Vision Research xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Vision Research

journal homepage: www.elsevier.com/locate/visres

Efficient coding and multiple motions

Erhardt Barth [a,*], Michael Dorr [a], Eleonora Vig [a], Laura Pomarjanschi [b], Cicero Mota [a,1]

[a] Institute for Neuro- and Bioinformatics, University of Lübeck, Ratzeburger Allee 160, D-23538 Lübeck, Germany
[b] Institute for Neuro- and Bioinformatics and Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Ratzeburger Allee 160, D-23538 Lübeck, Germany

ARTICLE INFO

Article history:
Received 19 October 2009
Received in revised form 8 February 2010
Available online xxxx

Keywords:
Efficient coding
Intrinsic dimension
Multiple motions
Visual subspaces
Coherent motion
Motion layers
Mid-level vision
Structure tensor
Aperture problem
Curvature
Motion stimuli
Eye movements

ABSTRACT

Based on the principle of efficient coding, we present a theoretical framework for how to categorize the basic types of changes that can occur in a spatio-temporal signal. First, theoretical results for the problem of estimating multiple transparent motions are reviewed. Then, confidence measures for the presence of multiple motions are used to derive a basic alphabet of local signal variation that includes motion layers. To better understand and visualize this alphabet, a representation of motions in the projective plane is used. A further, practical contribution is an interactive tool that allows generating multiple motion patterns and displaying them in various apertures. In our framework, we can explain some well-known results on coherent motion and a few more complex perceptual phenomena such as the 2D–1D entrainment effect, but the focus of this paper is on the methods. Our working hypothesis is that efficient representations can be obtained by suppressing all the redundancies that arise if the visual input does not change in a particular direction, or a set of directions. Finally, we assume that human eye movements will tend to avoid the redundant parts of the visual input and report results where our framework has been used to obtain very good predictions of eye movements made on overlaid natural videos.

1. Introduction

Motion selectivity is a key feature of visual processing and has been studied extensively. However, the motion patterns that occur in natural scenes are more complex than the state-of-the art motion models. This has been acknowledged in the computer vision literature; nevertheless, only a few selected problems related to more complex motion patterns have been solved (see Jähne, Mester, Barth, & Scharr (2007) for a review).

A particular case of more complex motion patterns is that of multiple overlaid motions that occur in natural environments due to transparencies, reflections, and occlusions. We will here consider the transparent superposition of motions. Our results on multiple motions have been presented in a number of technical publications and the state of the art is presented there (Aach, Mota, Stuke, Mühlich, & Barth, 2006; Barth, Stuke, Aach, & Mota, 2003; Mota, Stuke, & Barth, 2001). Here we summarize the results and apply them in the context of visual processing and efficient coding (Field, 1987; Olshausen & Field, 1996; Zetzsche, Barth, & Wegmann, 1993).

The problem of motion estimation is always linked to the problem of motion detection. This is because the assumptions under which the motion parameters can be correctly estimated are rarely fulfilled in real dynamic scenes. Therefore, a correct decision on what local or global motion model to use is often more important and difficult than the estimation of the motion parameters (Bergen, Burt, Hingorani, & Peleg, 1992, 1993). This issue relates to the need of having a basic categorization of spatio-temporal visual patterns as part of a "visual alphabet" of low- and mid-level vision (Adelson & Bergen, 1991). As an early and simple example, one might consider the barber-pole illusion: the perceived motion of lines is determined by the motion of terminators because there the confidence for a valid motion model is higher. This perceptual phenomenon is usually explained by invoking the aperture problem. In this paper we will extend such reasoning to multiple motion layers and higher-order "aperture problems".

As we shall see, the strength of our approach lies in providing not only new solutions for the multiple motion parameters, but also good confidence measures for the selection of an appropriate motion model. These confidence measures are closely related to more general aspects of multidimensional signal processing and efficient coding as presented, for example, in Barth and Watson (2000) and Zetzsche and Barth (1990). Some of these aspects will be briefly reviewed in the section on motion types.

