-x}}) + \kappa_4 \cdot \log\, s(o_{f_t}|o_{f_{t-x}}) \quad (6)$$

where $p(o_{f_t}|o_{f_{t-x}})$ and $s(o_{f_t}|o_{f_{t-x}})$ stand for the location and the scale prior and κ_3 and κ_4 are further constants to weight the priors. We model the priors using a Gaussian around the predicted location and the last scale. In case of occlusion the location prior is extended and surrounds the whole boundary of the occluder. Thus reappearing objects are accepted near the occluder and at the last seen location (see Fig. 5). Then the objects are sorted according to the tracking confidence value and added to the hypothesis tree.

Fig. 5. Probability map for one specific object computed using the predicted location and the result of the occlusion analysis.

6.3 Maintenance of the object models

The next step is to update existing object hypotheses. Therefore we compute an overlap matrix which describes the support of segmented regions and detected object hypotheses. We define the support

$$support_{i,j} = \frac{A_{seg} \cap A_{det}}{A_{seg} \cup A_{det}}. \tag{7}$$

where the support of a segmented region r_{seg} for a detected object hypothesis o_{det} is the ratio of the intersection and the union of the segmented area A_{seg} and the area of the detected object hypothesis A_{det}. For our experiments we used a *winner takes all* updating strategy, meaning that the detection result with the highest tracking confidence value is updated if the support is larger than a threshold t_{update}. Additionally we use a second threshold t_{new} for creating a new object hypothesis. If a segmented region does not support any detection result more than t_{new} a new hypothesis is created. Depending on the detection results and these two thresholds an over-complete set of object hypotheses is created from which hypotheses explaining the scene in a consistent way are selected.

6.4 Hypotheses selection

Our hypotheses selection framework was introduced in [18] and adapted by [5]. The idea is that the same data set cannot be occupied by more than one object and that the models cannot be fitted sequentially. Thus an over-complete set of hypotheses is generated and the best subset is chosen using a minimum description length criterion.

In our case the data set consists of the interest points and each interest point can only be assigned to one object model. Hence, overlapping models compete for interest points which is represented by the interaction costs q_{ij}. In contrast q_{ii} represents the merit term of an object hypothesis. Finding the optimal set of models leads to a Quadratic Boolean Problem (QBP)

$$\max_{n} \mathbf{n}^T Q \mathbf{n} \ , \ Q = \begin{bmatrix} q_{11} & \cdots & q_{1N} \\ \vdots & \ddots & \vdots \\ q_{N1} & \cdots & q_{NN} \end{bmatrix} \tag{8}$$

where $\mathbf{n}$= $[n_1, n_2, \cdots, n_N]$ stands for the indicator vector with $n_i = 1$ if an object hypothesis is selected and $n_i = 0$ otherwise. Q is the interaction matrix with the diagonal elements $q_{ii} = c_{track}(o|m, f_t)$ and the off-diagonal elements

$$q_{ij} = -\frac{1}{n_{o,weak}} \cdot ((1 - \kappa_2) \cdot n_{overlap} + \kappa_2 \cdot s_{overlap}) \tag{9}$$

where $n_{o,weak}$ is the number of interest points within the projected boundary to the current frame of the weaker hypothesis, i.e. with the lower confidence value, $n_{overlap}$ stands for the number of interest points which are shared by both objects and $s_{overlap}$ is the sum of the weights of all shared interest points (cp. Eq. 5).

6.5 Pruning of weak object hypotheses

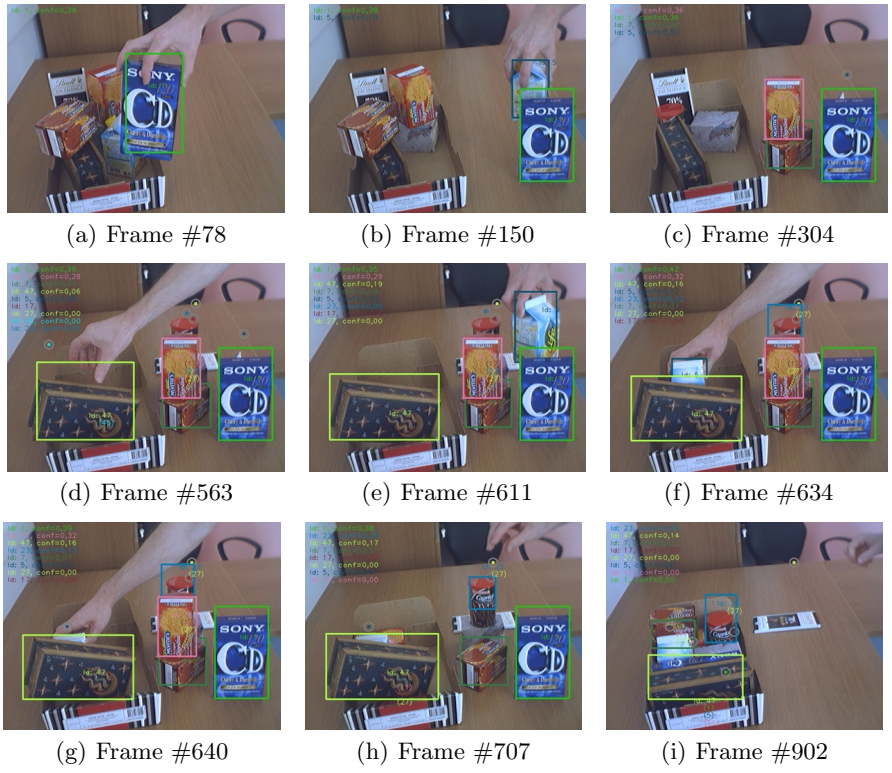
In case of a weak support of a segmentation and a detection our system generates additional object hypotheses. Continually extending the object hypothesis graph would lead to an intraceable system. Thus we introduced a *lifetime* of object hypotheses and delete models if they are not continuously updated. Motivated by the human brain, which has an exponential forgetting curve – discovered by Hermann Ebbinghaus in 1885 – we introduced an exponential lifetime

