

Representation and Learning of Invariance

Klas Nordberg Gösta Granlund Hans Knutsson

LITH-ISY-R-1552
1994-01-21

Representation and Learning of Invariance

Klas Nordberg Gösta Granlund Hans Knutsson
Computer Vision Laboratory,
Department of Electrical Engineering
Linköping University, S-581 83 LINKÖPING, SWEDEN
Phone: +46-13-28 16 34, Telefax: +46-13-13 85 26

February 9, 1994

Abstract

A robust, fast and general method for estimation of object properties is proposed. It is based on a representation of these properties in terms of channels. Each channel represents a particular value of a property, resembling the activity of biological neurons. Furthermore, each processing unit, corresponding to an artificial neuron, is a linear perceptron which operates on outer products of input data. This implies a more complex space of invariances than in the case of first order characteristic without abandoning linear theory. In general, the specific function of each processing unit has to be learned and a fast and simple learning rule is presented. The channel representation, the processing structure and the learning rule has been tested on stereo image data showing a cube with various 3D positions and orientations. The system was able to learn a channel representation for the horizontal position, the depth, and the orientation of the cube, each property invariant to the other two.

Keywords: Representation, channels, learning, estimation.

1 Introduction

Estimation procedures of object properties from image data often suffer from computational complexity as well as from sensitivity to image noise. As an example, estimation of depth from stereo is often implemented as a correlation between the left and right images in order to determine disparity from which depth can be computed [24] [13]. Though simple in its nature, this correlation process is computationally tough. Furthermore, in presence of image noise we would either expect the accuracy of the estimate to decrease more or less gracefully or we would have to employ more complex correlation methods. The same argument holds for many state-of-the-art algorithms like depth from shading [14], and depth from motion [11]. Also algorithm classes such as optical flow algorithms [11], object recognition algorithms [5] [4] [6] [19] and algorithms for shape description [14] [2] have these characteristics. In short, it does not seem possible to design an algorithm for extraction of object properties from image data which is not complex or computationally heavy and, in addition, robust with respect to image noise as well as to accuracy in the representation of data.

All algorithms for estimation of object properties have one thing in common, invariance. As an example, estimation of depth implies that the depth estimate is invariant to all movements of an object which leave the depth constant. Another example is recognition of objects where we wish the object label to be invariant to e.g. position, orientation and size of the object. In practice, however, these invariances emerge in quite different ways. An object detector, for instance, may detect a specific object by correlating a template with any possible image region and each image in various scale in order to obtain invariance to position and size. Invariance to orientation may be accomplished by employing a set of templates showing the object from different view points. On the other hand, a depth-from-stereo algorithm is often based on a disparity estimator from which depth is calculated by compensating for camera parameters such as base-line and vergence.

Usually, we represent information using scalars or vectors which correspond to physical entities like Cartesian or angular coordinates in two or three dimensions, velocities, etc. These are natural to use since they possess an intuitive interpretation in terms of what is happening in the image scene. If, however, estimated properties like e.g. depth should be invariant to a number of such scalars, we would have to figure out an expression for this property which does not contain any of the invariant scalars, or an expression where the influence of the invariant scalars is negligible. Often, there is no obvious way how to do this and the resulting expression and algorithm will be highly dependent on the application. As consequence, algorithms tend to be highly application specific and the way they represent information incompatible.

In the following, we will investigate a particular representation of information, called channel representation. This representation is demonstrated to have useful properties used in conjunction with a specific processing structure of second order characteristic, implying a non-trivial space of invariances. Hence, by employing the channel representation it can be expected that quite complex object properties like e.g. depth from stereo can be calculated in a very simple manner. As a consequence of its construction, each channel has to learn its specific transfer function in a neural network manner. A learning rule is presented for this task which has proven both fast and robust provided that the channel representation is used. An example of how the representation and processing structure can be used is also included.

2 The Channel Representation

One of the main characteristics of a biological neuron is that it gives a sharp response to some very specific events in its corresponding receptive field. This response gradually decreases when the event changes to something else, but it may also be the case that the response is constant to some specific transformation of the input pattern. As an example, consider a complex cell in the primary visual cortex, as described by Hubel and Wiesel [15]. Such a cell gives a sharp response provided that its receptive field contains a linear structure with a specific orientation. If the orientation of the input pattern changes, the response from the complex cell decreases and eventually vanishes, usually after some ten degrees. If, however, the position of the pattern changes within the receptive field, e.g. by translating an edge with constant orientation, the response remains the same. Neurons in this part of the cortex also have a similar characteristic with respect to spatial frequency [7] [23]. Based on psychophysical experiments, a multi-channel hypothesis has been suggested by Blakemore and Campbell which assumes that groups of neurons are tuned to specific frequencies [20].

The activity of neurons can, with a certain degree of simplification, be characterized as either high, low, or something in between. To model this behaviour, we may use a scalar in the range zero to one. A unit value simply means that the input pattern belongs to the class for which this cell gives optimal response, while a zero value indicates that the cell is completely insensitive to that particular pattern. Looking at an individual cell is not enough, however. An object property can not be represented by a single cell, but rather a group of cells, where each cell represents a particular value of that property. Hence, when the response of one cell decreases, indicating that the value of some property changes from the optimal value of that cell, the response of some other cell increases, corresponding to the property value approaching the optimal value of this cell. To represent a continuously varying property, e.g. position or orientation, there is no need for a continuous spectra of cells. Instead, experimental results indicate that the cells partition e.g. the orientation range of some region of the visual field into a finite number of overlapping patches, each patch corresponding to one or a few orientation selective cells.

Findings from biological vision are not always applicable to image processing tasks, but one may ask oneself why biological neural networks have chosen this particular representation of information. Ballard has investigated the structural and functional consequences of this representation in the cortex [1]. In this paper, it is argued that this particular representation is useful from a computational point of view. In this section we will formalize this representation and in subsequent sections see that it has some interesting properties regarding how invariances can be coded in sets of such information descriptors. The representation will also be tested on an imaging task. The presentation is based on the magnitude representation suggested by Granlund [10]. Apart from being related to biological results, as mentioned above, it also relates to ideas in computer vision based on signal processing. As an example of the last case, it will become evident that the concept of channels can be applied to e.g. estimation of local orientation by filtering [9] [17] [16]. It is also applicable to joint space and frequency representations such as the Gabor representation [21].

Consider an object, be it a physical object like a solid cube, or an abstract object like a trajectory of a physical object in a 3D scene. To any such object we assign various properties like position, size, direction, orientation, etc, and to each such property there is a corresponding value. For a specific property let its value be denoted p , and let $p \in M$,

where M is the range of p . Let 2^M denote the set of all subsets of M and let M be equipped with some measure of dissimilarity, denoted d where $d : M \times 2^M \rightarrow \mathbb{R}$ such that

$$\begin{aligned} d(x, Y) &\geq 0 && \text{for all } x \in M \text{ and } Y \in 2^M, \\ d(x, Y) &= 0 && \text{if and only if } x \in Y. \end{aligned} \tag{1}$$

In some cases, d can be defined in terms of a norm on some vector space, but in other cases d must be defined in more intuitive terms. Choose a set $\{N_j\}$ where each N_j is a subset of M . Each N_j will correspond to a channel and each channel will have a value c_j which is one when $d(p, N_j) = 0$ and decreases monotonically to zero when $d(p, N_j)$ increases. It should be noted already here, that transformations of the object which leave $d(p, N_j)$ constant must also leave c_j constant, implying that the value of that channel is invariant to all such transformations of the object.

Given the above definition of channels, it is not obvious how they should be characterized more specifically, neither with respect to how the subsets N_j are chosen nor how c_j depend on $d(p, N_j)$. These two properties are, however, related for a set of channels. As mentioned above, the channels should be overlapping, resulting in a continuous transition from one channel to the next when the value of p changes. In general, we demand that the subsets N_j and the mapping from $d(p, N_j)$ to c_j must be chosen in such way that p , the property value, is represented uniquely. There should not be some value of p which is of interest to represent that causes all channels to have zero value or a pair of channels to have unit value, since that would constitute an ambiguous statement about p . In the following section, a specific choice of both the subsets N_j and the function which maps $d(p, N_j)$ to c_j are suggested.

As an example of how the channel representation works, we may consider a simple scalar, p . This may for instance be the position of a physical object along some axis. We may as subsets N_j choose equidistant points along the axis, finitely many if the range of p is limited, otherwise enumerable many. The dissimilarity function d can here be defined simply as $d(p, N_j) = |p - n_j|$, where n_j is the single element of N_j . The map from $d(p, N_j)$ to c_j can be defined in terms of any smooth and unimodal function which has unit value when $d(p, N_j) = 0$ and which vanishes when $d(p, N_j) \rightarrow \infty$. An example that will be used in the following is

$$c_j = \begin{cases} \cos^2(p - n_j) & |p - n_j| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

As an example, if $n_j = j\frac{\pi}{2}$, the channels can be illustrated as in Figure 1.