* Corresponding author.
E-mail addresses: barth@inb.uni-luebeck.de (E. Barth), dorr@inb.uni-luebeck.de (M. Dorr), vig@inb.uni-luebeck.de (E. Vig), pomarjanschi@inb.uni-luebeck.de (L. Pomarjanschi), mota@ufam.edu.br (C. Mota).
[1] Present address: Departamento de Matematica, Universidade Federal do Amazonas, Manaus, Brazil.

0042-6989/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.visres.2010.08.011

Multiple transparent motions have often been used to probe the visual system – see Braddick and Qian (2001) for an overview. As an early result, it was reported that when the number of moving patterns is increased beyond two, subjects are no longer able to perceive all the patterns simultaneously (Mulligan, 1992, 1993). Later experiments showed that up to three motions can be detected (Andrews & Schluppeck, 2000) if their directional separation is sufficiently large and that two motions are hard to detect if the separation is small (Mota, Dorr, Stuke, & Barth, 2004).

Our key results will appear in a table (Table 2), where we will give a complete description of the different ways in which a multidimensional signal can be constant and thus contain redundancy. By that we provide a theoretical framework for dealing with multidimensional signal variation that can be applied, for example, to multiple motions but might also open a new perspective on the issue of efficient visual coding.

To better visualize and synthesize complex motion patterns, we use the projective-plane representation of motion and offer an interactive graphical tool for multiple-motion synthesis that can be used to illustrate, for example, the different types of overlaid motions in Table 2.

Finally, we present an experiment in which we have recorded eye movements of subjects who were viewing overlaid movies, and we show how our framework can be used to obtain very good predictions of the eye movements.

The paper is structured as follows. For didactical purpose, we first treat the case of multiple motions of one-dimensional spatial objects (multiple motions in x and t) and then expand this to the more relevant case of (x,y,t) motions. In the section on motion types, we generalize our results such as to deal with different combinations of multiple motions, and define a complete scheme of signal classification based on the degrees of freedom that a signal is using. We then present our approach to multiple-motion synthesis and show a demonstration based on the interactive tool. Next, we describe an experiment in which we investigated how eye movements are influenced by overlays and how well they can be predicted. Here, we will provide step-by-step directions how our framework can be used in practice without references to the details of the mathematical background, which are given in an appendix. Some readers might find it useful to start with our interactive tool described in Appendix B and available at http://www.ebarth.de/demos/ppmotion and use it as a companion throughout the paper.

2. Multiple motions in (x,t)

For didactical purpose, we first introduce the concept of multiple motions for the case of only one spatial dimension x. Suppose we are dealing with a one-dimensional pattern of light intensity $f(x)$ that translates in time with speed u. We observe the signal

$$f(x,t) = g(x - tu) \tag{1}$$

To estimate the motion parameter u, we can use the well-known constraint

$$uf_x + f_t = 0 \tag{2}$$

that formalizes the observation that image intensities remain constant in the direction of motion (see left panel of Fig. 1). Indices denote partial derivatives. Note that the above constraint is obtained by applying to the left and right side of Eq. (1) an operation (the derivative along direction $(u,1)$, i.e., along the motion vector in homogeneous coordinates) that will cause the right part, i.e., the unknown g, to vanish. Since the constraint is linear, we can use linear regression methods to estimate the parameter u based on the partial derivatives (estimated at a number of locations for which the constraint in Eq. (2) is supposed to hold) of the known function f. As a consequence, the vector $(u,1)$ will be the eigenvector to the smallest eigenvalue of the structure tensor

$$J = \omega * \begin{pmatrix} f_x^2 & f_x f_t \\ f_x f_t & f_t^2 \end{pmatrix}, \tag{3}$$

which consists of products of partial derivatives that are locally averaged (pooled) by convolution (denoted by "$*$") with the kernel ω.

If exactly one eigenvalue of J is zero (the second eigenvalue will then be positive)

$$u = -\frac{J_{12}}{J_{11}} = -\frac{J_{12}}{J_{22}} = \frac{\omega * (f_x f_t)}{\omega * (f_x^2)}, \tag{4}$$

where $J_{12},\ldots$ are the components of the structure tensor. Our motion model is only appropriate if the rank of J equals one, and we will extend this observation in the section on confidence measures.

