

P-Channels: Robust Multivariate M-Estimation of Large Datasets

Michael Felsberg
Linköping University
Computer Vision Laboratory
mfe@isy.liu.se

Gösta Granlund
Linköping University
Computer Vision Laboratory
gosta@isy.liu.se

Abstract

In this paper we introduce a new technique that allows to estimate modes of a high-dimensional probability density function with linear time-complexity in the number of dimensions and the number of samples. The method can be implemented in an order-independent incremental way, such that the space-complexity is linear in the number of dimensions and the number of modes. The number of required samples to get reliable estimates depends linearly on the number of dimensions even if we replace the assumption of independent stochastic variables with the weaker assumption of data clustered in submanifolds. These submanifolds need not to be known, but smoothness assumptions are made. The new technique is based on representing data in what we call P-Channels.

1. Introduction

This paper is addressing the issue of robust multi-dimensional M-estimation on large datasets, which is a frequently met problem in computer vision and pattern recognition.

1.1. M-Estimation and Clustering

M-estimation is closely related to clustering in a way that the modes of a multivariate distribution can be estimated as the centroids of the data clusters. Implementing M-estimation in this way means to split the estimation into two steps: clustering and parameter estimation of the cluster. Without having a priori knowledge (as e.g. class labels), we are restricted to local, unsupervised clustering, i.e., M-estimation is related to k -means types of methods [7] which are related to Kohonen nets [14], learning of vector quantization [12, 20], and mean-shift clustering [6].

The approach that we propose is more related to the equal-interval-width method [5], with the difference that we combine the clustering and the estimation step, such that the

influence of the fixed intervals (the quantization effect) is reduced significantly. In contrast to k -means, our method can be applied incrementally and with lower complexity, c.f. [12], and generalization for, e.g., circular domains is trivial, c.f. [4]. In contrast to the leader algorithm (also called Taylor-Butina algorithm [3]), our method is not order dependent, c.f. [12].

Despite the relation to unsupervised clustering, the proposed method should not be reduced to a pure vector quantization or discretization as it is often required for, e.g., decision trees [16], Chapter 18. The method maintains the full continuous information about the modes and can be subject to further soft-computing techniques, e.g. fuzzy methods, clustering ensembles [19], or associative networks [10].

One fundamental problem for M-estimation in the multivariate case is the curse of dimensionality. If we cannot assume, e.g., independent stochastic variables, we have to perform the estimation on the joint probability density of the full space. In order to get reliable empirical distributions, the number of required samples grows exponentially with the number of dimensions [1].

1.2. Approaches for Reducing the Dimensionality

Many approaches try to deal with the curse of dimensionality by reducing the dimensionality based on the assumption of statistical independence of subspaces. A standard technique is to perform a PCA, Karhunen-Loève transform, or an SVD to the data,¹ which aim at identifying the non-zero subspace of the input space. This subspace can be further decomposed by the Independent Component Analysis (ICA) [11]. Assuming sufficiently independent dimensions, one can determine the joint distribution through the marginals, which reduces the number of necessary samples to increase linearly instead of exponential.

The problem with all named approaches is that they act

¹Note that PCA is just another name for the Karhunen-Loève transform and that both are algebraically equivalent to an SVD on the data matrix - c.f. also the Eckart-Young-Mirsky theorem.

on subspace, i.e., we assume that the data can *globally* be separated into subspaces. If the data lives on a curved or folded manifold, these approaches fail to significantly reduce the dimensionality. One approach to straighten manifolds is to virtually embed these into a higher-dimensional space and to use linear discrimination (subspace) methods in the embedding space. By applying the *kernel trick*, the required scalar products are computed directly in the lower-dimensional space - see e.g. support vector machines [17]. The difficulty with this approach is to identify the suitable low-dimensional space and to find the appropriate kernel.

1.3. The Channel Representation

Another approach which is based on high-dimensional embedding is the more biologically motivated channel representation [10, 18]. It is based on the idea of placing local functions, the *channels*, pretty arbitrarily in space and to project the data onto the channels - i.e., we have some kind of (fuzzy) voting. The most trivial case are histograms, but their drawback of losing accuracy is compensated in the channel representation by knowledge about the algebraic relation between the channels.

