

Fusing Dynamic Percepts and Symbols in Cognitive Systems

Michael Felsberg and Gösta Granlund

Abstract—A novel method for Bayesian tracking and generalized sensor fusion is proposed. The new approach is based on a localized histogram-like representation, the channel representation. The channel representation allows fusing continuous percepts and symbolic input, which enables the proposed scheme to implement perception-action loops with an additional control input. By connecting several such loops through action-control links, a hierarchical network with flexible structure is generated. The novel tracking loops are validated in a simple experiment with real world data.

I. INTRODUCTION

In the field of cognitive system research, a common way to model perception-action loops is the perceptual cycle of Neisser [1], see Fig. 1 (a). The perceptual cycle basically implements the perception-action loop, where the latter consists of three rather than two steps. In each cycle, some action precedes the perception step, which can be used for adaptation and incremental learning [2]. Controlled by the schemata, which can be different internal models of the world, the system performs an exploration step (action). Depending on the internal model, percepts of the object (world) are predicted and verified against sampled observations of the object. The actual percepts in relation to the predicted ones allow to modify (update) the internal model. A similar consideration has been made in [3].

The previous consideration can be formalized by relating the perceptual cycle to generalized Bayesian tracking [4], [5]. In Bayesian tracking, the current state of the system, i.e., the internal model, is represented as a probability density function of the system's state space. In the prediction step, this density is modified according to the system model, i.e., a new, typically smoother density is obtained as the prediction of the next system state. In the observation step, measurements of the system are used to update the predicted density. Typically, the density is sharpened by some type of inference and it serves as the next system state.

Relating Bayesian tracking to the perceptual cycles leads to a model for implementing low-level perception-actions loops, see Fig. 1 (b). A short introduction to Bayesian tracking is given in the main part of the paper. The reason for calling the step between the prediction and the observation *matching* will also be explained in the main part.

The main contribution of the proposed approach is to implement Bayesian tracking using channel representations [6] and linear mappings on channel representations, so-called

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 215078 and from the CENIIT project CAIRIS.

M. Felsberg and G. Granlund are with the Computer Vision Laboratory, Linköping University, S-58183 Linköping, Sweden mfe@isy.liu.se

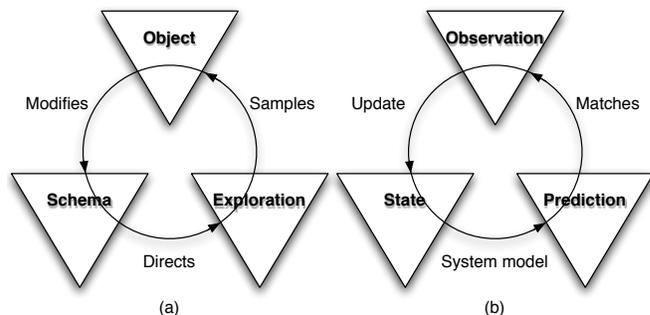


Fig. 1. (a) Neisser's perceptual cycle [1] (redrawn). (b) Bayesian tracking represented as a perceptual cycle.

associative networks [7]. The term *channel* is well established in vision literature for a representation using a band pass tuning function [8], while some may at times have a reason to view it as a population coding [9], [10]. The proposed method is closely related to grid-based methods [4] without having the drawback of quantization effects and high computational effort. Thus, the novel method is considered being superior to particle filters, since the latter also make use of quantized functions when the new samples are to be drawn, see [5], [11].

A second and even more important advantage of using channel-based Bayesian tracking is the possibility to represent symbolic states by channel representations, see [12], where channel representations and associative networks are related to fuzzy logic. As a consequence, channel-based Bayesian tracking can also be performed at higher levels of the system that incorporate symbolic information. The novel technique thus allows fusing information of different kinds and therefore goes much beyond what is treated as sensor fusion in the literature.

Based on this capability, it becomes possible to arrange channel-based tracking loops in a hierarchy, where the higher level loops activate and modify the lower-level tracking loops. The state space of the higher-level loop contains binary flags for activating lower-level loops as well as the weights for the corresponding networks. Hence, an update in the higher-level loop leads to an online learning step of the lower-level loop. Since the actual implementation of the loops itself is the same at all levels, an implicit hierarchy is built that is structured by the wiring of the loops to each other.