Note that the structure tensor is constructed to deal with noise by assuming that the left part of Eq. (2) is not exactly equal to zero, and then searching for the velocity u that minimizes a weighted sum of squares of $uf_x + f_t$ (standard least-squares regression). If the assumption that the term $uf_x + f_t$ is small in a local neighborhood (same as assuming a translational motion) is not true, the rank of J will not be equal to one. In the remainder of the paper we will extend these observations to more motions and more dimensions.

Now suppose that the observed intensity signal f is the additive superposition of two moving patterns (see right panel of Fig. 1):

$$f(x,t) = g_1(x - tu) + g_2(x - tv). \tag{5}$$

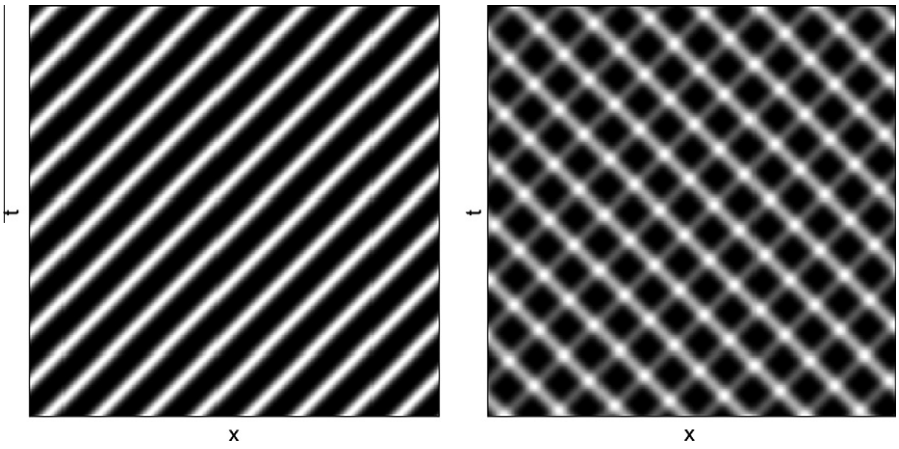

Fig. 1. One (left) and two (right) (x,t) motions.

Please cite this article in press as: Barth, E., et al. Efficient coding and multiple motions. *Vision Research* (2010), doi:10.1016/j.visres.2010.08.011

In this case, we observe f and need to estimate u and v. To obtain a constraint for one of the motions we proceed as above, i.e., we cause g_1 and g_2 to vanish. The derivative operator which causes g_1 and g_2 to vanish (concatenated derivatives in the directions of the two motions) yields the constraint

$$uvf_{xx} + (u+v)f_{xt} + f_{tt} = 0. \tag{6}$$

Note that the above constraint for two motions is nonlinear while the one for one motion was linear. We now linearize the two-motion constraint by introducing the mixed-motion parameters vector $c = (a,b,1)^T$ with components

$$\begin{aligned} a &= uv, \\ b &= u+v. \end{aligned} \tag{7}$$

Eq. (6) now reads

$$af_{xx} + bf_{xt} + f_{tt} = 0. \tag{8}$$

and, being linear, can be solved for the mixed motion parameters by regression as in the case of one motion above. The solution obtained for c will thus be the eigenvector related to the minimal eigenvalue of a generalized structure tensor

$$J_2 = \omega * \begin{pmatrix} f_{xx}^2 & f_{xx}f_{xt} & f_{xx}f_{tt}, \\ f_{xx}f_{xt} & f_{xt}^2 & f_{xt}f_{tt}, \\ f_{xx}f_{tt} & f_{xt}f_{tt} & f_{tt}^2. \end{pmatrix}. \tag{9}$$

However, we now need to recover the motion parameters u, v from the mixed-motion parameters in c. This is achieved by observing that

$$\begin{aligned} uv &= a, \\ u+v &= b. \end{aligned} \tag{10}$$

Hence, the motion parameters are simply the roots of the polynomial

$$Q_2(x) = x^2 - bx + a.$$

This concludes the explanation of our strategy for solving the problem of multiple motions in the general case. Note that the above solution is a way of linearizing the problem that becomes linear in the mixed-motion parameters.