The projections onto the channels result in tuples of numbers which - although often written as vectors (bold-face letters) - do not form a vector space. In particular the value zero (in each component) has a special meaning, *no information*, and need not be stored in the memory.

Formally, the channel representation is obtained from a finite set of *channel projection operators* F_n . These are applied to the feature vectors \mathbf{f} in a point-wise way to calculate the *channel values* p_n :

$$p_n = F_n(\mathbf{f}) \quad n = 1, \dots, N. \quad (1)$$

Each feature vector \mathbf{f} is mapped to a vector $\mathbf{p} = (p_1, \dots, p_N)$, the *channel vector*.

The projection operators can be of various form, e.g., \cos^2 functions, B-splines, or Gaussian functions [9]. The channel representation can be used in different contexts, but typically it is applied for associative learning [13] or robust smoothing [8]. In context of robust smoothing it has been shown that summing B-spline channel vectors of samples from a stochastic variable ξ results in a sampled kernel density estimate of the underlying distribution $p(\xi)$:

$$E\{\mathbf{p}\} = E\{[F_n(\xi)]\} = (B_2 * p)(n). \quad (2)$$

The global maximum of p is the most probable value for ξ and for locally symmetric distributions, it is equivalent to the maximum of $B_2 * p$. The latter can be approximately extracted from the channel vector \mathbf{p} using an implicit B-spline interpolation [8] resulting in an efficient semi-analytic method. The extraction of the maximum can therefore be considered as a functional inverse of the projection onto the channels.

In what follows, we name the projection operation also *channel encoding* and the maximum extraction *channel decoding*.

2. P-Channel Method

In this section we introduce a novel type of channel representation which overcomes the major limitation of the B-spline channels as proposed in [8]. The major drawback of channel representations in context of mode estimation is the quantization effect, i.e. a bias towards the channel centers. This bias is nearly entirely removed by the semi-analytic B-spline scheme, but the latter method is hard to generalize to multiple dimensions. Especially in the case of high dimensions a quantization-free decoding method would be extremely useful, as dense channel vectors should be avoided. The projective channels, *P-Channels*, which are proposed below, allow a nearly quantization free decoding in multiple dimensions and increase the sparseness compared to other channel representations.

2.1. 1D P-Channels

The idea of P-Channels is borrowed from projective geometry where homogeneous coordinates are used to represent translations as linear mappings and where vectors are invariant under global scalings. The P-Channels are obtained by dropping the requirements for real-valued and smooth basis functions for channel representations. Instead, we consider rectangular basis functions as in the case for histograms. Since rectangular basis functions do not allow exact reconstruction, we simply add a second component which stores the offset from the channel center. As a consequence, the channels become vector-valued (boldface letters) and the channel vector becomes a matrix (boldface capital). The encoding is visualized in Fig. 1.

For a single value f , we obtain the P-Channels as follows. Without loss of generality we scale the values such that the channels are located at integer positions. The value f is *virtually* placed at $(f, 1)^T$, but it is accounted only to the channel with the center $[f]$, where $[f]$ is the closest integer to f , and it is transformed to the local coordinate system of this channel, i.e.,

$$\mathbf{p}_i = \begin{pmatrix} p_{1i} \\ p_{2i} \end{pmatrix} = \delta(i - [f]) \begin{pmatrix} f - i \\ 1 \end{pmatrix}, \quad (3)$$

where δ denotes the Kronecker delta. Hence, the first component of the channel contains the linear offset from the channel center. If several values f_j are to be encoded into the same channel vector, we obtain

$$\mathbf{p}_i = \sum_j \delta(i - [f_j]) \begin{pmatrix} f_j - i \\ 1 \end{pmatrix}, \quad (4)$$