Besides generalizing sensor fusion to general information fusion, the novel method does not rely on a strict synchronization of states and observations during the learning step.

It has been shown in [13] that associative networks can be trained on unordered sets of output – input pairs, i.e., the point-wise correspondences need not be known. This can be exploited for learning associations between states and responses with an unknown but limited delay (weakly synchronized) by simple recursive low-pass filtering in the time domain.

II. BAYESIAN TRACKING

In this section some concepts from Bayesian tracking are introduced based on the tutorial paper [4].

The concept of Bayesian tracking is based on the definition of a process model and a measurement model, both assumed to be distorted by independent identically distributed noise \mathbf{v} and \mathbf{n}

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) . \quad (2)$$

Here, \mathbf{x}_k denotes the system state at time k and \mathbf{z}_k denotes the observation that is made at time k . Note that both models are in general non-linear and time-dependent.

The current state can be estimated, given that the previous state and all previous observations are known, by using the prediction equation. Assuming a Markov process of order one allows us to consider the conditional density of the novel state as an integral over its conditional density given the previous state

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} . \quad (3)$$

If the new measurement becomes available, the prediction is updated through the update equation

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k} . \quad (4)$$

In the literature, a number of methods to compute the respective equations are known. Probably the best known is the Kalman filter, which is however based on a number of assumptions that do not hold in the case of image-based data – in particular the linearity assumption. Even if the linearity requirement is dropped, e.g., by linearization of non-linear models as it is done in the extended Kalman filter, it is still assumed that the noise distribution is Gaussian. Also this assumption does not hold, as one often faces the problem of multi-modal densities.

Another known technique from the literature is grid-based methods. Assuming a discrete state space, the densities can be replaced with histograms without losing information. Thus, conditional probabilities of state transitions can be replaced with linear mappings and no assumptions on Gaussianity or linearity need to be made. Again, it is assumed that the previous density (histogram) is known

$$w_{k-1|k-1}^i = \Pr(\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i | \mathbf{z}_{1:k-1}) \quad (5)$$

such that the prediction and the update equations become (note the different combinations of the indices)

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \sum_i w_{k-1|k-1}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (6)$$

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \sum_i w_{k|k}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) . \quad (7)$$

The histograms involved in the previous two equations are computed as

$$w_{k-1|k-1}^i = \sum_j w_{k-1|k-1}^j p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^j) \quad (8)$$

$$w_{k|k}^i = \frac{w_{k-1|k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i)}{\sum_j w_{k-1|k-1}^j p(\mathbf{z}_k | \mathbf{x}_k^j)} . \quad (9)$$

It should be noted that the grid-based method also works for continuous state spaces, but the solution is, due to the quantization in the histograms, only approximate and suffers from a high computational load in case of high-dimensional state spaces.

The most prominent approach to circumvent the drawbacks of grid-based techniques is the particle filter. Instead of using a fixed grid for approximating the state density, random samples are drawn from the state density and are propagated according to the system model and weighted according to the observations. The sampling is done according to different procedures, e.g. condensation [5]. The choice of the sampling procedure is a well-discussed topic in the literature which reflects the main shortcoming of particle filters: Only if the sampling is done in a correct way, the filter leads to meaningful results. The main drawback here is that many sampling techniques, e.g. the numerical transformation method [11], relies on cumulative histograms and in order to achieve an unbiased sampling of the density, the number of histogram bins explodes.

III. CHANNEL-BASED BAYESIAN TRACKING

For the purpose of this section, the channel representation [6] and related coding techniques [14] can be considered as a variant of histograms where the bins are replaced with smooth, overlapping basis functions, same as in [15]. A well established case for the basis functions are \cos^2 kernels, see Fig. 2. It is known from the literature [16], that the channel representation reduces the quantization effect of ordinary histograms significantly, typically by a factor of 20. This suggests to use channel representations to avoid the

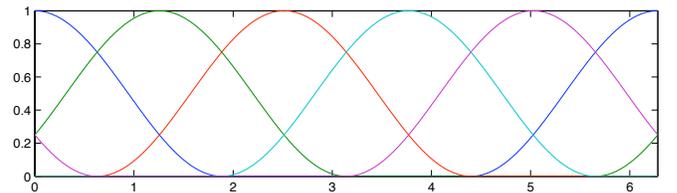


Fig. 2. Five basis functions (\cos^2) for computing the channel representation of the (periodic) orientation interval $[0, 2\pi]$.

accuracy problem with grid-based tracking methods. In order to avoid high computational costs for computing the channel representation in higher-dimensional spaces, the concept of P-channels [17] can be used, but this is of no direct relevance in this context.