3. Multiple motions in (x,y,t)

We now consider the realistic case of two spatial dimensions $\mathbf{x} = (x,y)$. We will restrict the presentation to multiple transparent motions that are superimposed additively. The extension of the models and results to multiplicative layers is straightforward. The more general case of both transparent and occluded motions has been treated in Barth et al. (2003). Suppose that an image sequence f is the overlaid superposition of two image layers moving with constant but different velocities $\mathbf{u}$, $\mathbf{v}$ respectively:

$$f(\mathbf{x},t) = g_1(\mathbf{x} - t\mathbf{u}) + g_2(\mathbf{x} - t\mathbf{v}). \tag{11}$$

By applying the operator $\alpha(\mathbf{u})\alpha(\mathbf{v})$ to f and expanding we obtain a constraint equation for transparent motion (Shizawa & Mase, 1990):

$$\begin{aligned} \alpha(\mathbf{u})\alpha(\mathbf{v})f = {} & (u_x v_x)f_{xx} + (u_y v_y)f_{yy} + (u_x v_y + u_y v_x)f_{xy} + (u_x \\ & + v_x)f_{xt} + (u_y + v_y)f_{yt} + f_{tt} \\ = {} & 0. \end{aligned} \tag{12}$$

We linearize the above equation by introducing the mixed motion parameters vector $\mathbf{c} = (c_{xx}, c_{yy}, c_{xy}, c_{xt}, c_{yt}, c_{tt})^T$ with components

$$\begin{aligned} &c_{xx} = u_x v_x, \quad c_{yy} = u_y v_y, \quad c_{xy} = u_x v_y + u_y v_x, \\ &c_{xt} = u_x + v_x, \quad c_{yt} = u_y + v_y, \quad c_{tt} = 1. \end{aligned} \tag{13}$$

Eq. (12) now reads

$$c_{xx}f_{xx} + c_{yy}f_{yy} + c_{xy}f_{xy} + c_{xt}f_{xt} + c_{yt}f_{yt} + c_{tt}f_{tt} = 0 \tag{14}$$

and can be solved for the mixed-motion parameters by regression as shown above. The best estimator for the mixed-parameters vector $\mathbf{c}$ is therefore the eigenvector related to the minimal eigenvalue of the generalized structure tensor

$$\begin{aligned} J_2 &= \omega * (LL^T) \\ L &= (f_{xx}, f_{yy}, f_{xy}, f_{xt}, f_{yt}, f_{tt})^T, \end{aligned} \tag{15}$$

where $\omega(x,y,t)$ is again a convolution kernel. This eigenvector can be estimated by principal component analysis or by using the minors of J_2 (see Appendix A). What remains is to recover the motion vectors $\mathbf{u}$, $\mathbf{v}$ from the mixed-motion parameters. This is achieved by reinterpreting them as complex numbers, i.e.,

$$\mathbf{u} = u_x + \mathrm{j}u_y, \quad \mathbf{v} = v_x + \mathrm{j}v_y, \tag{16}$$

and observing that

$$\begin{aligned} \mathbf{uv} &= c_{xx} - c_{yy} + \mathrm{j}c_{xy} = A_0, \\ \mathbf{u} + \mathbf{v} &= c_{xt} + \mathrm{j}c_{yt} = A_1. \end{aligned} \tag{17}$$

Hence, the motion vectors are simply the roots of the complex polynomial

$$Q_2(z) = z^2 - A_1 z + A_0.$$

A generalization for the case of an arbitrary number of N motions and the definition of the generalized structure tensor J_N for N motions is given in Mota et al. (2001) and not further elaborated here.

4. Motion types

We will not further consider the problem of estimating multiple motions, but we needed to introduce the above framework to understand how the generalized structure tensors arise. We will now use these tensors and their invariants to present a complete classification scheme for different combinations of multiple motions that describe different types of redundancies in the signal. We start with the concept of intrinsic dimension, which defines the dimension of signal-energy subspaces, and extend it to different combinations of such subspaces.