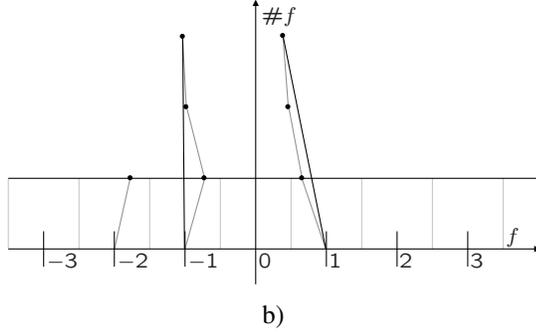
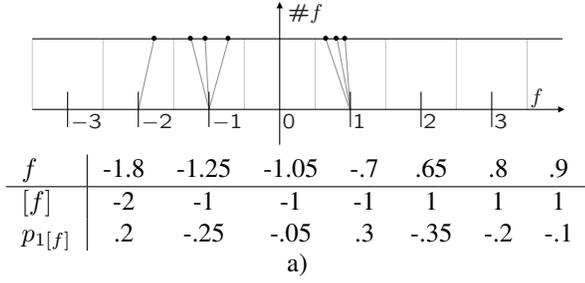


Figure 1. P-Channel encoding visualized. a) The values f are encoded into the vectors \mathbf{p}_i with $p_{1[f]}$ as indicated in the table. All $p_{i \neq [f]} = 0$ and $p_{2[f]} = 1$. b) Summing the P-Channels results in $\mathbf{P} = \begin{pmatrix} 0 & .2 & 0 & 0 & -.65 & 0 & 0 \\ 0 & 1 & 3 & 0 & 3 & 0 & 0 \end{pmatrix}$.

i.e., the second component is an ordinary histogram bin and counts the number of values contributing to the first component.² Hence, it serves as a normalization component and the linear average can be obtained as

$$\bar{f} = \frac{\sum_i i p_{2i} + p_{1i}}{\sum_i p_{2i}}. \quad (5)$$

In other words: the channel index defines the integer position, the first component the weighted offset, and the second component the normalization factor. The division by the normalization factor in (5) motivates the use of 'P-Channels for 'projective'.

If a further value f is added to an existing channel vector, we obtain

$$\mathbf{p}_i + \delta(i - [f]) \begin{pmatrix} f - i \\ 1 \end{pmatrix}, \quad (6)$$

which allows an *incremental* computation of the representation.

²Note: if the variances of the values f_j are known, the encoding can be modified by weighting the offset and the normalization with the inverse variance σ_j^{-2} . It can then easily be shown that the estimate according to (5) results in the maximum likelihood estimate for variables with Gaussian distribution.

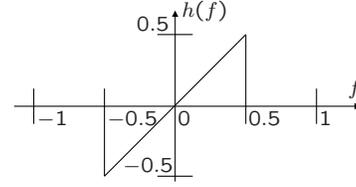


Figure 2. Influence function $h(f)$ of P-Channels.

If we want to replace the linear estimate in (5) with a robust M-estimate, we have to split the channel vector into partitions corresponding to the different clusters in the data or modes in the underlying distribution. To keep things simple, we start with the strongest mode and assume that the corresponding data cluster lies entirely in a single channel. Later we will relax these conditions.

The normalization components p_{2i} correspond to the histogram of the data or, equivalently, to a sampled kernel density estimate. Hence, $i_0 = \max_i p_{2i}$ indicates the channel index at the strongest mode. The position of the mode is computed as

$$\hat{f} = i_0 + \frac{p_{1i_0}}{p_{2i_0}} \quad (7)$$

corresponding to a piecewise linear *influence function*, cf. Fig. 2, i.e., we apply a truncated quadratic error norm [2].

2.2. Reducing the Quantization Effect

In general, we cannot assume that clusters fall entirely a single channel. If a cluster covers more than one channel, decoding of isolated channels leads to biased estimates of the modes in such a way that they are moved towards the channel center. This effect is called quantization effect of the channel representation [8], see Fig. 3.

To reduce the quantization effect, the neighbored channels have to be considered as well, such that the influence of an offset Δf is minimized. That means if all values f_j are shifted by Δf , the M-estimate \hat{f} must also be shifted by Δf :

$$f_j \mapsto f_j + \Delta f \Rightarrow \hat{f} \mapsto \hat{f} + \Delta f. \quad (8)$$

Unfortunately, this cannot be achieved in general, not even in expectation sense, since the distribution of f_j is unknown and since we can place our influence function (Fig. 2) only at integer positions. However, for certain distributions of f_j , we can derive an analytic expression of the neighbored influence functions with zero bias in expectation sense.