This example can be extended to two dimensions. Consider two scalars p_1 and p_2 , e.g. corresponding to two-dimensional Cartesian coordinates of some physical object. If we regard these coordinates as two independent properties, the channels may be chosen such that we obtain two sets of channels, one set representing the p_1 coordinate and one representing the p_2 coordinate. In this case, the subsets N_j can for each set be chosen as straight lines parallel either to the p_1 -axis or to the p_2 -axis. The choice of dissimilarity function and c_j map can then be the same as above for each set. The difference between this and the former example is that in the latter case, the channels in one set exhibit invariance to transformation of the object position with respect to the other coordinate.

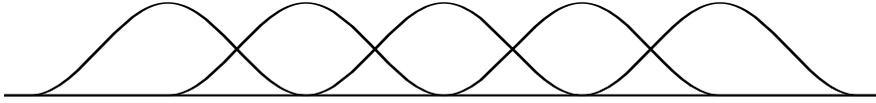


Figure 1: A set of channels shown as functions of p according to Equation (2)

Instead of regarding p_1 and p_2 as two separate properties, we may also see the pair (p_1, p_2) as *one* property and represent it using channels. In this case, each channel will represent a particular value of *both* p_1 and p_2 and will not be invariant to any transformation of either p_1 or p_2 . It will be the application or the situation that determine which of the two cases that should be implemented. The channel representation itself is indifferent to this issue and will work either way.

The last two examples only illustrate the use of the channel representation for some simple cases. In both cases, we would equally well have used the scalar p or the vector (p_1, p_2) as a representation for the object property. There are, however, other object properties for which the channel representation is more natural than using scalars. As an example, consider representation of local 2D orientation. A neighbourhood of an image which contains a linear structure can be represented for instance by the smallest angle between direction of variation in the neighbourhood and some fix line, resulting in a scalar, x , where $0 \leq x < \pi$. This scalar has, however, a discontinuity in the point $x = 0$, since arbitrarily small changes of orientation around this point causes a discontinuous jump in x between 0 and π . This discontinuity is not only an aesthetic flaw, it implies e.g. that the orientation descriptor x can not be used in averaging operations together with other descriptors.

If we instead represent the local orientation by using a set of channels this situation changes to the better. Choose as subsets

$$N_j = j\frac{2\pi}{3}, \quad j = 0, 1, 2, \quad (3)$$

and as a map from x to c_j

$$c_j = \cos^2[2(x - n_j)]. \quad (4)$$

The consequence of this choice is that e.g. channel c_0 will give unit response when the angle x is either 0 or π which is one and the same orientation. Each channel will thus have a unique optimal orientation with respect to its response and, furthermore, a local orientation in straight angle to that will give zero response. In this case the dissimilarity function looks something like

$$d(x, N_j) = |x - n_j| \bmod \frac{\pi}{2}. \quad (5)$$

The previous example is nothing but a reformulation of the algorithm for estimation of local orientation as defined by Knutsson in [17] based on a double angle representation

of local orientation proposed by Granlund in [9]. Instead of composing a vector for the representation from a set of filter responses, it is here suggested that the filter responses themselves can serve as a representation.

In the above example, three channels were used. Why not fewer or more? This relates to an important feature of channels, namely uniqueness of representation. In the above definition and examples of channels it has been discussed how values of object properties are mapped to sets of channels. To make a meaningful representation, however, this map should be unique, i.e. it should be possible to unambiguously determine p given the channels of its representation. Furthermore, this inversion process should not depend on the absolute values of the channels, but rather on their relative values. Hence, if the values of a set of channels were to be multiplied with some non-zero scalar, they should still represent the same p . This would of course violate the statement that the value of a channel lies in the range zero to one. But we may equally well say that the value of a channel primarily lies in that range, but that any set of channels which represents one and the same property value can be scaled by an arbitrary positive real number. We will discuss this feature more thoroughly in subsequent sections. Having realized this, it should be obvious that p can not be determined by considering one channel only, since the scaling factor is undetermined. Two channels would at most give some information about $d(p, N_j)$, the dissimilarity between p and the subset N_j . Three channels are the smallest set possible in order to determine p unambiguously, at least when the property can be represented by a periodic scalar. In more general cases, still more channels may be needed in order to determine p unambiguously.

The previous example illustrated how channels can represent an object property value, local orientation, continuously. Let us end this section with an extended example where the channel representation is superior to a standard scalar representation. Let the image neighbourhood under consideration have a well-defined spatial frequency as well as orientation. These two properties can of course be represented by two scalars, one for frequency and one for orientation. There is, however, one degenerate case where that representation fails to comply with our intuitive concept of a representation. When the neighbourhood is constant, i.e. the spatial frequency is zero, there is no local orientation to represent. In other words, the orientation is undefined and even if we choose to assign the orientation some default value in this case, the representation would not be continuous with respect to small changes of frequency and orientation around that point. If, however, we choose to see local orientation and spatial frequency as one single property, it can conveniently be represented using channels. As an example, we can use one channel to represent the zero frequency case in addition to a set of channels which covers various well-defined combinations of frequency and orientation. This is a reformulation of the Gabor representation of images in terms of channels [21].

3 Processing Structure

The channel representation, introduced in the previous section, was argued to be advantageous compared to e.g. scalar representations of certain properties. The discussion was, however, based on philosophical arguments rather than hard facts. In this section we will try to advocate the channel representation from a computational point of view.

Let the object under consideration be a vector \mathbf{v} , an element of a vector space $V = \mathbb{R}^n$. Furthermore, let the components of this vector relative to an orthonormal basis be the values of some set of channels. Hence, \mathbf{v} itself is a representation of a property value of

some other object. As a consequence, the interpretation of \mathbf{v} does not depend on $|\mathbf{v}|$, only its direction in V will matter. The goal is now to define a map from \mathbf{v} to channel values c_j , and we will begin by considering the simplest possible case, a linear map. However, a linear map will not meet the requirements set in Section 2 since the value range of a linear map is the entire \mathbb{R} , not zero to one. A linear expression can, however, be normalized to a proper range according to the following. Define a set of normalized vectors $\{\hat{\mathbf{w}}_j \in V\}$ and let

$$c_j = \frac{1}{2}[1 + (\frac{\mathbf{v}}{|\mathbf{v}|})^T \hat{\mathbf{w}}_j] = \frac{1}{2}[1 + \hat{\mathbf{v}}^T \hat{\mathbf{w}}_j], \quad (6)$$

If α_j is the angle between \mathbf{v} and $\hat{\mathbf{w}}_j$, i.e. $\cos \alpha_j = \hat{\mathbf{v}}^T \hat{\mathbf{w}}_j$, the channel value c_j can be rewritten

$$c_j = \frac{1}{2}[1 + \cos \alpha_j] = \cos^2 \frac{\alpha_j}{2} \quad (7)$$

The channel c_j will have its maximal value when $\mathbf{v} = a\hat{\mathbf{w}}_j$ where a is a positive real number. The minimal value, on the other hand, occurs when $\mathbf{v} = -a\hat{\mathbf{w}}_j$. The subset N_j which contains those property values which gives a maximal value for channel c_j is here represented by a half-line starting at the origin and running parallel to $\hat{\mathbf{w}}_j$ in its positive direction. As dissimilarity function, d , we may use α_j and the map from d to a channel value are given by Equations (6) and (7)

As mentioned in Section 2, invariance is an important feature of most algorithms and computing schemes. What types of invariances does c_j exhibit with respect to \mathbf{v} ? The answer is simply that any transformation of \mathbf{v} which leaves α_j constant will be an invariance transformation. It should be noted that this implies that if $c_j = 1$, the only invariance transformations possible are variations of $|\mathbf{v}|$, not in the direction of \mathbf{v} . The former transformations will, however, not be caused by variations in the property values of the object represented by \mathbf{v} . If $0 < c_j < 1$, the invariance spaces of V are half-cones with their tips at the origin and centered around $\hat{\mathbf{w}}_j$. Any transformation which keeps \mathbf{v} in the same half-cone will then be an invariance transformation with respect to c_j .

Each $\hat{\mathbf{w}}_j$ will correspond to a channel c_j and to accomplish a full representation of \mathbf{v} , the vectors $\hat{\mathbf{w}}_j$ must be spread out in V in some even fashion. If, for instance, any possible value of \mathbf{v} must be represented (which probably is not the case if \mathbf{v} itself is a channel representation of some object) we can choose a set of n orthonormal basis vectors $\{\hat{\mathbf{e}}_k \in V\}$ and define $2n$ vectors $\hat{\mathbf{w}}_j$ such that for each k there is some $\hat{\mathbf{w}}_i = \hat{\mathbf{e}}_k$ and some $\hat{\mathbf{w}}_j = -\hat{\mathbf{e}}_k$. Any \mathbf{v} which is directed along the positive or negative direction of an $\hat{\mathbf{e}}_k$ will then give a maximum value for exactly one channel. In other cases, some combination of channels will have non-zero values. In either case will the channels represent the direction of \mathbf{v} unambiguously.