When plugging channel representations into (5–9), two conditional densities have to be considered in more detail: The state transition matrix $\mathbf{A} = [p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^j)]_{ij}$ and the verification vectors $\mathbf{v}_k = [p(\mathbf{z}_k | \mathbf{x}_k^i)]_i$. Both are considered in more detail in the next section. Assuming for the moment that \mathbf{A} and \mathbf{v}_k are available, the following set of equations is obtained

$$\tilde{\mathbf{c}}_k = \mathbf{A} \mathbf{c}_{k-1} \quad (10)$$

$$\mathbf{c}_k = \frac{\mathbf{v}_k \cdot \tilde{\mathbf{c}}_k}{\mathbf{v}_k^T \tilde{\mathbf{c}}_k}, \quad (11)$$

where \mathbf{c}_k is the channel representation of \mathbf{x}_k , $\tilde{\mathbf{c}}_k$ is the prediction of \mathbf{c}_k , and \cdot is the element-wise product such that $\mathbf{v}_k^T \tilde{\mathbf{c}}_k = |\mathbf{v}_k \cdot \tilde{\mathbf{c}}|_{l1}$ (note that the $l1$ norm is equal to the sum of non-negative elements).

Up to this point, nothing is said about the nature of the channel representations, i.e., whether they encode continuous percepts or symbols. Since the channel representation can represent continuous signals and fuzzy logic events [12], the channel-based tracking equations can also be used to fuse continuous signals and symbolic information. In other words, exactly the same set of equations can be used to implement a finite state machine and to select different models for processing continuous information.

IV. LEARNING CHANNEL-BASED TRACKING

In this section, the matrix \mathbf{A} and the vector \mathbf{v} are discussed in more detail. The state transition matrix \mathbf{A} needs to be trained and updated. Basically, many different methods that preserve positivity could be applied, but good results have been obtained with correlation networks, i.e., using the covariance of subsequent states for estimating the matrix

$$\mathbf{A} = \left[\frac{p(\mathbf{x}_k^i, \mathbf{x}_{k-1}^j)}{p(\mathbf{x}_{k-1}^j)} \right]_{ij} \approx \text{mean}(\mathbf{c}_k \mathbf{c}_{k-1}^T) ./ \text{mean}(\mathbf{1} \mathbf{c}_{k-1}^T), \quad (12)$$

where $./$ denotes the point-wise division of two matrices and $\mathbf{1}$ is the column 1-vector with the same dimension as \mathbf{c} .

Since channel representations lead to density estimates smoothed by the basis function [16], applying the normalized covariance matrix as an operator onto the previous state representation will lead to two-fold smoothing of the density. Hence, and in accordance with Bayesian tracking, the basis function should represent the noise distribution [18]. The smoothing effect is however compensated in the update step which makes use of the current measurement (or percept). An important advantage of the covariance-based method for estimating \mathbf{A} is that one can easily incrementally update the transition matrix by adding the covariance of the updated new state and the previous one (and a subsequent re-normalization).

Note that in the dynamic case, the transition matrix needs to map two subsequent states onto the respective newer ones in order to cover the dynamics of the state space. In contrast to classical control theory, there is no need to specify a physical motion model, since the action at a certain position is always directed in the same direction. This is a result from motion trajectories being intrinsically one-dimensional. If a larger variety of motion models is required, these are obtained by switching between different state transitions at the next higher level.

Slightly more tricky is the estimation of the verification vector \mathbf{v} . Since the number of accessible prototypes in the memory is limited, i.e., it is assumed that a number of view representations is stored in the visual memory, only the conditional density for *particular* observations given an arbitrary state can be computed. In order to solve this issue, it is assumed that the conditional density can be replaced by a known one at the closest prototype. The latter density is obtained by simultaneously changing both, the current view representation and the state vector.

