Hierarchical Sensor Data Fusion by Probabilistic Cue Integration for Robust 3-D Object Tracking

O. Kähler, J. Denzler
Computer Vision Group
University of Passau
email: denzler@fmi.uni-passau.de
siolkaeh@immd5.informatik.uni-erlangen.de

J. Triesch
Cognitive Science Department
University of California, San Diego
email: triesch@ucsd.edu

Abstract

Sensor data fusion from multiple cameras is an important problem for machine vision systems operating in complex, natural environments. In this contribution we tackle the problem of how information from different sensors can be fused in 3-D object tracking. We embed an approach called Democratic Integration into a probabilistic framework and solve the fusion step by hierachically fusing the information of different sensors and different information sources (cues) derived from each sensor. We compare different fusion architectures and different adaptation schemes. The experiments for 3-D object tracking using three calibrated cameras show that adaptive hierarchical fusion improves the tracking robustness and accuracy compared to a flat fusion strategy.

1. Introduction

More and more applications arise in which several cameras are placed at different positions in the environment to solve a certain task. A prominent example are surveillance tasks in public areas such as airports or parking lots. Up to now it is an unsolved problem how sensor data from several sensors shall be fused considering that each sensor cannot contribute to the solution of the problem all the time the same way (for example, due to occlusions). Also even complete failures in individual sensors may occur (e.g., due to hardware problems), which should not result in a breakdown of the whole system. In addition to that it might be necessary that each sensor is adapting its processing to the environmental conditions (day/night, rain/sunshine, etc.). Data fusion from multiple sensors should be robust with respect to these problems.

The main contribution of this work is a robust cue integration and adaptation mechanism for object tracking using multiple cameras. The basis of our approach is the Democratic Integration mechanism [4]. Democratic Integration has originally been applied to fuse multiple cues arising from a single camera. We extend this approach towards hierarchically fusing cues originating from multiple calibrated cameras. Our goals are to demonstrate that cues from multiple cameras can be fused in a self-organized manner, such that the contribution of each of the cameras is dependent on the estimated reliability of that camera, and that such a system is robust with respect to unexpected failure of individual cues or entire cameras.

For estimating the 3-D position of the moving object a particle filter approach is applied, while 2-D tracking in the image plane is done by several simple cues, such as color, motion, and template matching. For the integration of the different cues and the results of each individual sensor, we compare different integration schemes: adaptive vs. non-adaptive fusion, and flat vs. hierarchical fusion. We demonstrate the robustness of our approach in extensive experiments featuring clutter, occlusions, lighting changes, objects with changing appearance, and simulated sensor failures.

2. Probabilistic Data Fusion

In the following, the data fusion concepts shall be explained in greater detail. First the original Democratic Integration approach [4] is briefly presented and reformulated as a probabilistic data fusion mechanism with adaptation as illustrated in Figure 1. This concept is then extended to a more general fusion architecture with different state spaces. Then we will present approaches for hierarchical data fusion.

2.1. Data Fusion with Democratic Integration

In Democratic Integration several adaptive cues are fused in a self-organizd fashion. A central concept is the so called

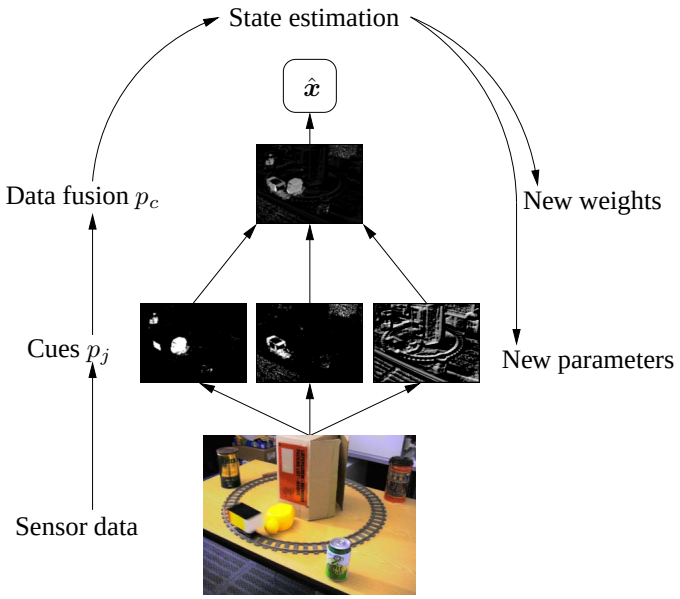

Figure 1. Overview of a general framework for probabilistic data fusion with adaptation