The normalization of each c_j with respect to $|\mathbf{v}|$, according to Equation (6) is in fact not necessary. The channels $\{c_j\}$ constitute a set which represents the direction of \mathbf{v} in V . Hence, the channel values can be multiplied by an arbitrary non-zero scalar without changing the property value. For instance, we can multiply all channels by $2|\mathbf{v}|$, resulting in

$$c_j = |\mathbf{v}| + \mathbf{v}^T \hat{\mathbf{w}}_j. \quad (8)$$

This way of computing channel values may be somewhat more practical, since in practice,

the case where $|\mathbf{v}| = 0$ sometimes can not be avoided, and the former definition, according to Equation (6), will then give an undefined value for all channels, whereas the latter definition, according to Equation (8), will give a well-defined value (zero) for all channels, even though their interpretation still is undefined.

Having fully understood the properties of a linear type of processing structure, it is evident that they are not useful unless the desired invariance spaces of \mathbf{v} are of the described type, i.e. half-cones. That a linear characteristic is not enough has since long been realized for instance by the neural network community which has introduced e.g. multiple layers of linear units which in addition use sigmoid transfer functions on their outputs in order to do something useful [12]. Here, we will instead investigate a processing structure with second order characteristic. As we will see, however, this does not mean that a linear theory has to be abandoned. Let c_j be defined as

$$c_j = \hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}}, \quad (9)$$

where \mathbf{X}_j is an $n \times n$ matrix corresponding to linear map from V to itself. Hence, c_j is a homogeneous quadratic function of \mathbf{v} . Already here, we see that \mathbf{X}_j can be restricted to symmetric matrices since only its symmetric part will contribute to c_j . Hitherto, we have considered the vector space V . Having introduced a quadratic relationship between $\hat{\mathbf{v}}$ and the channel value c_j , we will for a moment leave that vector space and instead look at its tensor product space $V \otimes V$. This space can be said to contain all square $n \times n$ matrices, e.g. \mathbf{X}_j but also $\hat{\mathbf{v}}\hat{\mathbf{v}}^T$. It is easy to show that

$$c_j = \text{trace}[\mathbf{X}_j \hat{\mathbf{v}}\hat{\mathbf{v}}^T] \quad (10)$$

and also that the right hand side of Equation (10) is suitable as a scalar product on $V \otimes V$. Hence, we can write

$$c_j = \langle \mathbf{X}_j | \hat{\mathbf{v}}\hat{\mathbf{v}}^T \rangle, \quad (11)$$

which implies that c_j still can be seen as a linear combination of two elements of a vector space, namely $V \otimes V$. The above scalar product can be used to define a norm according to

$$|\mathbf{Y}| = [\langle \mathbf{Y} | \mathbf{Y} \rangle]^{\frac{1}{2}}. \quad (12)$$

It is easy to show that given this norm, it must be the case that $|\hat{\mathbf{v}}| = 1$ implies $|\hat{\mathbf{v}}\hat{\mathbf{v}}^T| = 1$, i.e. if $\hat{\mathbf{v}}$ is normalized in V then $\hat{\mathbf{v}}\hat{\mathbf{v}}^T$ is normalized in $V \otimes V$. Hence, the invariant spaces in $V \otimes V$ of a particular c_j with respect to $\mathbf{v}\mathbf{v}^T$ are again circular half cones with their tips at the origin and centered around \mathbf{X}_j . This, however, does not imply that the invariant spaces of c_j in V with respect to \mathbf{v} have this appearance. As an example, these spaces must be symmetric with respect to change of sign since both \mathbf{v} and $-\mathbf{v}$ have the same outer product, $\mathbf{v}\mathbf{v}^T$.

The definition of a channel value according to Equation (9) does not automatically meet the requirements of Section 2 since an arbitrary matrix \mathbf{X}_j can give any value range

of c_j . Since \mathbf{X}_j is symmetric, we can write

$$\mathbf{X}_j = \sum_{k=1}^n \lambda_k \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T \quad (13)$$

where each $\hat{\mathbf{e}}_k$ is an eigenvector of \mathbf{X}_j and the corresponding eigenvalues λ_k are ordered such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-1} \geq \lambda_n$. The range of the corresponding channel c_j is then $\lambda_n \leq c_j \leq \lambda_1$. To accomplish the proper range, we can write

$$c_j = \frac{1}{\lambda_1 - \lambda_n} [\hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} - \lambda_n] \quad (14)$$

or, by multiplying this by $(\lambda_1 - \lambda_n)|\mathbf{v}|^2$,

$$c_j = \mathbf{v}^T \mathbf{X}_j \mathbf{v} - \lambda_n |\mathbf{v}|^2 \quad (15)$$

Hence, if \mathbf{X}_j is further restricted to the set of positive semi-definite matrices, c_j will be non-negative without an offset.

Having restricted the range of c_j appropriately, we will turn the attention to how the \mathbf{X}_j matrices are chosen. Of course, they must be chosen in such way that any interesting direction of \mathbf{v} is represented. In the case of second order channels, however, we can not distinguish between positive and negative directions since they result in the same value of $\mathbf{v}\mathbf{v}^T$. If the elements of \mathbf{v} are assumed to be channel values, this is not a severe problem since a complete change of signs within a channel set will not affect its interpretation. Instead, we consider any possible direction of $\mathbf{v}\mathbf{v}^T$ in $V \otimes V$. The subspace of symmetric matrices in $V \otimes V$ has dimensionality $\frac{n}{2}(n+1)$ and, hence, we can find a basis of $\frac{n}{2}(n+1)$ matrices in $V \otimes V$. As an example, if $\{\hat{\mathbf{e}}_k\}$ is an orthonormal basis in V , then

$$\{ \hat{\mathbf{e}}_k \hat{\mathbf{e}}_l^T + \hat{\mathbf{e}}_l \hat{\mathbf{e}}_k^T \} \quad (16)$$

is an orthogonal basis for symmetric matrices in $V \otimes V$ (this basis is not normalized). In the same way as for the linear unit, twice as many channels can be used in order to represent $\mathbf{v}\mathbf{v}^T$. However, that would be n too many since $\mathbf{v}\mathbf{v}^T$ will never point in the negative direction of any $\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T$. To summarize, $2 \frac{n}{2}(n+1) - n = n^2$ channels will be enough if a complete representation is needed, and we can set

$$\{\mathbf{X}_j\} = \{ \pm(\hat{\mathbf{e}}_k \hat{\mathbf{e}}_l^T + \hat{\mathbf{e}}_l \hat{\mathbf{e}}_k^T), k, l = 1, \dots, n, k \neq l \} \cup \{ \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T, k = 1, \dots, n \}. \quad (17)$$

It should be noted that

$$\mathbf{X}_j = \pm(\hat{\mathbf{e}}_k \hat{\mathbf{e}}_l^T + \hat{\mathbf{e}}_l \hat{\mathbf{e}}_k^T) \quad (18)$$

implies

$$\mathbf{X}_j = \pm \left(\frac{\hat{\mathbf{e}}_k + \hat{\mathbf{e}}_l}{\sqrt{2}} \left(\frac{\hat{\mathbf{e}}_k + \hat{\mathbf{e}}_l}{\sqrt{2}} \right)^T - \frac{\hat{\mathbf{e}}_k - \hat{\mathbf{e}}_l}{\sqrt{2}} \left(\frac{\hat{\mathbf{e}}_k - \hat{\mathbf{e}}_l}{\sqrt{2}} \right)^T \right) \quad (19)$$

i.e. if $k \neq l$ then \mathbf{X}_j has two one-dimensional eigenspaces with non-zero eigenvalue, namely

those spanned by

$$\begin{aligned} \frac{1}{\sqrt{2}}[\hat{\mathbf{e}}_k + \hat{\mathbf{e}}_l] & \quad \text{with eigenvalue } \pm 1 \\ \frac{1}{\sqrt{2}}[\hat{\mathbf{e}}_k - \hat{\mathbf{e}}_l] & \quad \text{with eigenvalue } \mp 1 \end{aligned} \tag{20}$$

If $k = l$ only the former eigenspace has non-zero eigenvalue and the corresponding eigenvalue is 1. In the former case, normalized channels values would be obtained by

$$c_j = \frac{1}{1-(-1)} [\hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} - (-1)] = \frac{1}{2} [\hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} + 1] \tag{21}$$

and in the latter case by

$$c_j = \frac{1}{1-0} [\hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} - 0] = \hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} \tag{22}$$

The entire set of n^2 channels can now be rescaled to an arbitrary magnitude. This is only an example of how to choose \mathbf{X}_j when a complete representation of the directions of $\mathbf{v}\mathbf{v}^T$ is needed. In general, we may not be interested in a representation of all directions of $\mathbf{v}\mathbf{v}^T$, and the optimal direction for each channel may be chosen in other ways than pairwise orthogonal directions. Furthermore, it may not be necessary to use an offset when \mathbf{X}_j is indefinite when only a part of all possible directions of $\mathbf{v}\mathbf{v}^T$ are considered, provided that the contributions of negative eigenvalues always are compensated for by the positive eigenvalues.