Technically, one might consider the stored prototypes as basis functions and project the current view onto this basis in order to generate another channel representation. The actual fitness could be computed using interpolation techniques on the new channel representation, but this is not required as the state vector itself is only available as a channel representation. In the simplest case, there is only one visual prototype for each state space basis function, such that the state channel vector is multiplied point-wisewith the vector containing the prototype matching scores, see (11). For the matching, any non-negative, normalized matching function can be applied.

For the experiments in Sect. VI, we encoded the image gradient orientation (double angle representation [19]) in channels and weighted the channels by the gradient magnitude. The resulting image descriptor has some similarity to the SIFT descriptor [20], which could also be used instead. If the image descriptor of the current view is denoted as \mathbf{f}_k and the image descriptors in the database (prototypes) are denoted as \mathbf{f}_p , where $p = 1, \dots, P$ and $P = \text{dim}(\mathbf{c}_k)$ equals the dimensionality of the state channel, we compute the verification vector as

$$\mathbf{v}_k = [\mathbf{f}_1 \cdots \mathbf{f}_P]^T \mathbf{f}_k. \quad (13)$$

Instead of the scalar product between the descriptors in (13), any other matching function would do. More advanced matching functions might be less sensitive to occlusion and other distortions.

If the system is supposed to learn a weakly synchronized fusion problem, i.e., there is no exact correspondence between observation and state [21], another step needs to be introduced. By associating the matching vector with some filter for the state rather than using the matching vector directly as a filter, one can exploit the effect of correspondence-free learning [13]. However, this is not further considered in here.

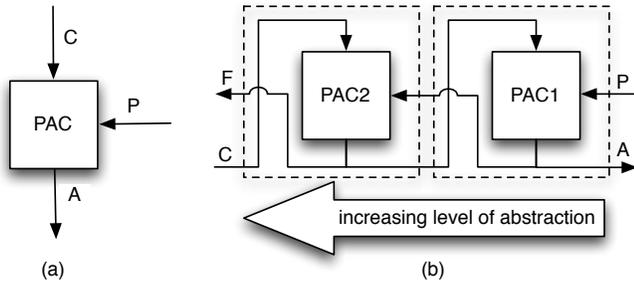


Fig. 3. (a) PAC module with control input C , percept input P , and action output A . (b) PAC module as building block for implicit hierarchical structure. The action output of each module is split into two parts: feedback F to the next higher level and action A propagated to the next lower level.

V. IMPLICIT HIERARCHIES FOR ONLINE LEARNING

The training of the transition matrix and the verification network should preferably happen online by incrementally updating the existing system. In the COSPAL project, it has been suggested to use a system structure of three layers [22], where the respectively higher levels take care of controlling the training of lower levels. Similar to many other suggestions for cognitive system structures, e.g. the ECOM model [23], the number of levels and the nature of connections has been part of the architecture specification.

If perception-action loops are supposed to have higher functionalities than purely reactive or homeostatic behavior, i.e., if the loops are supposed to perform more than just passive tracking, some driving force is required. This driving force enters the perception-action loop through a control input C , which complements percept input P and action output A , see Fig. 3 (a). The modified perception-action-control loop is called PAC module in what follows.

The control input of a PAC module is not an observation in the sense of Bayesian tracking, since it affects the prediction step and not the update step. It is not part of the state vector of the respective PAC module either, since it is by definition modified outside the PAC module in order to produce a non-homeostatic behavior. The only possible way to embed control signals into a Bayesian tracking loop is to make it part of *another* PAC module that is located at a higher level, see Fig. 3 (b).

As a consequence of this cross-level connection between PAC modules, modules that previously lived at a single level get ordered in a hierarchical way. In contrast to earlier mentioned architectures, this hierarchy is dynamic, as connections between the PAC modules might be modified or entirely removed within the respective modules. In order to make the hierarchy functional, feedback signals from the respectively lower-level modules are required, which might be considered as part of the action space. These signals are typically binary or fuzzy events, see also discussion above on representing symbols in channel representations.