result saliency map $p_c^{(t)}$ into which the different cues $p_j^{(t)}$ are fused to produce the final result for tracking with one camera. The cues used in this work are motion detection, color tracking, template matching, and trajectory prediction. The motion cue computes differences in subsequent edge images to estimate a moving objects location. The color cue is adaptively locating image areas that are compatible with the estimated color of the tracked object. The template cue relies on a simple correlation based template matching, again in edge images. The prediction cue uses the results of the other cues to predict the next object position assuming constant acceleration. All of these cues are fairly simple and well known from previous works [4].

Each of these cues computes — assuming proper normalisation — a probability distribution over the 2–D state space describing the position of the moving object in the image plane. The result saliency map combines the individual cues in a weighted sum with weights $w_j(t)$:

$$p_c^{(t)}(\boldsymbol{x}) = \sum_{j=1}^{J} w_j(t) \cdot p_j^{(t)}(\boldsymbol{x})\,. \qquad (1)$$

The final state estimation can then be performed on the fused probability distribution $p_c^{(t)}$ using maximum likelihood. This results in an estimated state $\hat{\boldsymbol{x}}$, the estimated position of the tracked object.

In an adaptation step the internal parameters of the cues, e.g. the tracked color or template, can be updated by feeding back this global result to the individual cues. This way the cues stay co-ordinated ensuring that they are all tracking the same object, and it is possible for them to react to changes in the environment or the object appearance.

In addition, the time dependent weights w_j can be adapted. For this, a quality measure q_j has to be defined for the individual cues. Basically this task can be done by comparing two probability distributions, $p_j^{(t)}$ and $p_c^{(t)}$. The more similar the distribution $p_j^{(t)}$ is to the global result, the higher the quality rating of the underlying cue. Various distance measures can be used for such measurements (see below). When comparing a cue's distribution $p_j^{(t)}$ with the fused $p_c^{(t)}$, it is generally advisable to discount the cue's own contribution to $p_c^{(t)}$, i.e. it is best to compare $p_j^{(t)}$ with the fused result that would obtained *without* the cue's own contribution. This effectively avoids situations where one cue can completely dominate the fusion process. This method manages to maintain the idea of encouraging agreement between cues while avoiding self-promoted domination of single cues.

With an adaptation parameter τ the qualities $q_j(t)$ lead to a dynamic weight update given by:

$$w_j(t+1) = (1-\tau) \cdot w_j(t) + \tau \cdot q_j(t)\,. \qquad (2)$$

The data fusion method discussed so far provides the required mechanisms for adapting to changing environmental conditions, but has only been defined for a single camera. In the next section we will discuss tracking with multiple cameras.

2.2. Data Fusion with Multiple Cameras

The main idea in our approach is, that for fusing the information gathered by multiple, calibrated cameras, the saliency maps are substituted by a probability distribution over the — in this case — 3–D state space [1]. In our approach we can also deal with the general case of an n–dimensional state space and observations that are made in several m_j–dimensional state spaces.

The probability distribution on the 3–D state space is approximated using a Particle Filter and the Condensation Algorithm [2, 3]. The particles are used to represent possible 3–D postitions of the tracked object. In our case the state further consists of the 3–D velocity and acceleration of the object, providing a propagation model for the Particle Filter. The data fusion mechanisms presented before can be incorporated for generating a rating of the particles. These are obtained by fusing ratings of the individual cameras. At this point projections from the 3–D state space into the 2–D state spaces have to be known, i.e. a camera calibration step providing the projections from world coordinates into image

coordinates is necessary. With these parameters it is possible to project the hypothesized object positions, i.e. the particle states, into the image planes, where probability distributions are known.