4.1. Intrinsic dimension

The intrinsic dimension (Zetzsche & Barth, 1990) describes how many of the degrees of freedom of a signal are used within a local neighborhood (see Table 1). If a signal is completely constant, e.g., a uniform wall, the intrinsic dimension is zero. Straight edges and all kinds of stationary gratings have intrinsic dimension one. Stationary or translated corners or line ends then are *i2D* and transient corners are *i3D*. We will refer to a signal with intrinsic dimension n as an *inD* signal or an *inD* feature.

Images and image sequences can be reconstructed from only those regions where the intrinsic dimension is larger than one (Mota & Barth, 2000), which means that *i0D* and *i1D* signals are redundant. Moreover, the statistics of natural scenes reveal that signals with low intrinsic dimension occur more frequently than signals with high intrinsic dimension (Zetzsche et al., 1993). This combination of geometrically proven uniqueness and statistically measured low probability of occurrence makes signals with higher intrinsic dimension an efficient representation. The resulting representation will be sparse (due to the above mentioned statistics),

Table 1
Intrinsic dimension in 3D. The parameters $a, b, c, \ldots$ define the directions in which the signal does not change.

Intrinsic dimension	Description	Constraint	Signal energy
0	Constant in all directions	$f(x,y,t) = c$	Point
1	Constant in two directions	$f(x,y,t) = g(ax + by + ct)$	Line
2	Constant in one direction	$f(x,y,t) = g(a_1x + b_1y + c_1t, a_2x + b_2y + c_2t)$	Plane
3	No constant direction	None	Volume

and will also minimize the loss of information (due to the uniqueness theorem). Indeed, recent results obtained with sparse coding and overcomplete bases indicate that *i2D* operators emerge as nonlinear filters (Labusch, Barth, & Martinetz, 2009; Olshausen, 2009).

It then seems a straightforward hypothesis that visual processing should exploit this potential efficiency, and suppress signals with lower intrinsic dimension. This simple hypothesis suffices to explain the occurrence of lateral inhibition (*i0D* signals are suppressed), end-stopping (*i1D* signals are suppressed), and motion selectivity (Barth & Watson, 2000).

It is useful to consider the Fourier transform and energy distributions of signals with different intrinsic dimensions. The problem of determining the intrinsic dimension is then equivalent to the problem of determining whether the energy of the signal is restricted to a certain subspace, e.g., a plane (within a volume). A well-known result is that the energy of a rigid-motion signal is restricted to a plane (Watson & Ahumada, 1983). In case of multiple motions we are dealing with the problem of detecting that the energy is on multiple planes, and of estimating the parameters of the planes. In general, however, different combinations of different kinds of subspaces are possible, e.g., of planes and lines if a moving dot pattern and a moving grating are overlaid.

4.2. Intrinsic dimension and motion

Image regions that are *i2D* (e.g., corners, line ends) are not only the most informative but also the only regions where motion can be estimated correctly, because *i1D* signals suffer from the aperture problem. The early experiments by Wallach (see Wuerger, Shapley, & Rubin, 1996) demonstrate that the perceived direction of motion is dominated by the motion of the *i2D* regions (the so-called terminators). This strategy has been confirmed by neurophysiological data showing that neurons in area MT of the macaque prefer *i2D* motion signals (Pack, Gartland, & Born, 2004), and that *i2D* selective end-stopped neurons in the primary visual areas seem to be involved in avoiding the aperture problem (Pack, Livingstone, Duffy, & Born, 2003) – see Born and Bradley (2005) and Rust et al. (2006) for review papers on the role of MT neurons.

4.3. Categorization of motion types

The classification according to the intrinsic dimension has the following limitation. A signal with intrinsic dimension three may have an energy distributed not in a volume but in multiple planes, or maybe in a combination of multiple planes and lines. One might refer to such cases as fractional intrinsic dimensions, where the intrinsic dimension is not two but also not really three. An interesting special case is that of multiple orientations in images (Aach et al., 2006). The more general problem of how to detect and estimate multiple orientations in multidimensional signals was solved only recently (Stuke, Barth, & Mota, 2006).