As previously mentioned, the invariant spaces of c_j with respect to \mathbf{v} in V are not half-cones. Instead they will be hyperellipsoidal cylinders which are not necessarily circular and which sometimes degenerate into a pair of crossing planes [22]. As an example, consider a matrix \mathbf{X}_j where

$$\mathbf{X}_j = \hat{\mathbf{e}}_k \hat{\mathbf{e}}_l^T + \hat{\mathbf{e}}_l \hat{\mathbf{e}}_k^T. \tag{23}$$

The expression $\hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}}$ will then assume its maximal value, 1, only when

$$\mathbf{v} = a(\hat{\mathbf{e}}_k + \hat{\mathbf{e}}_l), \quad a \neq 0 \tag{24}$$

and its minimal value, -1, only when

$$\mathbf{v} = a(\hat{\mathbf{e}}_k - \hat{\mathbf{e}}_l), \quad a \neq 0 \tag{25}$$

These two invariant spaces are lines. The expression vanishes when $\mathbf{v} \perp \mathbf{e}_k$ or $\mathbf{v} \perp \hat{\mathbf{e}}_l$, corresponding to two $n - 1$ dimensional subspaces of V . In the case where $-1 < \hat{\mathbf{v}}^T \mathbf{X}_j \hat{\mathbf{v}} < 1$, the invariant subspaces with respect to \mathbf{v} will go from the first line, to the two planes, and to the second line.

With this discussion in mind it seems natural that channels which have a second order characteristic with respect to the vector \mathbf{v} will have more interesting invariance properties than first order channels. This has been claimed also by Giles and Maxwell [8], who have investigated neural networks of higher order. Furthermore, the channels can be seen as linear units if we go from the usual vector space V to its tensor product space $V \otimes V$. These two features, non-trivial invariant spaces and a linear structure will in the following

sections prove to be of main importance when implementing the channel representation on specific problems. It should be noted that the previous discussion has been based on the assumption that \mathbf{v} contains a channel representation of some other object. If the invariances needed to represent properties of that object are of the type encompassed by the above second order channels, it means that a representation of property values can be computed in a fast and robust manner by simple computational units. This is our next subject.

4 Towards an implementation

In Section 2 the channel representation was introduced and in Section 3 it was argued that a particular processing structure should be used in order to compute values of channels. In this section these two ideas will be merged into a general approach for solving computational tasks in computer vision algorithms.

The object under consideration in Section 3 was the vector \mathbf{v} , but it was nowhere mentioned how this vector came about. In principle, \mathbf{v} may represent any type of information in a processing system. It may for instance be the input signals to the system, i.e. an image. Or it may be a channel representation of some property in an image. In the first case, the elements of \mathbf{v} with respect to some basis are the pixel values of the image. The object can then be said to be the image itself, but it is evidently so that any physical object in the scene registered by the image must result in a particular pattern of pixel values. Furthermore, if properties of such an object change, e.g. its position changes, then also the image itself changes accordingly. It should be noted that these variations can be associated with variations of the *direction* of \mathbf{v} in V . If \mathbf{v} changes its norm, this can be seen as a global intensity variation of the entire image, which should not affect the interpretation of the perceived objects. Provided that the invariant spaces of the processing structure are complex enough, this representation of objects and their properties can be brought from an implicit form, carried by the pixel values, to an explicit form given by the resulting channel values.

An example of the second case, where \mathbf{v} explicitly represents object properties, is computation of curvature. Let us assume that \mathbf{v} contains a channel representation of local orientation. Each channel will then give an optimal response for a specific orientation in a particular image region. Curvature can be defined in terms of spatial variations of the local orientation and, hence, in each image region we may define a channel representation of curvature, e.g. by defining a set of channels where each channel is optimally tuned for a specific magnitude and direction of the curvature. Each curvature channel can be made invariant to e.g. the position of the curved structure within a region as well as insensitive to other types of local orientation variations like line crossings. Having defined the characteristics of the curvature representation, a processing structure must be found which can compute the desired channel values given a representation of local orientation. With some luck, simple structures like the first or second order units described in Section 3 will do the job.

In Section 2 the function which mapped a measure of dissimilarity to a channel value was left quite unspecified. We will now look at some specific choices which will suite the principles discussed in Section 3. Assume that we consider an object property which can be represented by a single scalar, x , e.g. position. If this property were to be represented also by some set of channels, corresponding to a vector \mathbf{v} , we may choose the channel representation in such a way that $|\mathbf{v}|$ does not change when x do. Since $|\mathbf{v}|$ is immaterial to

the interpretation of a set of channels, this may seem like a somewhat arbitrary approach. However, if we at some point want to decode a channel set and obtain the value of x , the channel values have to be normalized. With the suggested approach, however, the normalization does not depend on x . A feasible choice is

$$v_j(x) = \begin{cases} \cos^2(x - \frac{\pi}{3}j) & |x - \frac{\pi}{3}j| < \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases}, \quad (26)$$

It can easily be shown that for integers α and β

$$\sum_{k=\alpha}^{\beta} |v_k(x)|^2 = \frac{3}{2}, \quad (27)$$

for all x satisfying

$$\alpha \frac{\pi}{3} + \frac{\pi}{6} < x < \beta \frac{\pi}{3} - \frac{\pi}{6}. \quad (28)$$

In fact, any even power of a truncated cos function will do, provided that the channels are evenly distributed with proper. In Appendix A, a more general statement regarding these functions is presented. This illustrates that there are infinitely many channels functions which will do even with the constant norm constraint.

5 Learning

In the previous sections, it has been suggested that the channel representation can be used as a means to provide for simple computational units, e.g. of first or second order characteristic. However, even if a map from \mathbf{v} to c_j can be described, the choice of the specific $\hat{\mathbf{w}}_j$ of a linear unit or a \mathbf{X}_j of a second order unit that will do the job is still undetermined. For the following discussion, let the vector $\hat{\mathbf{w}}_j$ of a linear unit or the matrix \mathbf{X}_j of a second order unit be referred to as the state of the unit. Depending on the specific problem, there are several possible strategies for finding appropriate states for the computational units. As an example we may find the state which minimizes the mean error between the desired channel value and the actual value, given a set of inputs. In this case the state is precomputed before the operation of the processing structure commences. It may also be the case that we prefer to compute desired channel values only for those input vectors \mathbf{v} which are presented and iteratively change the state in such a way that the difference between actual and desired channel value decreases. This is of course the usual supervised learning approach used in e.g. neural networks, where the specific parameters of a system have to be learned or adapted in order for the units to produce the desired output [12].

The example presented in the following section uses the learning approach. The reason is that we want to define the desired channel values only for those values of \mathbf{v} which actually are presented to the system, rather than for any conceivable set of \mathbf{v} . Moreover, to precompute the state may be tough from a computational point of view. In some situations, it may not even be possible to define the exact events which each channel is optimally tuned for, rather that the entire set of channels must provide a representation of the events that actually occur. Today there is quite a number of algorithms for learning, e.g. as described for neural networks [12]. These are usually based on the so called steepest descent strategy, implying that the state of each unit is adjusted in the negative direction

of the gradient of an error measure. The assumption is that if this procedure is repeated enough many times, the state will reach a minima with respect to the error measure, resulting in an accurate output value of each unit. Usually the learning procedure takes a vast amount of interactions for the same input in order for the error measure reach an acceptable level. The reason is often that the representation of information into and out from these systems are of no major importance.

Here, channels are used as a means to represent information about object properties. This is the case both at the input as well as at the output of each processing unit. According to the definition of channels in Section 2, channel values should be smooth functions with respect to object properties. There are no discontinuities which has to be taken care of neither at the input nor at the output. The only thing that matters is that the computational units have enough complex invariance spaces. As a consequence, a brute force version of the steepest descent approach has been proven useful. Instead of adjusting the state of each unit by some fraction of the gradient at each iteration, the error measure is made to vanish by adding the appropriate amount of the gradient. As an example, consider a second order channel

$$c_j = \mathbf{v}^T \mathbf{X}_j \mathbf{v}. \quad (29)$$

If the desired channel value for a specific value of \mathbf{v} is \hat{c}_j , the following error measure can be used

$$\epsilon = |\hat{c}_j - c_j|^2 = |\hat{c}_j - \mathbf{v}^T \mathbf{X}_j \mathbf{v}|^2. \quad (30)$$

This error measure is differentiated with respect to \mathbf{X}_j , resulting in

$$\frac{\partial \epsilon}{\partial \mathbf{X}_j} = 2 (\hat{c}_j - c_j) \mathbf{v} \mathbf{v}^T. \quad (31)$$

A new state, \mathbf{X}'_j is given by adding some amount of this gradient according to

$$\mathbf{X}'_j = \mathbf{X}_j + a (\hat{c}_j - c_j) \mathbf{v} \mathbf{v}^T. \quad (32)$$

The parameter a is now chosen so that ϵ vanishes, i.e.

$$\mathbf{v}^T \mathbf{X}'_j \mathbf{v} = \hat{c}_j, \quad (33)$$

which gives

$$a = (\hat{c}_j - c_j) |\mathbf{v}|^{-4}, \quad (34)$$

and finally

$$\mathbf{X}'_j = \mathbf{X}_j + (\hat{c}_j - c_j) |\mathbf{v}|^{-4} \mathbf{v} \mathbf{v}^T. \quad (35)$$

As will be proven in the following section, this learning rule will give accurate channel values only after some few iterations provided that the desired unit output can be approximated sufficiently good by a second order expression.