$$t_{life} = \frac{n_{seg}}{n_{life}} \cdot e^{\frac{n_{seg}}{c_{oblivion}}}. \tag{10}$$

where n_{seg} is the number of supports of a segmentation for an object, n_{life} is the number of frames since the object hypothesis was created and $c_{oblivion}$ stands for a constant to care for inaction time. Thus object hypotheses are only maintained if $t_{life} > 1$, otherwise the object model is deleted. This leads to a linear characteristics at the time when the hypothesis is created. If the object is supported by a segmentation more often it will be stored almost forever.

7 Results

We processed six video sequences to test our system. In the following, we present three sequences, which show the strengths as well as the weaknesses. The system has to detect object hypotheses because of consistent moving interest points, interpret the sequence correctly including hypotheses for totally occluded objects and build object models with all seen views. In our first video sequence, called *Sorting the Shopping Basket* we arranged typical household articles in a crowded manner in a box. Then a person empties the box, resorts the articles and places them into the box again. The sequence shows a lot of complex interactions and it is taken at a low framerate (objects move more than 40 pixels between two frames) to show that our system can handle motion blur and that it is not bounded to a strict tracking assumption, but rather selects the best interpretation which is currently available. Fig. 6 shows selected frames of the sequence. Currently available object models are depicted with bounding boxes and the according IDs and confidence values are displayed at the upper left area of each image. In Fig. 6(a) the first object is grasped and because of the motion an object hypothesis with ID 1 is generated. The next Fig. 6(b) shows the second object (ID 5) which moves behind the first object. Correct occlusion assignment and the last detected location are indicated with the ID within brackets under the occluder ID and with a coloured dot surrounded by a grey circle. During complex actions sometimes "hallucinated" object hypotheses are created (Fig. 6(d) object ID 45) which are not confirmed and thus deleted in the following frames. In Fig. 6(e) the object with ID 5 re-appears. In this frame all eight correctly learned object models are listed in the upper left area of the image. After some interactions shown in Fig. 6(f), 6(g) and 6(h) the sorted box with correct occlusion assignment is depicted in Fig. 6(i). Only the chocolate bar (ID 27) is



(a) Frame #78 (b) Frame #150 (c) Frame #304

(d) Frame #563 (e) Frame #611 (f) Frame #634

(g) Frame #640 (h) Frame #707 (i) Frame #902

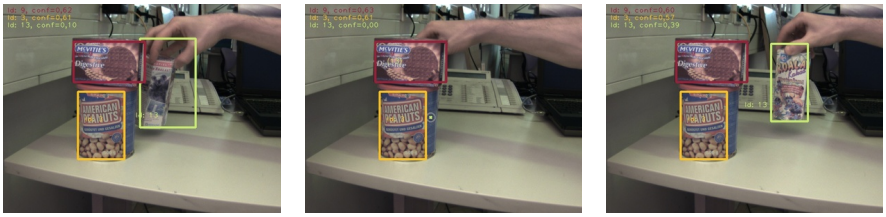
Fig. 6. Selected frames of the video named *Sorting the Shopping Basket*, indicating the complex interactions. Bounding boxes of learned objects are shown with different colours and the according IDs and confidence values of all currently available models are depicted at the upper left area of each image. If an object is lost the last position is depicted with a coloured dot surrounded by a grey circle and the ID of the occluded object is displayed under the occluder ID within brackets.

not recognised again, because of a too drastic change of the size and a too large rotation while it was occluded (i.e., no model was generated of this view before).