As we shall see, the major benefit of the generalized structure tensor is that it provides a natural categorization of the visual input, or any three-dimensional signal, in terms of its complexity and redundancy. The rank of the tensor J_1 is known to correspond to the intrinsic dimension, and it thus follows from (Mota & Barth, 2000) that signals with a rank of J_1 less than two are redundant. Although signal categorization by the rank of J_1 has proven useful in many technical applications, it should be noted that the concept of intrinsic dimension is more general, and that the eigenvalue analysis of J_1 is just one specific way of determining the intrinsic dimension. Nevertheless, for simplicity, it is a major focus of this work to provide a categorization of dynamic visual patterns in terms of the ranks of the tensors J_N, with $N = 1, 2, 3$.

Fortunately, although hard to derive, the results can easily be presented as shown in Table 2. We here restrict the patterns in Table 2 to overlaid motions that are defined as the additive superposition of elementary patterns that are either 1D or 2D spatial patterns. Except for the 3D case (last line in the table), these patterns move with constant velocity. 1D patterns can thus be, for example, moving gratings and straight lines, whereas 2D patterns can be, for example, moving dots or moving noise patterns. 3D patterns are, for example, dynamic noise, transient dots, or transient corners. Note that within the traditional theory of only one motion (as represented by the tensor J_1), one cannot distinguish between the motion of two 1D patterns (e.g., a plaid) and the motion of one 2D pattern (dot, corner, or noise) because the rank of J_1 is the same in the two cases. These two types of patterns differ, however, in the ranks of both J_2 and J_3.

Finally, we should note that the signal types presented in Table 2 form a complete set, i.e., the table contains all possible combinations of up to three motion layers (other combinations of ranks cannot occur). The results summarized in Table 2 therefore make a contribution to the alphabet of how to "measure stuff" (Adelson & Bergen, 1991).

4.4. Confidence measures

We now address the question of how to derive confidence measures for the different types of motions in the above table. As shown in the table, the different types of motion patterns can be classified in terms of the rank of the structure tensors. A theoretically obvious choice would be to determine the rank based on an

Table 2
Basic dynamic patterns (left column) that have intrinsic dimensions indicated in the third column and that correspond to the ranks of the generalized structure tensors for $N = 1, 2$, and 3 given in the right columns. Colors are used to illustrate the cases of generalized aperture problem (red), translation (green, indicating a proper motion model), high (blue), and low (orange) complexity. *MGs* are moving patterns that are spatially 1D, e.g., moving gratings. *MDPs* are moving patterns that are spatially 2D, e.g., moving dots. *Transient dots* stands for the class of signals that show no constant direction, the simplest example being dots that appear or disappear. The table summarizes all possible combinations of up to three moving patterns (but for the trivial possibility of adding 0D patterns to all combinations) and shows that these patterns can be distinguished in terms of the ranks of the structure tensors.

Pattern	Examples	Intr.Dim	rank J_1	rank J_2	rank J_3
0D	Uniform wall	0	0	0	0
1D	Moving grating (MG)	1	1	1	1
1D+1D	2 MGs	2	2	2	2
1D+1D+1D	3 MGs	3	3	3	3
2D	Moving dot pattern (MDP)	2	2	3	4
2D+1D	1 MDP + 1 MG	3	3	4	5
2D+1D+1D	1 MDP + 2 MGs	3	3	5	6
2D+2D	2 MDPs	3	3	5	7
2D+2D+1D	2 MDPs + 1 MG	3	3	6	8
2D+2D+2D	3 MDPs	3	3	6	9
3D	Transient dots	3	3	6	10

eigenvalue analysis of the tensor. However, confidence measures for the different ranks can also be defined based on the invariants of the structure tensors:

$$\begin{aligned} H_1 &= 1/3 \text{ trace } (J_1) = \lambda_1 + \lambda_2 + \lambda_3, \\ S_1 &= |M_{11}| + |M_{22}| + |M_{33}| = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3, \\ K_1 &= |J_1| = \lambda_1\lambda_2\lambda_3, \end{aligned} \tag{18}$$