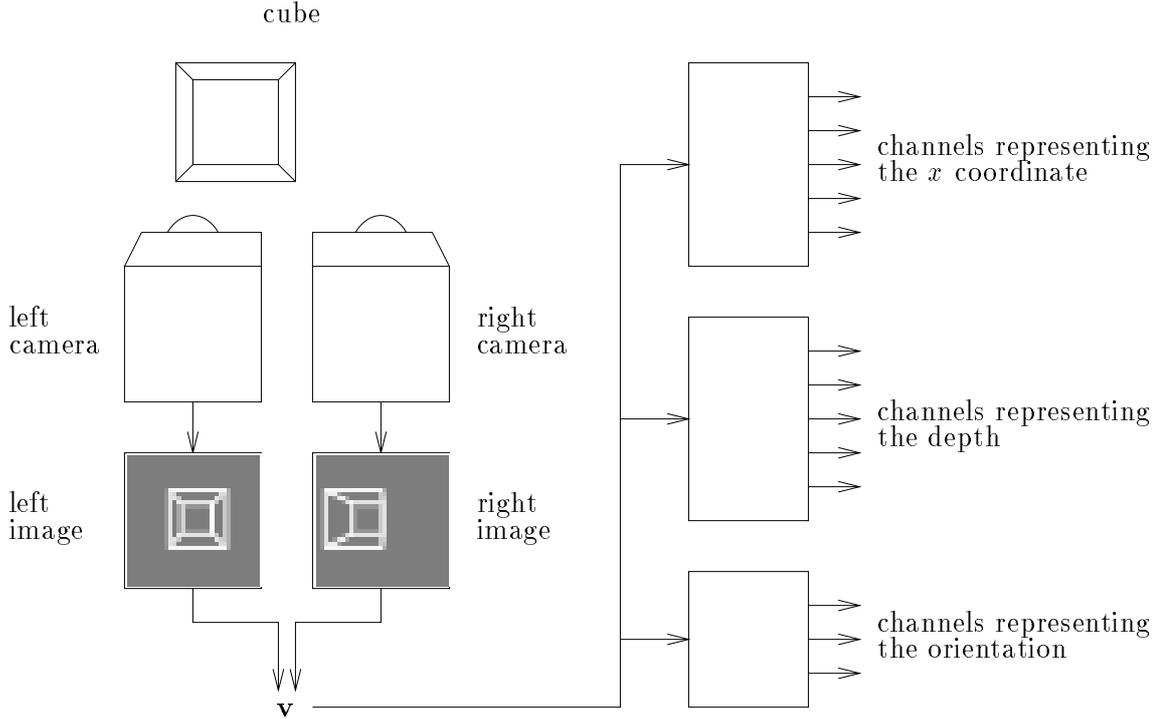


Figure 2: A 3D transparent cube in a scene registered by a stereo camera pair. The two images from the cameras constitute the vector \mathbf{v} , and are input to three sets of second order units which compute the corresponding channel values.

6 An Example

In this section, we will illustrate how the channel representation of Section 2, the second order processing structure of Section 3 and the learning rule of Section 5 can be used on a computational problem where the input data are images.

The object under consideration is a transparent cube in a 3D scene. Two cameras with no vergence are registering the cube which can have various positions and orientations. The cube moves along the horizontal axis, along the depth axis and it rotates around an axis perpendicular to the image plane. The task of the system is to describe the x position, the depth, and the orientation of the cube. The two images from the cameras constitute the vector \mathbf{v} which describes the state of the object. See Figure 2. Three sets of channels are used in order to represent the position/orientation of the cube in the scene. This means, for example, that the channel values of the x coordinate set should be invariant to variations of the depth of the cube as well as to changes in its orientation.

When the object properties changes, usually also the norm of \mathbf{v} will change as well. According to the previous discussions, however, this should not be reflected in the representation of the properties of the cube. Hence, we may normalize \mathbf{v} before using it in the processing structure to compute channel values. Each channel value is computed by the expression

$$c_j = \mathbf{v}^T \mathbf{X}_j \mathbf{v}, \quad (36)$$

i.e. no offset is used and each matrix \mathbf{X}_j is unrestricted apart from being symmetric. The desired output from each unit is normalized to the range zero to one and by learning, the actual output from each unit will become an approximation of these values, eventually resulting in channel values which are normalized. The learning is made by employing the learning rule described in Section 5, i.e. for each unit, with index j , its state \mathbf{X}_j is changed to \mathbf{X}'_j according to

$$\mathbf{X}'_j = \mathbf{X}_j + (\hat{c}_j - c_j) |\mathbf{v}|^{-4} \mathbf{v}\mathbf{v}^T, \quad (37)$$

where \hat{c}_j is the desired channel value, c_j is the actual channel value given by Equation (36), and \mathbf{v} is the input vector or image data.

Each image is 32×32 pixels, and presents an anti-aliased projection of the cube onto the corresponding image plane. Each side of the cube is six units of length (UL), the size of each image is three UL in square, the focal length is 1.5 UL and the cameras are seven UL apart. See Figure 3. The channel outputs are made to approximate desired values by running a training sequence of stereo images. In this sequence, the x coordinate varies from -3 to +3 UL in steps of 0.5 UL, with the origin in the middle of the cameras, the depth of the centre of the cube varies between 20 to 25 UL in steps of 0.5 UL and the cube rotates a quarter of a turn in eight steps. This results in a sequence of $13 \times 11 \times 8 = 1144$ instances of the vector \mathbf{v} , each of dimensionality $32 \times 32 \times 2 = 2048$. Such a high dimensionality proves impractical as well as unnecessary, since the correlation matrix \mathbf{C} , defined as

$$\mathbf{C} = E[\mathbf{v}\mathbf{v}^T], \quad (38)$$

has only a few eigenvalues which are relatively large. Using only the eigenspaces of \mathbf{C} with corresponding eigenvalues of 10% of the largest eigenvalue or more, each \mathbf{v} can be reduced to a dimensionality of less than 100, a trick which significantly reduces the computational efforts of this implementation.

There are five channels for the representation of x coordinate and depth, respectively, and three channels for representation of the orientation. Each such set of channels represents an interval of the corresponding object property using the channel function described in Section 4. The five channels of the x coordinate set give maximum response at -4, -2, 0, 2 and 4 respectively. The five channels of the depth set give maximum response at $\frac{115}{6}$, $\frac{125}{6}$, $\frac{135}{6}$, $\frac{145}{6}$ and $\frac{155}{6}$, respectively. The three channels of the orientation set give maximum response at 0, $\frac{\pi}{6}$ and $\frac{\pi}{3}$, respectively.

For each instance of \mathbf{v} in the training sequence, there is a corresponding value for the x coordinate, the depth and the orientation of the cube, each of which is represented by a set of channels and their values. Each set can be trained separately or in parallel to the others. Having presented the training set, and for each instance of \mathbf{v} adjusted the state of each channel, the result can be checked by running a test sequence. The test sequence is made by letting all the three properties of the cube change simultaneously, rather than in combination. The cube moves 100 steps from x coordinate -3 to +3, from depth 20 to 25, and it rotates a quarter of a turn. In Figure 4, the resulting channels are presented after ten presentations of the training sequence. Since the resulting channels only approximate the desired behaviour, there is no unambiguous interpretation of their values. Here, we have chosen to make a linear combination between the channel values of a set and complex numbers of unit magnitude which are evenly distributed on the unit circle in

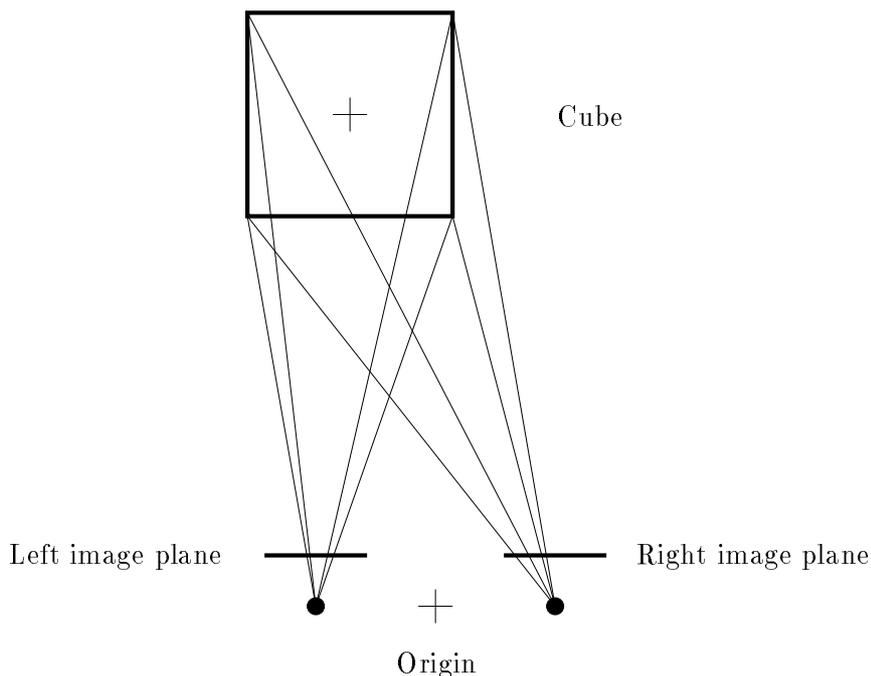


Figure 3: The scene seen from above. The two cameras have no vergence.