In general to fuse the probability distributions $p_j^{(t)}$ on the m_j–dimensional state spaces into one $p_c^{(t)}$ on the n–dimensional state space and using projections $\boldsymbol{\pi}_j$, the ratings $P_i^{(t)}$ for a state $\boldsymbol{x}_i$ are computed as:

$$P_i^{(t)}(\boldsymbol{x}_i) = \sum_{j=1}^{J} w_j(t) \cdot p_j^{(t)}(\boldsymbol{\pi}_j(\boldsymbol{x}_i)) \qquad (3)$$

The remaining steps of the Condensation-Algorithm are performed as usual and provide a high particle concentration at the more probable states in the overall state set. The weighted average of all particles can then be used for the global state estimation. However, using the Particle Filter we can only provide an approximation of the distribution $p_c^{(t)}$. According to general theory of Particle Filters the approximation asymptotically approaches $p_c^{(t)}$ for the particle number $I \to \infty$.

The adaptation step as presented for the 2–D case can be applied to this more general case as well, again adapting the weights w_j. We now have to compare the probability distributions that are estimated locally with the one resulting global distribution. For this quality measurement the ratings provided by one input of the data fusion and the globally fused ratings are normalized to give probability distributions on the states covered by the particle set. The qualities q_j then can be measured by comparison of two probability distributions with the same measures as before. To avoid the Despotism-Situation this is again done without comparing $p_j^{(t)}$ with its own contribution to $p_c^{(t)}$.

We can now track objects on 2–D image planes or in the 3–D space using adaptive data fusion algorithms. The fusion concept of Democratic Integration was generalized for fusing the probability distributions of arbitraty state spaces, as long as the required projections $\boldsymbol{\pi}_j$ can be defined. The next idea is to hierarchically combine both kinds of data fusion.

2.3. Hierarchies

In the given framework arbitrary fusion hierarchies could be realized. As examples we will discuss a flat fusion and a two-step hierarchical fusion.

In a flat fusion architecture the probability distributions $p_j^{(k,t)}$ of all cues j tracking the object in the image planes k are directly fused into $p_c^{(t)}$, the combined distribution of the estimated 3–D position. This approach allows each cue to be weighted individually. The computational overhead is minimized and only one fusion step is necessary. The flat fusion is the simplest possible way of using our approach for object tracking with multiple cameras.

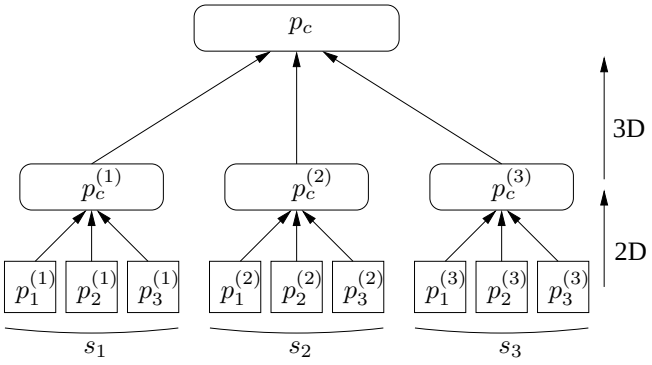

Figure 2. Example of a two level hierarchy for adaptive sensor data fusion. At the top level a 3D estimate of the position of the moving object is computed from local estimates computed in each sensor. For local estimation each sensor uses its own set of cues.

For the other hierarchy the distributions $p_j^{(k,t)}$ are first fused locally within each sensor resulting in several locally combined $p_c^{(k,t)}$. In a next step these local distributions are fused to determine $p_c^{(t)}$ at the top level, as shown in Figure 2. Several fusion steps are necessary in this approach, but being local most of them can be computed in parallel. One intuitive assumption lead to the conclusion, that this hierarchy could be superior to a flat fusion. Fusion is more reliable the more accurate the fused inputs are. In the presented hierarchy we would get more reliable inputs through the first fusion step and fuse them to get a more reliable total result. Interpreted differently, misinformation could be detected earlier and would not interfere with that many other cues as in a flat hierarchy.

An experimental comparision of these two hierarchies and examples of the overall performance will follow in the next section.

3. Experiments and Results

In order to evaluate our approach we choose the following basic experimental setup: a toy train is moving on a circular path in front of three calibrated cameras [5] (SONY DFW–VL500 firewire cameras with 320×240 pixel resolution opertaing at 25Hz). Various objects are placed in the scene producing clutter, occlusions, and reflections. Furthermore, in some of the sequences a handheld lamp introduces dynamic spotlights, turning off a lamp gives global lighting changes, and covering a camera with a hand simu-

lates sensor failures. To obtain ground truth data for a numerical analysis of the estimation error, the motion along the rail track was modelled as a circular movement with constant speed. Thus it was possible to compute the 3-D estimation error in each frame.