For instance, we assume uniformly distributed f_j in $[\Delta f - 1/2; \Delta f + 1/2]$ with $\Delta f \in [0; 1]$. Integer shifts of the

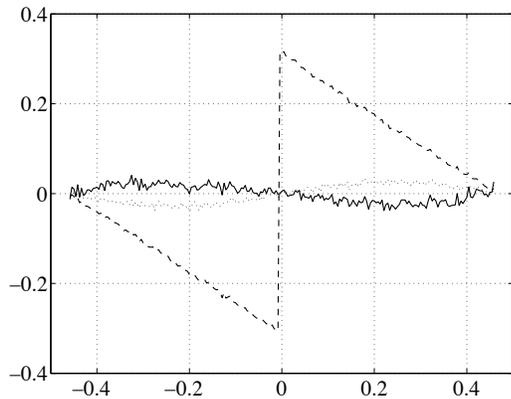


Figure 3. Quantization effect of channel representations: Comparison of B-spline decoding [8] (dotted line), P-Channel decoding of single channels according to (7) (dashed line), and decoding of pairs of channels according to (12) (solid line). The axes are normalized w.r.t. the channel width, the origin lies on a channel boundary. Each estimate is computed from 10^4 samples drawn from a normal distribution with variance one.

interval are obtained by index manipulation. We decode by combining the two channels adjacent to Δf , i.e., channels \mathbf{p}_0 and \mathbf{p}_1 . The expectation values of the components are

$$\mathbf{p}_0 = \begin{pmatrix} \frac{1}{2}(\Delta f - \Delta f^2) \\ 1 - \Delta f \end{pmatrix}, \mathbf{p}_1 = \begin{pmatrix} \frac{1}{2}(\Delta f^2 - \Delta f) \\ \Delta f \end{pmatrix}. \quad (9)$$

If these channels are decoded separately, we obtain respective biases as

$$\frac{p_{10}}{p_{20}} - \Delta f = -\frac{\Delta f}{2}, \quad 1 + \frac{p_{11}}{p_{21}} - \Delta f = \frac{1 - \Delta f}{2}. \quad (10)$$

These biases compensate each other if we combine the estimates from \mathbf{p}_0 and \mathbf{p}_1 according to

$$(1 - \Delta f) \frac{p_{10}}{p_{20}} + \Delta f \frac{p_{11}}{p_{21}} = \Delta f. \quad (11)$$

Solving this equation yields the decoding formula

$$\Delta f = \frac{p_{10}p_{21}}{p_{10}p_{21} - p_{11}p_{20}}. \quad (12)$$

As can be seen in Fig. 3, this bias compensation reduces the quantization effect by nearly one order of magnitude, even if the distribution is not uniform, but a normal distribution. For uniformly distributed data, the quantization is fully suppressed.

2.3. ND P-Channels

The multi-dimensional P-Channel representation is based on an advanced algebraic structure given by products of paravectors [15], but for practical purposes, one can use the following inductive definition. The closest integer operator for vectors is defined as an element-wise closest integer operator, i.e., index $[\mathbf{f}]$ means $([f_1], [f_2], \dots)^T$.

Let \mathbf{f} be an n -dimensional feature vector. The P-Channel representation \mathbf{P} of \mathbf{f} is defined for

$n = 1$: Eq. (3).

$n > 1$: Decompose $\mathbf{f} = (\mathbf{g}, f)^T$ where \mathbf{g} is an $(n - 1)$ -dimensional feature vector. Let \mathbf{Q} be the P-Channel representation of \mathbf{g} . The components of \mathbf{P} are given as

$$p_{ki} = 0 \text{ for } i \neq [\mathbf{f}] \quad (13)$$

$$p_{k[\mathbf{f}]} = q_{k[\mathbf{g}]} \text{ for } k \in \{1, \dots, n\} \quad (14)$$

$$p_{(n+1)[\mathbf{f}]} = (f - [\mathbf{f}])q_{(n+1)[\mathbf{g}]} \quad (15)$$

$$p_{(n+2)[\mathbf{f}]} = q_{(n+2)[\mathbf{g}]} \quad (16)$$

It can be shown by induction that independently of the measurement dimension, a single measurement leads to exactly one single non-zero channel. This observation leads to an upper bound of non-zero channels – and connected to that to the number of values to be stored – for the P-Channel representation: Given M measurements of dimension N added up in the P-Channel representation requires at most $M(2N + 1)$ values to be stored. In the worst case, all M measurements fall into different channels. For each non-zero channel its ND index vector and its $(N + 1)$ D value need to be stored.