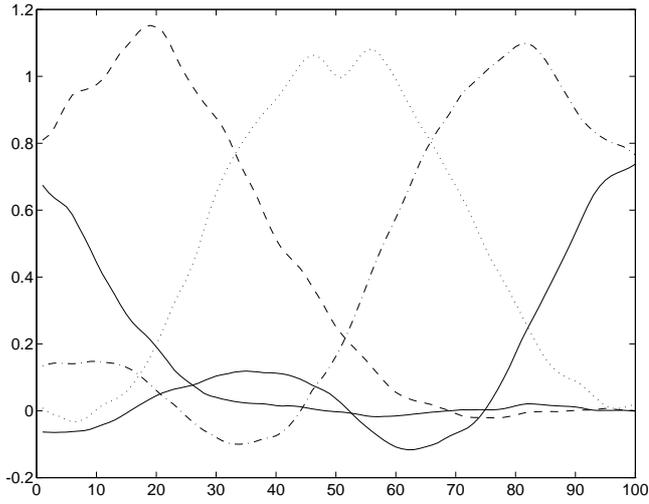
C. The argument of this linear combination can be used as a scalar representation of the corresponding object property. An interpretation of the three channel sets, according to this scheme, along with the correct values are presented in Figure 5. It should be noted that there is a very good agreement between the estimated and the correct values.

The processing structure was also tested for robustness with respect to image noise. The previous test sequence was corrupted with noise corresponding to 10 dB and 6 dB SNR and presented to the processing units. The resulting channel values are shown in Figure 6. As seen, the performance decreases but it do so with some grace.

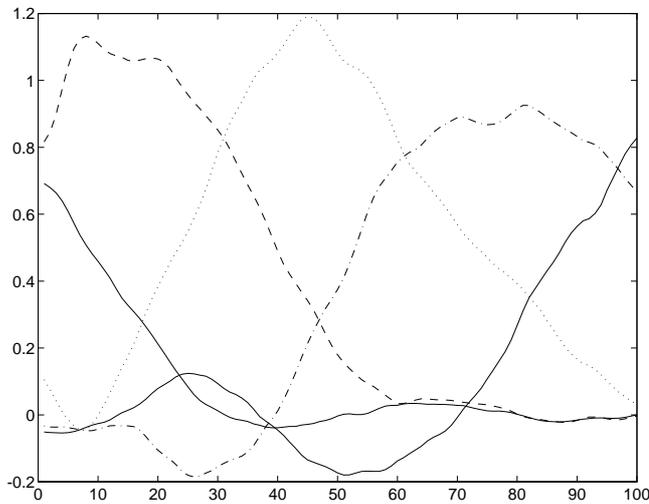
7 Summary and Discussion

In this paper it has been argued that a channel representation of object properties, inspired by biological neurons, has some advantages compared to the normal scalar representation. In short these are

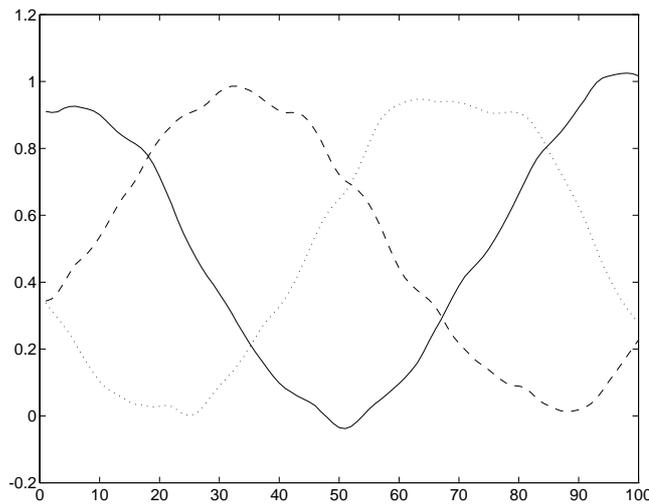
- The representation is continuous. The value of each channel should be a smooth function of the object properties which it represents. An example is representation of 2D orientation which must be discontinuous if represented by a real scalar, but can be brought to a continuous representation by using channels. The aspect of continuity is specially important if the descriptors are to be averaged over a region or over scale.
- Flexibility with respect to the topology of object properties. An example is representation of orientation in 3D. We may either use a scalar representation such as



The five channels representing the x coordinate of the cube

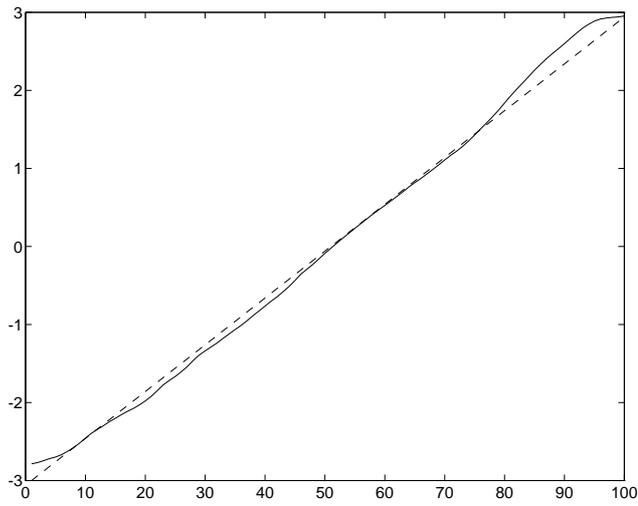


The five channels representing the depth of the cube

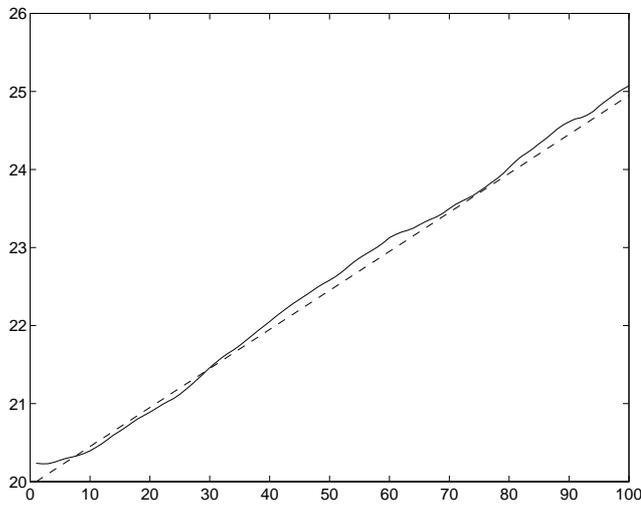


The three channels representing the orientation of the cube

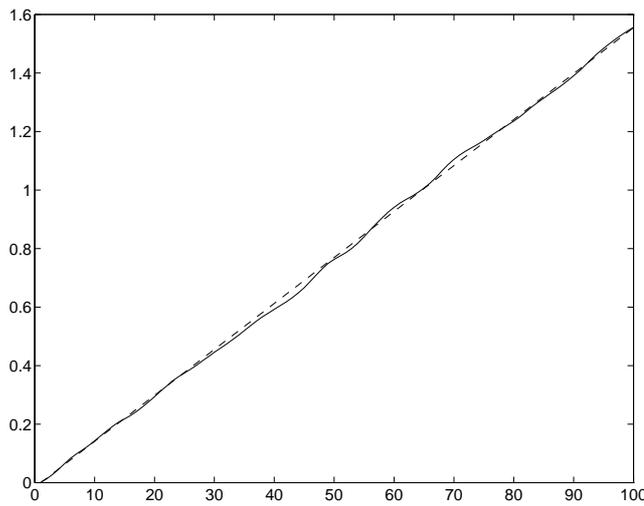
Figure 4: The channel values of the three sets when the test sequence is input to the system. This sequence shows the cube making a linear translation both along the x and the depth axes as well as a rotation a quarter of a turn around the depth axis.



The x coordinate



The depth



The orientation

Figure 5: A scalar interpretation of the channel values according to the text. The solid curves are the estimated values, the dashed curves are the correct values.

