Probabilistic Integration of Cues From Multiple Cameras

J. Denzler[1], M. Zobel[1] and J. Triesch[2]

[1] Lehrstuhl für Mustererkennung
Universität Erlangen–Nürnberg
email: {denzler,zobel}@informatik.uni-erlangen.de
[2] Cognitive Science Department
University of California, San Diego
email: triesch@CogSci.ucsd.edu

Abstract Cue integration from multiple cameras is an important aspect for machine vision systems operating in complex, natural environments. One successful approach for self–organized cue integration is Democratic Integration. The hallmark of Democratic Integration is that different cues can autonomously determine whether and in how far they are useful for the current task, giving the system flexibilty to engage in different tasks and robustness in the face of sudden failures of cues. In this paper we embed Democratic Integration in a probabilistic framework and extend it hierachically in order to model *adaptive* cue integration for the general case of n calibrated cameras. Our experiments show that the method is capable of robust cue integration and adaptation during object tracking using three cameras placed arbitrarily in the scene.

1 Introduction

It is an unsolved problem in computer vision how sensor data selection and fusion should be done in the case that multiple cameras and multiple cues from each of the cameras are available. Such problems arise for example in surveillance tasks, where different sensors (e.g. infrared and daylight cameras) are placed at different positions in the environment and information from these sensors needs to be combined dependent on the environmental conditions (day/night, rain/sunshine, etc.). Also, the estimated position of the tracked object in the scene will have an influence on the contribution, each sensor can make. Of particular importance for real world applications in this respect is also, that individual sensors or cues may sometimes (unexpectedly) fail due to, e.g., limited view, occlusions, or hardware problems, or other reasons, and that the system must be robust with respect to such disturbances.

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 [3]. It is briefly summarized in the next section. 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.

2 Democratic Integration

The idea behind Democratic Integration is to integrate different perceptual cues in a self-organized manner [3]. Adaptation of the cues is driven by the agreement or compatibility between the different cues and sensors in the system. This idea was first studied in a face tracking system [3]. The system employed a stationary camera monitoring a room. Five simple cues analyzed the camera images. Each cue computes a 2-dim. *saliency map* registered to the camera image, in which high values inidicate a high confidence of the cue that there is a face at that location. The different cues are integrated or fused by computing a *result saliency map* which is a weighted average of the individual saliency maps. Importantly, the weights are time dependent and are constantly adpated in a self-organized fashion. To this end, an agreement or quality function is defined, that compares a cue's saliency map to the result saliency map. A cue whose saliency map is very similar to the result saliency map currently has a high quality. The important step now is to change the cue weights based on these qualities. A cue whose quality becomes very small, indicating disagreement of its saliency map to the result saliency map, will reduce its weight to no longer disrupt the overall system. Conversely, a cue that has recently been in very good agreement with the result will increase its weight. In addition, each cue can adapt internal parameters in order to better match its saliency map to the result saliency map. This allows the system to recalibrate cues and to use cues for a particular task that have no a priori information about the task. These cues are bootstrapped by other cues and simply adjust their internal parameters to match the result.

3 Probabilistic Fusion with Multiple Cameras

In Democratic Integration one of the key concepts is the result saliency map into which all different cues are fused to produce the final result for tracking with one camera. The main idea in our approach is, that for fusing the information gathered by multiple, calibrated cameras, the local and result saliency map is substituted by a probability distribution over a state space. Note, that it is quite intuitive to interpret the saliency map in 2–D — assuming proper normalization — as a distribution over a 2–D state space. In this special case the 2–D state consists of the position of the moving object on the image plane. In our approach we deal with the general case of an n–dimensional state space and observations that are made in several 2–D image planes.

The key idea of the hierarchical probabilistic approach can be summarized in the following informal way:

Probabilistic modeling of the state A particle filter framework is used to estimate the state of the object in 3–D (in the experiments the position, velocity, and acceleration of a moving object). This gives us a distribution over the state space represented by a particle set. A similar approach in the case of cue integration for a single camera has been proposed in [2].

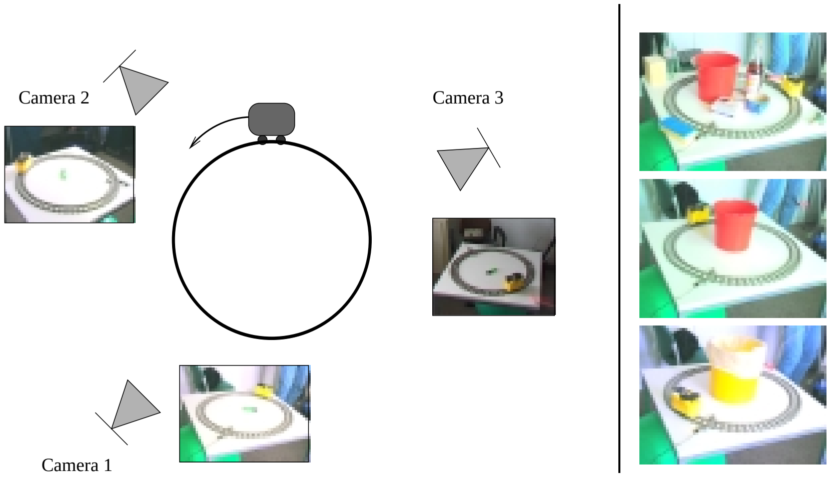

Figure 1. Left: Experimental setup. Position of the three cameras and the rail track. The images show the view on setup `basic`. Right: Images of setup `complex`, `bucket` and `yellow_bucket` (taken from camera 1).