As soon as several measurements fall into the same channel, the number of values to be stored decreases. If we assume that the measurements are drawn from C clusters of hypervolume V in units of channel-distances to the power of N , i.e., each cluster covers V channels, and if we further assume L outliers, i.e., spurious measurements somewhere outside the clusters, the number of values to be stored is given by $(CV + L)(2N + 1)$. Note that V depends exponentially on N , i.e., we should try to use channels which are sufficiently large to cover whole clusters. If all clusters fit entirely into respectively single channels, the number of values to be stored depends linearly on the dimensionality and is constant in the number of samples (up to the outliers).

From a learning-theoretic point of view, the P-Channels memorize single events and average or generalize close-by multiple events. Time and space complexity are clearly sub-linear if the data clusters such that generalizing is actually applied. Search times in the potentially huge channel space are significantly reduced by using hash-tables on the index set. Note in this context that the P-Channels should not be stored as ordinary matrices, but as sparse matrices in order to benefit from the low time- and space complexities.

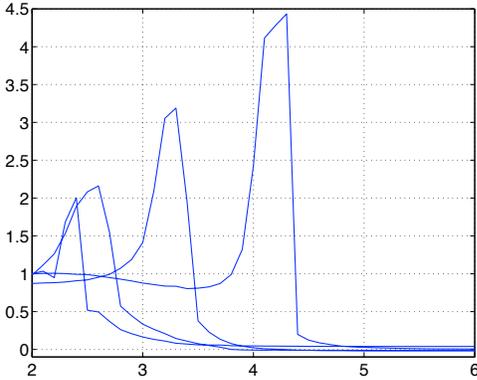


Figure 4. Discrimination test for the P-Channel method. On the abscissa, the distance between the two modes (with variance one) is given. On the ordinate the error of the mode estimate (left mode) is given. The four graphs illustrate the error for four different channel widths: 1, 2, 3, and 4 (from left to right).

3. Experiments

In this section we investigate two properties of the P-Channel representation more in detail: the discrimination capability and the performance for high-dimensional feature spaces.

3.1. Discrimination Capabilities

For robust methods it is of central importance to know the minimum distance between two modes which can still be distinguished as different modes. Typically, the M-estimates become more and more biased as the modes get closer to each other until the estimation breaks down and the two modes are mixed. To test the discrimination capability of P-Channels, we set up the following experiment.

We draw respectively 10^4 samples from two normal distributions with variance one and means at different positions. We successively reduce the distance between the means until the P-Channels cannot distinguish between the two modes. We repeat this experiment for different channel resolutions, i.e., different ratios between variance and channel width. The results are illustrated in Fig. 4

According to this figure (and as expected), the discrimination breaks down earlier for larger channel widths than for smaller widths. The break-down points are at a distance of about 2.5 (channel width 1), 2.8 (channel width 2), 3.5

(channel width 3), and 4.4 (channel width 4). When comparing these results with other methods, one should keep in mind that the P-Channel method does not assume a normal distribution of data, i.e., it should not be compared with e.g. Gaussian mixture models.

3.2. Performance on High-Dimensional Feature Spaces

In this section, we run some performance tests in order to illustrate the theoretic complexity estimates with some realistic timings. The platform that we use is Matlab 7.2 on a PowerBook with 1GHz PPC. In all cases the limitation for the dimensionality was the limited index range for Matlab matrices given by 32 bit and not the computational time or the memory requirements.

The implementation is written in native Matlab code using the Sparse toolbox, i.e., we had no influence on the internal processing of sparse matrices, e.g., searching indices in hash tables could not be realized.