J_1 is the structure tensor for one (x,y,t) motion and is defined in Eq. (21). The λ_i are the eigenvalues of the structure tensor, but note that we might not need the eigenvalues to determine the rank of the tensor. The rank can be determined in terms of the invariants H_1, S_1, and K_1. H_1 is the trace of the structure tensor J_1, i.e., the invariant H_1 results from the sum of the basic measurements (eigenvalues, i.e., variations in main directions). S_1 is the sum of the minors ($|M_{11}|$,…, the minors of J_1 are determinants of sub-matrices, see Appendix A). Accordingly, S_1 is a sum of pairwise products of eigenvalues. Finally, K_1 is the determinant, i.e., the product of all eigenvalues. Note that the above invariants are nonlinear combinations of linear filters.

The invariants can be used in a straightforward way. If $K_1 \neq 0$, the rank is three, so K_1 can quantify the likelihood of having a full-rank tensor (blue entries in Table 2). If $S_1 \neq 0$ the rank is at least two. As a consequence, the green entries in the table (well-defined motions) are defined by having a large S_1 and a small K_1. If $H_1 \neq 0$ the rank is at least one. Details on how to normalize the invariants when comparing them can be found in Mota et al. (2001), see also Section 7.

The reddish fields in the table (inconsistent motions) are defined by zero values of S_1 and K_1. Remember that those features in a movie where S_1 and K_1 (of J_1) are zero, are not only those features where motion cannot be determined but also those regions that are redundant (see intrinsic dimension above). This is why it would make sense for any vision system to have estimates of the invariants S_1 and K_1. In other words, a system that represents movies in terms of H_1 will suppress (not represent) all those visual features that are *i0D*, i.e., constant in all directions. S_1 will filter out from the representation *i0D* and *i1D* features, and K_1 will suppress *i0D*, *i1D*, and *i2D* features. So, when moving from *H*-like to *S*-like and *K*-like representations, the degree of sparsification will increase since the signals with three, two, and one constant directions will be suppressed. However, as shown in Table 2, the invariants H_1, S_1, and K_1 cannot differentiate between the different types of motion superpositions. For example, the linear superposition of two *i1D* signals cannot be differentiated from a single *i2D* signal, and the superposition of two *i2D* signals cannot be distinguished from an *i3D* signal.

These problems can be solved by using the generalized invariants H_2, H_3, K_2, and K_3 of J_2 and J_3, which are defined as in Eq. (18) (but for a higher number of eigenvalues). The invariants of type *S* of J_2 and J_3 are defined similarly. S_{22} and S_{32} are defined as the sum over all possible products (15 in case of J_2) of pairs of eigenvalues. The higher-order invariants are defined as sums of products of triples, quadruples, … of the eigenvalues. Accordingly, S_{23} is the sum over all 20 products of triples of eigenvalues of J_2, S_{24} the sum over all 15 products of four eigenvalues, and S_{25} the sum over all six products of five eigenvalues. Overall, the invariants are defined in terms of $\sum\prod$ structures, with *H* and *K* being the extreme cases of only $\sum$ (sum of all eigenvalues, no product) and only $\prod$ (product of all eigenvalues, no summation), respectively. Note that in case of one-dimensional signals the structure tensor J_1 is degenerated and no $\Sigma\Pi$ structures arise (we have only *H*), and in case of two-dimensional signals only the two extreme cases of *H* and *K* exist (no *S*-type invariants).

When comparing K_1 and K_2 by looking at the 4th and 5th columns of Table 2, we see than K_2 can suppress patterns that result from superpositions of patterns with low intrinsic dimension. For example, in case of a 2D + 2D transparent motion (row 9) K_1 is different from zero but K_2 can "see" this higher-order redundancy and is equal to zero (a response in K_2 would require a rank of J_2 of six). However, once we are looking at motions of more than two *i2D* patterns, we need K_3 to "see" and suppress that redundancy. Note that filtering out *i0D* signals is simple and can be done with linear systems. But as we move on to suppress increasingly complex redundancies, we therefore need increasingly complex nonlinearities. The invariants of the generalized structure tensors can tell us what kind of nonlinearities are required. The resulting operators can be understood as a sandwich structure of two linear and two nonlinear stages that can be summarized as follows: (i) linear spatio-temporal filtering (derivatives f_{xx}, …), (ii) nonlinear stage (product terms, e.g., $f_{xx}f_{yy}$ in J_2), (iii) linear spatio-temporal pooling (convolution with the kernel ω), (iv) nonlinear stage (defined by determinants or sum of minors, i.e., product terms again).