Local state estimation For each sensor local state estimation is done using the original cue integration mechanism of Democratic Integration, i.e. a result saliency map is generated for each sensor from the different cues. This saliency map is used as likelihood function for evaluating the likelihood of each particle, that is drawn while applying the particle filter. In the case of calibrated cameras each particle, which might be interpreted as a kind of hypothesis for the 3–D state, is projected into the image plane and a score can be computed for each hypothesis by the likelihood function (for a detailed introduction on how particle filters are used the reader is referred to [1]). The weights of the different local cues as well as the other parameters of the cues are adjusted as described in [3] afterwards.

Global state estimation In an additional step a global state estimate is computed in a similar manner as it is done for each of the local state estimates. Each particle is projected onto the image planes of the different cameras. The global score of a particle is now computed as a weighted average of the local scores (already computed during the local state estimation). The weights, assigned to each camera, are updated in an additional Democratic Integration step. The main difference is, that now distributions represented as particle sets have to be compared, to figure out the agreement of the local estimates with the global ones. For comparison different metrics can be used to measure correspondence (agreement) between two distributions. One example is the Kulback–Leibler distance.

4 Experimental Setup and Results

During the experiments a moving toy train is tracked in 3-D using our proposed framework. 3-D estimation is conducted with a particle filter. The state (i.e. each

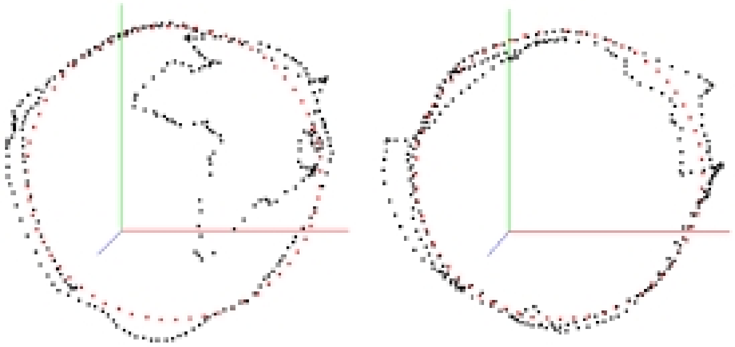

Figure 2. Estimated versus true motion path for setup `complex_occl`. Left: without sensor weight update. Right: with sensor weight update.

particle) consists of the 3-D position, velocity and acceleration of the object. For all experiments 2000 particle have been used.

In order to analyze our approach we choose the for the following basic experimental setup: the toy train is moving on a circular path in front of three cameras. Camera 1 and Camera 2 are SONY DFW–VL500 firewire cameras with a resolution of 320 × 240 at 25Hz. Camera 3 is a SONY digital camera with a resolution of 720 × 576 at 30Hz. The positions of the rail track and the three cameras are indicated in Figure 1. This setup is called `basic` in the following. In the beginning the cameras have been calibrated using Tsai's method [4].

Three different scenes are built up modifying the basic setup: a scene `complex` that contains a lot of different objects inside and outside the rail track to induce occlusions for one or the other camera and heterogeneous background. The scene `bucket` consists of a big red bucket in the center of the circular track, while in scene `yellow_bucket` a yellow bucket that has similar color as the moving toy train is used. Two more setups are constructed: `basic_occl` and `complex_occl`. In both cases the setups `basic` and `complex` are used, except for a sensor failure that was simulated by totally covering one of the cameras for a couple of seconds.

For each of the six setups a 10s sequence has been recorded for each of the three cameras simultaneously. The cameras have been manually synchronized only once at the beginning of the recording and in the end to subsample the 30Hz sequence of the third camera to match the 25Hz sequences of the first two cameras. The resolution of the images has been reduced to 80 × 60 for the first two cameras and to 75 × 60 for the third one. Additionally, the RGB images have been transformed to HSV color space.

To evaluate the quality of tracking for the different setups the circular rail track was reconstruced in 3–D using the calibration information of the cameras. As quality measure the mean euclidian distance between the estimated position of the toy train during tracking and the reconstruced circle in 3–D is used.