The toy train is tracked in 3-D using our proposed hierarchical, adaptive fusion scheme, unless stated otherwise. In a first set of experiments we systematically varied the number of particles of the particle filter, the adaptation scheme for updating the weights of each sensor and cue, and the way the reliability of a sensor or cue is estimated. In a second set of experiments, we vary the hierarchal structure of the fusion architecture, as discussed before.

For each setup 5 independent runs were started, each time with different random numbers for the transition noise of the Condensation Algorithm.

In a first experiment we varied the number of particles in the Particle Filter. As expected the error rate drops with increasing particle number, while the computational complexity is getting higher. As the accuracy saturated at around 2000 particles, we decided to run all further experiments with this number.

In a second experiment, we varied the way the qualities of sensors and cues are estimated. As explained above, this measurement can always be reduced to a comparison of probability distributions. We compared measures such as the sum of absolute differences, sum of squared differences, two metrics based on the correlation coefficient, and the Kullback-Leibler-divergence. The impact of the quality measure on the final state estimation results turned out to be only marginal, however. All measures gave roughly similar tracking performance. For the subsequent experiments we used a correlation based measure.

To test the adaptation mechanisms and the potential benefits of hierarchical fusion described above, different strategies were compared. The results are shown in Table 1. On the lefthand side the results of flat fusion with and without adaptation are shown, on the right hand side the same experiments are performed with a hierarchical architecture. The numerical values in the table denote the average estimation error — the size of the tracked toy train is roughly 50 mm. In brackets the average of the standard deviations in each run is shown. Note that very high values are due to total failure in tracking, i.e. the algorithms start tracking a different object or the states covered by the Particle Filter are out of sight for some cameras.

For both types of hierarchies the adaptation step leads to a decrease of the estimation error. In the sequences `seq4` and `seq6`, featuring many occlusions, the advantage of the adaptation step is especially significant, while at the other sequences both adaptive and non-adaptive fusion peform similar. Further the significant decrease in the error rate between a flat fusion and a two step hierarchical fusion justifies the assumptions stated earlier. Altoghether the results indicate an increase in the tracking accuracy both by adaptation and by a hierarchical data fusion.

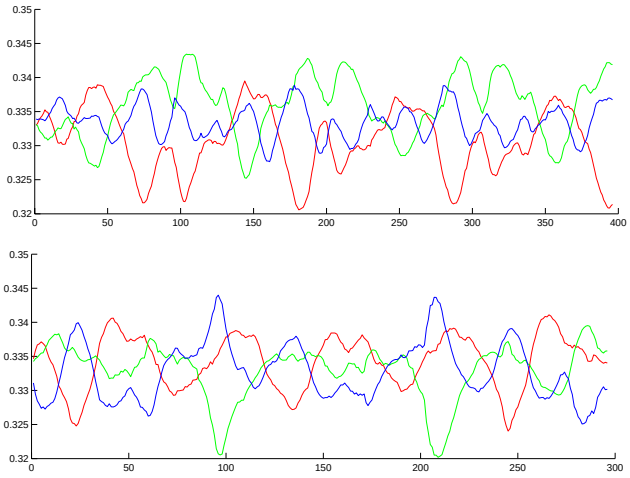

Figure 4. Fusion weights (influence) of the three cameras as a function of time (frame number): Upper: sequence `seq7_light`. Lower: `seq8`. In both cases from time to time one of the sensor's weight is reduced significantly due to the occlusion that occur by the obstacle (the coffee can).

In Figure 4 one example of the adaptive weighting of the three sensors can be seen for the sequences shown in Figure 3. It can be seen that the influence of the cameras on the final result varies over time – in these two examples due to occlusions of the object. The difference between the two examples is the more stable reliability of camera 1 (Figure 4, red curve in lower graph), which is mainly due to the better viewing angle on the scene compared to the other two cameras (see Figure 3, second row). It is worth noting, that although the change in weights and the differences between the weights of the cameras seems to be ony marginal, we get a large reduction in the 3–D estimation error in comparison with a non adaptive fusion approach, as shown before.