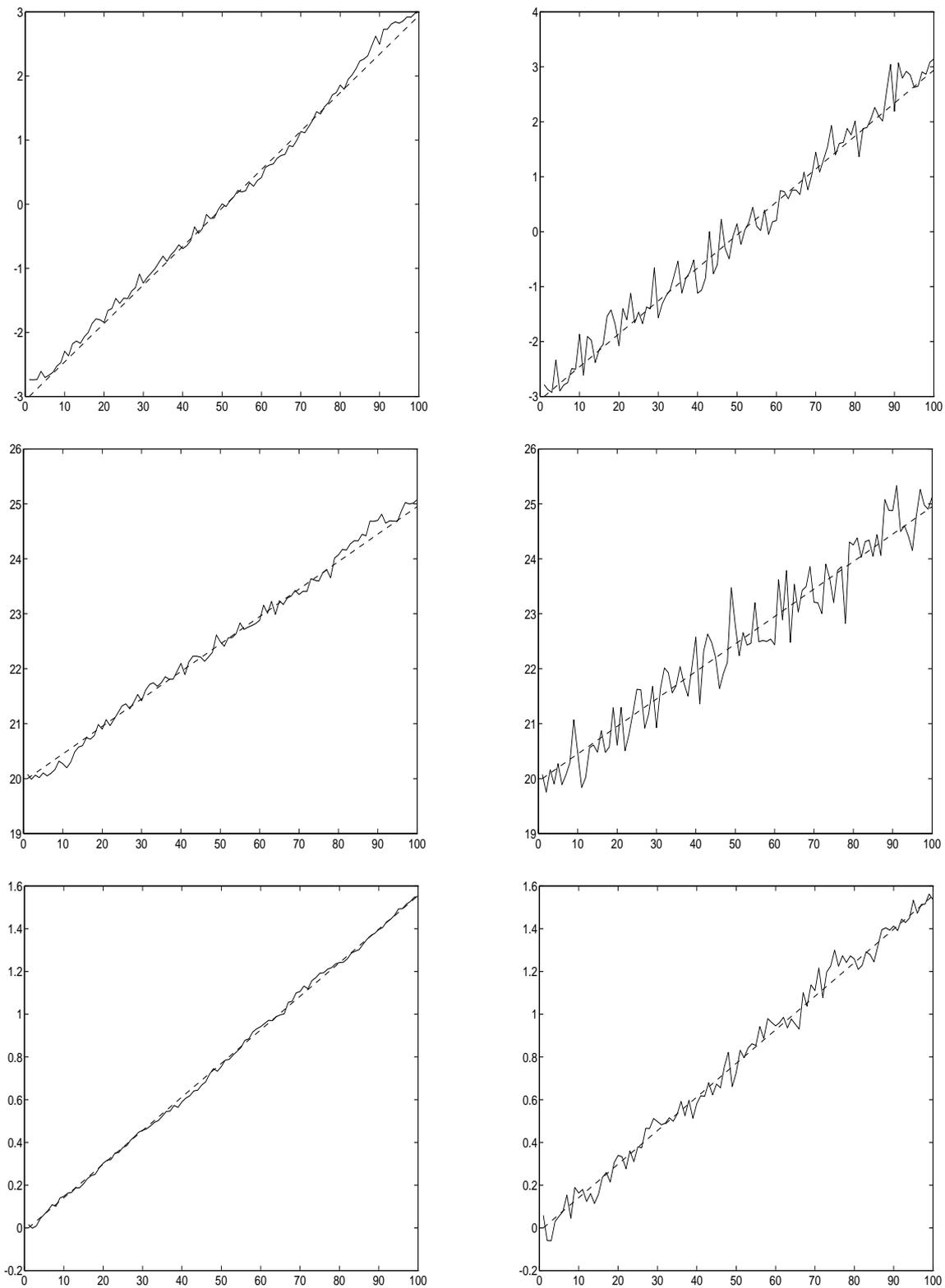


Figure 6: Same plots as in Figure 5, but here is the test sequence corrupted with 10 dB SNR noise, left, and 6 dB SNR noise, right. Top row: the x coordinate. Middle row: the depth. Bottom row: the orientation.

angular coordinates, and find that it is both discontinuous and ambiguous for the case when the azimuth angle is zero, or we may use a channel representation consisting of channels representing evenly distributed directions in 3D, as suggested by Knutsson in [16] for a tensor representation of 3D orientation. Another example, previously mentioned, is a compound representation of 2D spatial frequency and orientation.

- Computational advantages. It has been suggested, and exemplified, that the channel representation implies that very simple computational units, of first or second order characteristic, can be used for generating new channels with higher order of invariances.
- Fast and simple learning rules can be employed. As demonstrated, only a few iterations for each input pattern are necessary for the channels to produce sufficiently accurate outputs.

Of these, maybe the last two are the most striking. In the example of Section 6 an artificially generated sequence of stereo images showing a cube was used. The system was able to find states for each of its second order units such that the resulting channels represent some aspects of the cube. It should be noted that the resulting system is by no means claimed to be a "cube detector", or something which can represent these aspects for other types of objects. It should rather be seen as a demonstration of how one channel representation of object properties, the implicit representation in terms of the images, can be brought to an explicit form using very simple computations. It is by no means trivial to describe how the value of each pixel in the images depend on x coordinate, the depth, and the orientation of the cube. If these properties are to be estimated by conventional means, a 3D model of the entire scene, including the cameras, has to be established. Given this model we would be forced to measure distances between the positions of things like lines, corners, crossings, etc, in the image. Hence, these events must be detected in each image which may prove to be quite a complex task specially if the images contains noise. If, however, we choose to represent these properties using channels, it has been demonstrated that the value of each channel can be computed in a simple manner.

In this paper we have analysed a first and a second order processing structure. It was argued that a second order structure has more complex invariance spaces than a first order structure. In Section 6 it was demonstrated that a second order structure was sufficient for representation of some simple object properties. In general, however, second order structures may not be enough. It seems evident that the higher order we use, the more complex will the corresponding invariance spaces be. However, we may also take the position that complex invariance spaces are related to abstraction. For example, if we consider two signals e.g. images, both describing different objects with properties which are to be represented. Let us assume that the first object is such that a second order processing structure is sufficient for a channel representation of the corresponding properties, and assume that a fourth order processing structure must be used for the second object. This can be taken as an indication that the properties of the second object are more abstract than the properties of the first object. Furthermore, a fourth order processing structure can be divided into two steps of second order structures. This implies that concatenation of simple structures (of least second order) can be used in order to build processing structures of higher order, thereby obtaining sufficiently complex invariance spaces. This approach will fit a hierarchical processing scheme, where each level corresponds to a processing

structure of, say, first or second order units. The invariance space of each level with respect to its input is quite simple, but the invariance space of a high level with respect to the input at the lowest level can be made immensely complex.

The channel representation implies, in general, that the processing units must learn their specific parameters for the generation of a channel value. To become a successful representation, therefore, it must be established what responses are appropriate from the units. In the example of Section 6, the desired channel values were precomputed and made available to each unit by supervised learning. In a more general approach, supervised learning can not be employed since we can not a priori define, for complex image scenes, what object properties and what values thereof are of interest to represent. This is also the case for the invariances which each unit must exhibit. However, regardless of how this is done, the previous presentation has demonstrated that the usage of the channel representation implies that fast and simple learning rules can be employed.

A Constant norm channel functions

In Section 4 a particular channel function was presented in Equation (26). This function was claimed to be useful since a set of channels employing this function exhibits a constant square sum. This implies that the vector which has the corresponding channel values as its elements has a constant norm with respect to the object property under consideration. It was also mentioned that the presented function was not the only one of its kind. In this appendix, a necessary and sufficient condition is developed for any function which has the above property of constant norm.

First, some preliminaries. Let $\text{III}(x)$ denote an infinite sum of impulse functions according to

$$\text{III}(x) = \sum_{k=-\infty}^{\infty} \delta(x - k). \quad (39)$$

The Fourier transform of $\text{III}(x)$ is $\text{III}(\frac{u}{2\pi})$, i.e.

$$\int_{-\infty}^{\infty} \text{III}(x) e^{-iux} dx = \text{III}\left(\frac{x}{2\pi}\right). \quad (40)$$

This gives

$$\begin{aligned} \int_{-\infty}^{\infty} \text{III}(x) e^{-iux} dx &= \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(x - k) e^{-iux} dx = \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(x - k) e^{-iux} dx = \\ &= \sum_{k=-\infty}^{\infty} e^{-iuk} = \text{III}\left(\frac{u}{2\pi}\right), \end{aligned} \quad (41)$$

and, hence,

$$\sum_{k=-\infty}^{\infty} e^{iuk} = \text{III}\left(\frac{u}{2\pi}\right). \quad (42)$$

Let f be a function of one variable with a corresponding Fourier transform F , i.e.