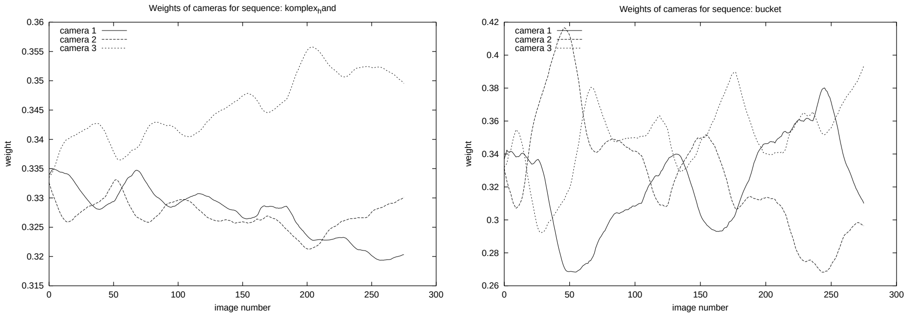

Figure 3. Cameras' weights for scenes `complex_occl` (left) and `bucket` (right)

For tracking the moving object, each camera uses the cues motion, prediction, contrast and color (for the computation and the parameters of these cues see [3]). Each experiment starts using only color and motion cue, i.e. the weights for color and motion cue are both set to 0.5. The other two cues are bootstrapped by the former ones.

In the experiments we tested different settings for the time constants τ_s (for sensor weight adaptation) and τ_c (for cue weight adaptation, see [3]). The time constants directly control how fast the influence of a sensor or a cue is changed. Since the different scenes differ in the demands on the adaptation, a compromise has been chosen between fast adaptation but not over–reacting on sensor noise or processing errors. Due to lack of space we only present results for $\tau_s = \tau_c = 10000$msec. Smaller values tend to improve the results for the sequences `complex` and `complex_occl` while at the same time the quality for `basic` and `bucket` is slighly reduced. For the setup `complex_occl` the advantage of the sensor weight adaptation can be best shown. Without sensor weight update tracking of the 3–D position breaks down during the simulated failure of sensor 1. With our proposed method (Figure 2, right) the system keeps track of the moving object with high accuracy. In Figure 3, left, the weights for cameras 1–3 are plotted over time. Evaluating the weights of the sensor over time, we can observe that the influence of each sensor is changed due to the visibility condition of the object (a periodic up and down of the weights can be observed). During failure of camera 1 the weight of this camera is decreased, as expected. A similar plot for scene `bucket` is shown in Figure 3, right, that again shows the periodic increase and decrease of the cameras' influence due to the visibility situation in the scene.

In Table 1 the estimation error is summarized for the different setups, Democratic Integration without and with sensor weight update as well as a result achieved if no cue and sensor adaptation is applied. In the latter case a non–adaptive particle filter approach is used to estimate the position in 3–D by probabilistic fusion of all three cameras.

	no weight update		weight update		no DI	
setup	mean	std. dev.	mean	std. dev.	mean	std. dev.
basic	24.6	14.4	22.3	13.0	39.1	23.7
bucket	22.7	13.3	26.6	16.6	50.7	34.2
yellow_bucket	46.7	32.5	38.8	28.3	130.4	73.4
complex	33.2	20.4	37.5	27.0	53.0	37.7
basic_occl	30.9	29.6	26.3	21.7	39.6	28.1
complex_occl	52.5	56.5	32.5	20.6	59.3	48.5
total	35.1		30.6		62.0	

Table 1. Mean euclidean error and standard deviation in the 3–D estimation of the moving toy train (in mm). Left column: without sensor weight update. Middle column: with sensor weight update. Right column: non–adaptive sensor data fusion using particle filters without adaptation of cues' or sensors' influence. The size of the toy train is approx. 110 × 80 × 90mm at a distance of 1.5-2.0m from the cameras.

5 Conclusions

In this paper we have shown first, that the integration of cues from multiple cameras can be done very elegantly in a probabilistic framework using particle filters, and second, that adaptation in Democratic Integration can not only be performed locally in each sensor but also globally giving more influence to more reliable sensors at the current situation. The circumstances in our experiments (i.e. weak synchronisation of the cameras, different types of cameras, different and low resolution of the images) prove that our approach is robust and also capable for handling systematic differences in the reliablity of the sensors, as well as unexpected temporary failure of one or the other sensor[3]. The particle filter allows for handling multi–modal distributions over the state space, i.e. dealing with multiple hypotheses and objects in the scene.

Acknowledgment

The work was partially supported by the Bavaria California Technology Center under grant 2410-2001 and the German Science Foundation (DFG) under grant SFB603 TP B2.

References

1. A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer, Berlin, 2001.
2. M. Spengler and B. Schiele. Towards robust multi–cue integration for visual tracking. In *ICVS 2001 Vancouver, Canada, 2001*, pages 93–106. Springer, 2001. Lecture Notes in Computer Science.
3. J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001.
4. 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.

[3] The reader is referred to http://www5.informatik.uni-erlangen.de/~di for image sequences and results of the processed scenes