4. Conclusions

In this paper we presented a probabilistic extension of an adaptive sensor fusion framework, called Democratic Integration. We compared two different integration schemes, hierarchical and flat, with respect to the estimation error in 3-D object tracking. The results clearly show that a hierarchical approach outperforms the flat version. Also, the ben-

Sequence	flat		hierarchy	
	noadapt	adapt	noadapt	adapt
seq4	>1000mm (>1000)	>1000mm (>1000)	50.70mm (32.06)	65.63mm (43.26)
seq6_hand	>1000mm (>1000)	>1000mm (>1000)	234.74mm (414.18)	73.50mm (40.73)
seq6_globlight	>1000mm (>1000)	97.05mm (48.98)	94.46mm (107.83)	63.16mm (35.43)
seq6_light	>1000mm (>1000)	>1000mm (>1000)	56.76mm (34.84)	52.06mm (32.24)
seq5	>1000mm (>1000)	>1000mm (>1000)	47.01mm (21.32)	45.98mm (24.44)
seq7_hand	>1000mm (>1000)	>1000mm (>1000)	54.02mm (51.66)	45.27mm (28.63)
seq7_globlight	>1000mm (>1000)	64.89mm (30.91)	33.90mm (16.70)	38.46mm (19.92)
seq7_light	>1000mm (>1000)	>1000mm (>1000)	47.18mm (25.50)	48.31mm (27.25)
seq8	>1000mm (>1000)	229.37mm (>361.80)	53.41mm (30.82)	56.74mm (31.65)
	>1000mm (>1000)	**>1000mm (>1000)**	**74.69mm (81.66)**	**54.35mm (31.51)**

Table 1. Average 3-D estimation error and standard deviation for the different sequences.

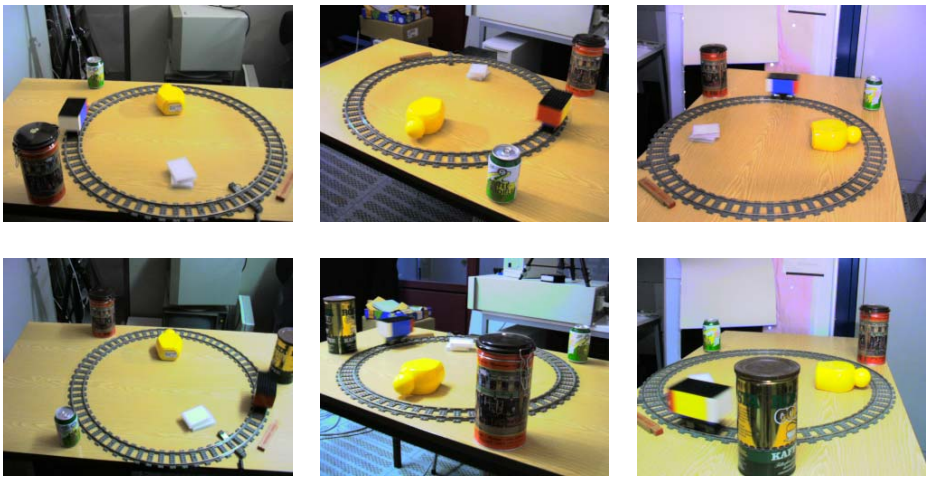

Figure 3. Sample images of images sequence `seq7_light` (top) and `seq8` (down) from the three cameras' perspective.

efits for adaptive fusion have been confirmed in the case of significant changes in the environment or failure of individual sensors. In the hierarchical fusion architecture, we could not demonstrate the need for adaptive fusion as clearly. Although the mean estimation error is reduced by 27%, the gain of adaptation depends on the scene under investigation. Our guess is, that the speed of adaptation could be adapted to the scene and the kinds of changes in the scene, which was not done here. To automatically set the speed of adaptation is one of our future research goals.

Acknowledgment

The work was partially supported by the Bavaria California Technology Center under grant 2410-2001, by the German Academic Exchange Service (DAAD 315/ab) and the National Science Foundation (NSF INT-0233200).

References

[1] J. Denzler, M. Zobel, and J. Triesch. Probabilistic integration of cues from multiple cameras. In *Dynamic Perception*, pages 309–314, Berlin, 2002. Akademische Verlagsgesellschaft Aka GmbH.

[2] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, Berlin, 2001.

[3] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[4] J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001.

[5] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, Ra-3(3):323–344, August 1987.