In the first performance experiment, we constructed the P-Channel representations for 5000 respective 10^4 samples in spaces of dimension 1-9 with 10 respective 20 channels in each dimension. Each experiment was repeated 10 times to give reliable time estimates. The time consumption increases approximately linearly in the dimensions and the number of samples, see Fig. 5 a), with an upper bound of 0.4 sec for the most time consuming variation.

In the second performance experiment, we decoded P-Channel representations for clustered and scattered data with 10^k respective 20^k channels, see Fig. 5 b). The decoding is also linear in the number of dimensions, but one order of magnitude faster than the encoding. Hence, the effort for the decoding can be neglected in most cases.

4. Conclusion

In this paper, we have proposed the P-Channel representation as a method for robust M-estimation for large datasets in high dimensions. The new method is a combination of classical histogram techniques and calculations known from projective geometry. Using an advanced decoding method, the M-estimates become bias-free (quantization-free) for known distributions and approximately bias-free otherwise. We have shown that the method has low upper bounds on time and space complexity. For M measurements of dimension N there are at most $M(2N+1)$ values to be stored and processed. In typical applications with C clusters of hyper-volume V and L outliers, there are $(CV+L)(2N+1)$ values to be stored and processed.

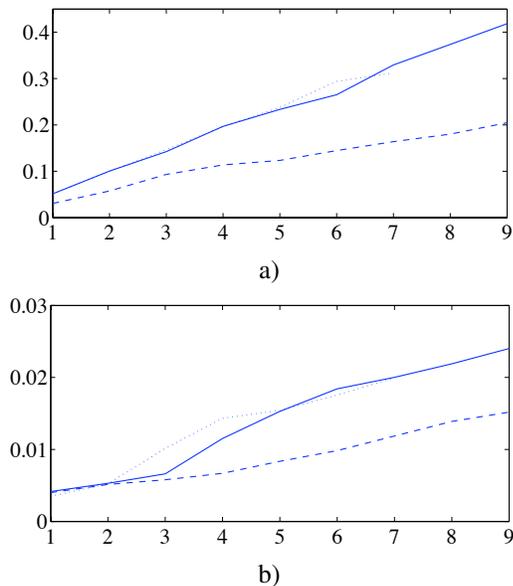


Figure 5. Time consumption for encoding (a) and decoding (b). Ordinate: dimensions $k = 1, \dots, 9$, abscissa: time in seconds. a) Dashed line: 5000 samples, 10^k channels. Solid line: 10^4 samples, 10^k channels. Dotted line: 10^4 samples, 20^k channels. b) Dashed line: 10^4 clustered samples, 10^k channels. Solid line: 10^4 scattered samples, 10^k channels. Dotted line: 10^4 samples, 20^k channels.

Acknowledgment

This work has been supported by EC Grants IST-2003-004176 COSPAL and IST-2002-002013 MATRIS. This paper does not represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

References

- [1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.
- [2] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.
- [3] D. Butina. Unsupervised data base clustering based on daylight's fingerprint and tanimoto similarity: A fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.*, 39:747–750, 1999.
- [4] D. Charalampidis. A modified k-means algorithm for circular invariant clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1858–1865, 2005.
- [5] M. R. Chmielewski and J. W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15:319–331, 1996.
- [6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.
- [7] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In A. Prieditis and S. Russel, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, 1995.
- [8] M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, 2006.
- [9] P.-E. Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, 2004.
- [10] 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.
- [11] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.
- [12] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [13] B. Johansson. *Low Level Operations and Learning in Computer Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, December 2004. Dissertation No. 912, ISBN 91-85295-93-0.
- [14] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, 1997.
- [15] I. R. Porteous. *Clifford Algebras and the Classical Groups*. Cambridge University Press, 1995.
- [16] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.
- [17] B. Schoelkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, P. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transaction on Signal Processing*, 45(11):2758–2765, 1997.
- [18] H. P. Snippe and J. J. Koenderink. Discrimination thresholds for channel-coded systems. *Biological Cybernetics*, 66:543–551, 1992.
- [19] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.
- [20] K.-L. Wu and M.-S. Yang. Alternative learning vector quantization. *Pattern Recognition*, 39:351–362, 2006.