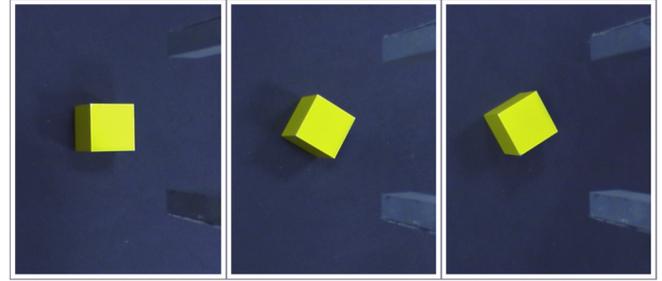


Fig. 4. Some views of a cube that has been used for the experiments.

VI. EXPERIMENTAL RESULTS

In this section experiments validating the concept of channel-based Bayesian tracking within a single PAC module are discussed. The test data has been taken from the final demonstrator of the COSPAL project¹, a shape sorter puzzle that has been learned to be solved by an artificial cognitive system. In particular, views of rotated instances of the same object, e.g. a cube, are considered, see Fig. 4. The rather simplistic scenario has been chosen to validate the tracking method independently of shortcomings of the feature extraction method.

The considered state space of the PAC module consists of the concatenation of two channel representations: one for the current orientation of the object and one for the previous one. In this way, linear dynamics of the object are covered as discussed earlier. The whole algorithm is given by the following steps:

- 1) Compute prototype descriptors f_p for the learning set
- 2) Compute (12) to establish a system model
- 3) Initialize state channel vector
- 4) Iterate:
 - a) Compute prediction (10)
 - b) Compute verification vector (13) from new image descriptor f_k
 - c) Compute correction (11)
 - d) Decode maximum for state channel vector

The channel representations have been built for 6 and 9 orientation channels, which corresponds to 30 degrees and 20 degrees distance between the prototypes. The tracking has been initialized by assuming to be close to the angle zero with unknown previous angle, i.e., the state vector is initialized as a concatenation of an impulse and a uniform distribution. The error of the decoded angles for 9 channels is about five times lower than for ordinary histogram bins, see Fig. 5.

The numerical evaluation gives the following results. For 6 channels and 12 prototypes, i.e., one prototype every 30 degrees, the standard deviation has been measured as 2.8 degrees. For 9 channels and 18 prototypes, i.e., one prototype every 20 degrees, the standard deviation has been measured as 1.3 degrees. The oscillations of the error in Fig. 5 is caused

¹<http://www.cospal.org>

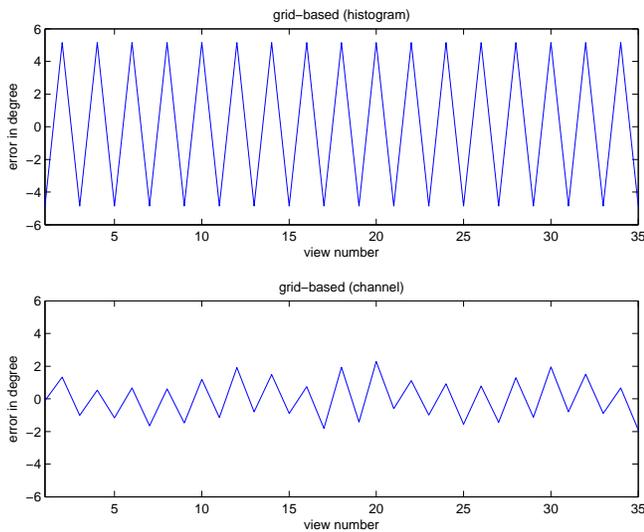


Fig. 5. Tracking error for 9 orientation channels. Top: ordinary grid-based method using histograms. Bottom: channel-based tracking.

by the remaining quantization effect of channel decoding which is about five percent of the channel width. This quantization introduces a small systematic error towards the center of the channels such that the estimates are slightly biased.

VII. CONCLUSIONS AND FUTURE WORK

A. Conclusions

The experimental results for the PAC module established as a channel-based Bayesian tracker shows that the channel-based technique provides an accuracy that is higher than the expected accuracy for a grid-based method. The results allow to use the proposed methodology in larger, learning based systems that make use of several connected PAC modules, which form an implicit hierarchy. Learning within this structure is entirely governed by the individual PAC modules.

B. Future Work

In future work channel-based tracking will be compared with state-of-the-art particle filters. Furthermore, experiments on hierarchical networks of PAC module have to be performed in order to validate the theories about (online) learning of PAC modules and implicit hierarchies.