The second sequence depicted in Fig. 7 contains three foreground objects. One of the objects (ID 13) is rotated to different views. Then this object moves behind the other two and – triggered by the occlusion event – a model of all views, which have been shown before, is computed. During full occlusion the object is rotated and re-appears with a view shown at the beginning. It is correctly recognised again in Frame #850 (Fig. 7(f)).

In Fig. 8 another sequence with household articles is shown. Despite the correctly learned object models two errors occurred. The first one is that the model of the xerox box (ID 27) has disappeared. This object hypothesis is not confirmed often enough during tracking and thus it has been deleted due to our forgetting curve. The second error is that the occluded object with ID 5 is not linked to the occluder 37. Because of a rotation in depth during occlusion

(a) Frame #578 (b) Frame #640 (c) Frame #676



(d) Frame #772 (e) Frame #810 (f) Frame #850

Fig. 7. Part of a 900 frames long video which indicates the learning of an object model including the history of the object. The model of object 13 is learned while rotating to completely different views. Then it is moved behind object 3 and 9. During full occlusion the object is rotated and appears again with a view learned at the beginning.

(a) Frame #1820 (b) Frame #2175 (c) Frame #2460

Fig. 8. Three images of a 2590 frames long video are depicted showing two possible errors.

the prediction was wrong and thus object 5 did not get in contact with the occluder 37.

8 Conclusion

In this paper we presented a system that uses an affine model based motion clustering of interest points to create object hypotheses. If the hypotheses are confirmed in the following frames more complex object models are created. An occlusion reasoning framework is used to track objects even under full occlusion. This leads to an over-complete set of object hypotheses. We use an MDL-based model selection framework to select a consistent interpretation for each image frame. The result of our approach is a set of object models created from all previously seen frames and the assumed location for each object including completely occluded objects.

References

1. Smith, W.C., Johnson, S.P., Spelke, E.S.: Motion and edge sensitivity in perception of object unity. Cognitive Psychology **46**(1) (2003) 31 – 64
2. Gredebäck, G.: Infants Knowledge of Occluded Objects: Evidence of Early Spatiotemporal Representation. Number Dissertation, ISBN 91-554-5898-X. Acta Universitatis Upsaliensis; Faculty of Social Sciences (2004)
3. Spelke, E.S., von Hofsten, C.: Predictive reaching for occluded objects by 6-month-old infants. In: Journal of Cognition and Development. Volume 2(3)., Lawrence Erlaum Associates, Inc. (2001) 261–281
4. Pundlik, S., Birchfield, S.: Real-time motion segmentation of sparse feature points at any speed. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **38**(3) (June 2008) 731–742
5. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vision **77**(1-3) (2008) 259–289
6. Leibe, B., Schindler, K., Cornelis, N., Gool, L.V.: Coupled object detection and tracking from static cameras and moving vehicles. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10) (2008) 1683–1698
7. Brémond, F., Thonnat, M.: Tracking multiple non-rigid objects in video sequences. IEEE Transaction on Circuits and Systems for Video Technology Journal **8**(5) (September 1998)
8. McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H.: Tracking groups of people. Computer Vision and Image Understanding **80**(1) (October 2000) 42–56
9. Elgammal, A.M., Davis, L.S.: Probabilistic framework for segmenting people under occlusion. In: ICCV. (2001) 145–152
10. Wu, Y., Yu, T., Hua, G.: Tracking appearances with occlusions. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2003) 789
11. Yang, T., Li, S.Z., Pan, Q., Li, J.: Real-time multiple objects tracking with occlusion handling in dynamic scenes. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1, Washington, DC, USA, IEEE Computer Society (2005) 970–975
12. Huang, Y., Essa, I.: Tracking multiple objects through occlusion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05). Volume 2., San Diego, CA, USA (June 2005) 1051–1058
13. Bennett, B., Magee, D.R., Cohn, A.G., Hogg, D.C.: Using spatio-temporal continuity constraints to enhance visual tracking of moving objects. In: ECAI-04. (2004) 922–926
14. Matsuyama, T., Hwang, V.S.: SIGMA: A Knowledge-Based Aerial Image Understanding System. Perseus Publishing (1990)
15. Cucchiara, R., Piccardi, M., Mello, P.: Image analysis and rule-based reasoning for a traffic monitoring system. IEEE Transactions on Intelligent Transportation Systems **1**(2) (June 2000) 119–130
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
17. Lourakis, M.: homest: A c/c++ library for robust, non-linear homography estimation. [web page] `http://www.ics.forth.gr/~lourakis/homest/` (Jul. 2006) [Accessed on 20 Jul. 2006.].
18. Leonardis, A., Gupta, A., Bajcsy, R.: Segmentation of range images as the search for geometric parametric models. Int. J. Comput. Vision **14**(3) (1995) 253–277