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(u) e^{iux} du. \quad (43)$$

The square of the magnitude of f is then given by

$$e(x) = |f(x)|^2 = f^*(x)f(x) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^*(u)F(v) e^{i(v-u)x} du dv. \quad (44)$$

Let us assume that we have infinitely many channels, each of which employs f as their output function with respect to the parameter $x - ak$, i.e.

$$c_k = f(x - ak), \quad k = -\infty, \dots, \infty, \quad (45)$$

Let g denote the square sum of the channel values. It must then be the case that

$$\begin{aligned}
g(x) &= \sum_{k=-\infty}^{\infty} |c_k(x)|^2 = \sum_{k=-\infty}^{\infty} e(x - ak) = \\
&= \frac{1}{4\pi^2} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^*(u)F(v) e^{i(v-u)(x-ak)} du dv = \\
&= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^*(u)F(v) \left[\sum_{k=-\infty}^{\infty} e^{i(u-v)ak} \right] e^{i(v-u)x} du dv = \\
&= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^*(u)F(v) \text{III}\left(\frac{a(u-v)}{2\pi}\right) e^{i(v-u)x} du dv \tag{46}
\end{aligned}$$

By the substitution $w = v - u$, we get

$$\begin{aligned}
g(x) &= \frac{-1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F^*(v-w)F(v) \text{III}\left(\frac{-aw}{2\pi}\right) e^{iwx} dw dv = \\
&= \frac{-1}{4\pi^2} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} F^*(v-w)F(v)dv \right] \text{III}\left(\frac{aw}{2\pi}\right) e^{iwx} dw. \tag{47}
\end{aligned}$$

Hence, the Fourier transform of g is given by the expression

$$G(w) = \frac{-1}{2\pi} \left[\int_{-\infty}^{\infty} F^*(v-w)F(v)dv \right] \text{III}\left(\frac{aw}{2\pi}\right). \tag{48}$$

The integral in the bracket is usually referred to as the auto-correlation of F , denoted $F \star F$, see e.g. [3]. Hence,

$$G(w) = \frac{-1}{2\pi} (F \star F)(w) \cdot \text{III}\left(\frac{aw}{2\pi}\right). \tag{49}$$

We seek functions, $f(x)$, such that $g(x)$ is constant with respect to x , i.e. $G(w) = C\delta(w)$ for some constant C . This implies

$$(F \star F)(w) \cdot \text{III}\left(\frac{aw}{2\pi}\right) = C\delta(w), \tag{50}$$

or

$$(F \star F)(w) = 0 \quad \text{whenever } \frac{aw}{2\pi} \text{ is a non-zero integer.} \tag{51}$$

This derivation proves that g is a constant function with respect to x if and only if the condition of Equation (51) holds.

It should be noted that the auto-correlation is a convolution according to

$$(F \star F)(w) = \{F(u) * F^*(-u)\}(w), \quad (52)$$

and if f is a real and even function this implies that

$$(F \star F)(w) = \{F(u) * F(u)\}(w) = 2\pi\mathcal{F}\{f \cdot f\}. \quad (53)$$

Hence, in this case, f will be a suitable function if and only if the Fourier transform of $[f(x)]^2$ validates the condition of Equation (51).

The function suggested in Equation (26) squared is

$$[f(x)]^2 = \begin{cases} \cos^4(x) & |x| < \frac{\pi}{2} \\ 0 & \text{otherwise,} \end{cases} \quad (54)$$

which is the same as to say

$$[f(x)]^2 = \left[\frac{3}{8} + \frac{1}{2} \cos 2x + \frac{1}{8} \cos 4x\right] \cdot \text{rect}\left(\frac{x}{\pi}\right). \quad (55)$$

The Fourier transform of $[f(x)]^2$ is

$$\frac{\pi}{16} \left[6 \text{sinc}\left(\frac{u}{2}\right) + 4 \text{sinc}\left(\frac{u+2}{2}\right) + 4 \text{sinc}\left(\frac{u-2}{2}\right) + \text{sinc}\left(\frac{u+4}{2}\right) + \text{sinc}\left(\frac{u-4}{2}\right)\right]. \quad (56)$$

This function vanishes whenever $\frac{u}{6}$ is a non-zero integer, implying that if the channels use this function and are $\frac{\pi}{3}$ apart along the x axis a constant norm is obtained. This is the statement made in Section 4.

Let f be a function such that its corresponding Fourier transform F is zero outside an open interval centered around the origin and of width d . The auto-correlation function $F \star F$ must then be zero outside an open interval of width $2d$. This means that if $a \leq \frac{\pi}{d}$, where a is the distance between each channel, f will validate the constant norm criteria. Conversely, if a is defined a priori, we may choose a function f such that F is zero outside an interval of width d , where $d \leq \frac{\pi}{a}$. In practice, we may even choose functions f where $F \star F$ is sufficiently close to zero at the critical points, e.g. Gaussians with large variances.

Constant norm of derivative Having derived a criteria for constant norm of the channels values, it is easy to derive a criteria also for constant norm of change. In some applications it may be desirable that the change of channel values has a norm such that if we make one and the same small variation dx in the property value x , the norm is constant with respect to x . This is also referred to as *uniform stretch*, see [18]. Hence, we form a new function h as

$$h(x) = \sum_{k=-\infty}^{\infty} \left| \frac{d}{dx} c_k(x) \right|^2. \quad (57)$$

Following the same steps as above, it is easy to prove that a necessary and sufficient condition for h to be constant with respect to x is

$$(uF \star uF)(w) = 0 \quad \text{whenever } \frac{aw}{2\pi} \text{ is a non-zero integer.} \quad (58)$$

Again assuming that f is real and even, the autocorrelation of uF is given by the Fourier transform of $\frac{d}{dx}f(x)$ in square (disregarding the factor 2π).

Using the function f suggested in Equation (26), we see that

$$\left[\frac{d}{dx}f(x)\right]^2 = \frac{1}{2}(1 - \cos 4x) \cdot \text{rect}\left(\frac{x}{\pi}\right) \quad (59)$$

The Fourier transform of the right hand side of Equation (59) is

$$\frac{\pi}{2}\left[2 \text{sinc}\left(\frac{u}{2}\right) - \text{sinc}\left(\frac{u+4}{2}\right) - \text{sinc}\left(\frac{u-4}{2}\right)\right]. \quad (60)$$

This function vanished whenever $\frac{u}{6}$ is a non-zero integer, implying that if the channels are $\frac{\pi}{3}$ apart along the x -axis, the derivative of the channel values will have constant norm. Hence, the suggested function and distance between the channels validate the constant norm criteria as well as the constant norm of derivative criteria.

References

- [1] D. H. Ballard. *Vision, Brain, and Cooperative Computation*, chapter Cortical Connections and Parallel Processing: Structure and Function. MIT Press, 1987. M. A. Arbib and A. R. Hanson, Eds.
- [2] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.
- [3] R. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 2nd edition, 1986.
- [4] D. Cyganski and J. A. Orr. Object identification and orientation determination from point set tensors. In *Proc. of the Seventh Int. Conf. on Pattern Recognition*, pages 250–253, Montreal, Canada, 1984.
- [5] D. Cyganski and J. A. Orr. Object identification and orientation determination in 3-space with no point correspondence information. In *Proc. of IEEE Conf. ASSP*, San Diego, California, 1984.
- [6] D. Cyganski and J. A. Orr. Applications of tensor theory to object recognition and orientation determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(6):662–673, November 1985.
- [7] R. L. DeValois, D. G. Albrecht, and L. G. Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:549–559, 1982.
- [8] G.L. Giles and T. Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, 26(23):4972–4978, 1987.
- [9] G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–178, 1978.
- [10] G. H. Granlund. Magnitude representation of features in image analysis. In *The 6th Scandinavian Conference on Image Analysis*, pages 212–219, Oulu, Finland, June 1989.

- [11] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *Int. Journal of Computer Vision*, 7(2):95–117, Januari 1992.
- [12] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
- [13] W. Hoff and N. Ahuja. Depth from stereo. In *Proc. of the fourth Scandinavian Conf. on Image Analysis*, 1985.
- [14] B. K. P. Horn. *Robot vision*. The MIT Press, 1986.
- [15] D. H. Hubel. *Eye, Brain and Vision*, volume 22 of *Scientific American Library*. W. H. Freeman and Company, 1988.
- [16] H. Knutsson. Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis*, pages 244–251, Oulu, Finland, June 1989. Report LiTH-ISY-I-1019, Computer Vision Laboratory, Linköping University, Sweden, 1989.
- [17] Hans Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Linköping University, Sweden, 1982. Diss. No. 88.
- [18] Hans Knutsson. Producing a continuous and distance preserving 5-D vector representation of 3-D orientation. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management - CAPAIDM*, pages 175–182, Miami Beach, Florida, November 1985. IEEE. Report LiTH-ISY-I-0843, Linköping University, Sweden, 1986.
- [19] Y. Lamdan, J. T. Schwartz, and J. Wolfson. Object recognition by affine invariant matching. In *IEEE Computer Vision and Pattern Recognition Conf.*, jun 1988.
- [20] M. W. Levine and J. M. Shefner. *Fundamentals of sensation and perception*. Addison-Wesley, 1981.
- [21] B. Maclennan. Gabor representations of spatiotemporal visual images. Technical Report CS-91-144, Computer Science Department, University of Tennessee, September 1981.
- [22] Nils J. Nilsson. *Learning Machines*. McGraw-Hill, 1965.
- [23] D. A. Pollen and S. F. Ronner. Visual cortical neurons as localized spatial frequency filters. *IEEE Trans. on Syst. Man Cybern.*, 13(5):907–915, 1983.
- [24] T. D. Sanger. Stereo disparity computation using gabor filters. *Biological cybernetics*, 59:405–418, 1988.