VIII. ACKNOWLEDGMENTS

Most of the sketched ideas and achieved results are based on discussions with our project partners from the DIPLECS project and members from the Computer Vision Laboratory at Linköping University. We appreciate in particular the discussions with Erik Hollnagel on the topics of Neisser's perceptual loop and different levels of perception as well as with Richard Bowden on the topics of implicit perception-action hierarchies and definitions of cognitive systems.

REFERENCES

- [1] U. Neisser, *Cognition and Reality: Principles and Implications of Cognitive Psychology*. W. H. Freeman, San Francisco, 1976.
- [2] G. H. Granlund, "A Cognitive Vision Architecture Integrating Neural Networks with Symbolic Processing," *Künstliche Intelligenz*, no. 2, pp. 18–24, 2005, ISSN 0933-1875, Böttcher IT Verlag, Bremen, Germany.
- [3] M. Felsberg, P.-E. Forssén, A. Moe, and G. Granlund, "A COSPAL subsystem: Solving a shape-sorter puzzle," in *AAAI Fall Symposium: From Reactive to Anticipatory Cognitive Embedded Systems*, ser. AAAI Technical Report Series, no. FS-05-05, AAAI. Crystal City, Arlington, Virginia USA: AAAI Press, November 2005, pp. 65–69.
- [4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [5] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [6] G. H. Granlund, "An Associative Perception-Action Structure Using a Localized Space Variant Information Representation," in *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.
- [7] B. Johansson, T. Elfving, V. Kozlov, Y. Censor, P.-E. Forssén, and G. Granlund, "The application of an oblique-projected landweber method to a model of supervised learning," *Mathematical and Computer Modelling*, vol. 43, pp. 892–909, 2006.
- [8] I. P. Howard and B. J. Rogers, *Binocular Vision and Stereopsis*. Oxford University Press, Oxford, UK, 1995.
- [9] R. S. Zemel, P. Dayan, and A. Pouget, "Probabilistic interpretation of population codes," *Neural Computation*, vol. 10, no. 2, pp. 403–430, 1998.
- [10] A. Pouget, P. Dayan, and R. Zemel, "Information processing with population codes," *Nature Reviews – Neuroscience*, vol. 1, pp. 125–132, 2000.
- [11] W. Press et al., *Numerical Recipes in C*. Cambridge University Press, 1994.
- [12] P.-E. Forssén, "Low and medium level vision using channel representations," Ph.D. dissertation, Linköping University, Sweden, 2004.
- [13] E. Jonsson and M. Felsberg, "Correspondence-free associative learning," in *International Conference on Pattern Recognition*, Hong Kong, August 2006.
- [14] H. P. Snippe and J. J. Koenderink, "Discrimination thresholds for channel-coded systems," *Biological Cybernetics*, vol. 66, pp. 543–551, 1992.
- [15] E. Pampalk, A. Rauber, and D. Merkl, "Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*. Madrid, Spain: Springer, August 27-30 2002, pp. 871–876.
- [16] M. Felsberg, P.-E. Forssén, and H. Scharf, "Channel smoothing: Efficient robust smoothing of low-level signal features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 209–222, 2006.
- [17] M. Felsberg and G. Granlund, "P-channels: Robust multivariate estimation of large datasets," in *International Conference on Pattern Recognition*, Hong Kong, August 2006.
- [18] M. Felsberg, "Wiener channel smoothing: Robust Wiener filtering of images," in *DAGM 2005*, ser. LNCS, vol. 3663. Springer, 2005, pp. 468–475.
- [19] G. H. Granlund, "In search of a general picture processing operator," *Computer Graphics and Image Processing*, vol. 8, pp. 155–173, 1978.
- [20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [21] C. Spence and S. Squire, "Multisensory integration: Maintaining the perception of synchrony," *Current Biology*, vol. 13, pp. R519–R521, 2003.
- [22] M. Felsberg, J. Wiklund, E. Jonsson, A. Moe, and G. Granlund, "Exploratory learning structure in artificial cognitive systems," in *International Cognitive Vision Workshop*, 2007.
- [23] E. Hollnagel and D. D. Woods, *Joint cognitive systems: Foundations of cognitive systems engineering*. Boca Raton, FL: Taylor & Francis, 2005.