5. Implementation of derivatives and filters

This section discusses the relationship between derivatives and visual filters and thereby addresses critical aspects of the differential approach. It can be skipped by those not interested in the implementation issues.

Differential approaches are often discussed as being sensitive to noise. This is particularly relevant for our results on multiple motions since the order of differentiation increases with the number of motions. Therefore we now present two methods that are useful to overcome such problems. In our simulations in Section 7, however, we have used standard differences of Gaussians as derivatives and obtained very good prediction results with noisy natural movies.

5.1. Prefiltering

In Mota et al. (2001) solutions for multiple motions were derived by showing that the differential results still hold for any type of linear pre-filtering. This implies that the shape of the differential filter can be influenced to a large degree and thus adapted such as to improve sensitivity to noise. For the case of one motion, this property has been used in Srinivasan (1990).

5.2. Generalized derivatives

As an important consequence of the above-mentioned result, one can use fairly general filter functions instead of the derivatives. Consider the Fourier transform of the derivative kernels (*X*, *Y*, and *T* are the Fourier variables corresponding to *x*, *y*, and *t* respectively):

$$\frac{\partial^\kappa \partial^\mu \partial^\nu}{\partial x^\kappa \partial y^\mu \partial t^\nu} \Rightarrow (-i)^{\kappa\mu\nu} X^\kappa Y^\mu T^\nu. \tag{19}$$

A major weakness of calculus is that it cannot separate the symmetry of the derivative (i.e., whether $-i$ to the power, say, $\kappa\nu\mu$ in the above equation equals $-i$ or 1) from the shape of the filter (i.e., the actual value of the product $\kappa\nu\mu$) which determines the steepness of the filter function and thus the sensitivity to noise. Note, however, that the results of motion estimation do not depend on the shape of the filter function because the shape can be manipulated by pre-filtering. For example, if we had a filter X^4 as a fourth-order derivative with respect to *x*, we could use a pre-filter X^{-3} and thus end up filtering with *u* as an estimate of the derivative. The latter filter would be much less sensitive to noise since it would be much less a high-pass filter.

Please cite this article in press as: Barth, E., et al. Efficient coding and multiple motions. *Vision Research* (2010), doi:10.1016/j.visres.2010.08.011

6. Synthesis of multiple motions

Tables 1 and 2 are useful for categorizing basic properties of multidimensional signals in general and certain motion patterns in particular. In order to better exemplify the different motion patterns we now turn to the problem of motion synthesis. The study of motion patterns has profited from looking at Fourier correspondences, because then motion could be visualized as a plane in the transform domain (Watson & Ahumada, 1983). Multiple motions correspond to multiple such planes and the motion of a plaid pattern is said to be determined by the two lines in the transform domain (which correspond to the two 1D moving gratings) that define a plane. But this plane is not the only way to fit the two lines; cones would also fit them. This is one reason for proposing a new synthesis tool. Furthermore, without the tool, for more than two motions, it becomes hard to understand what overall motions would result from the superposition of different moving patterns.

We therefore use the projective plane as a means of better describing overlaid motions, and we shall see how this simplifies the analysis and synthesis of motion patterns.

If a signal f is the additive superposition of moving layers, its Fourier transform F is the superposition of Dirac planes that all pass through the origin. To represent these motions in the projective plane, we assign a point in the projective plane to a 2D moving pattern (a plane in the Fourier domain) and a line in the projective plane to a 1D moving pattern (a line in the Fourier domain). This representation is useful because a point will represent the unique motion vector (both direction and speed) in the case of 2D patterns, and a line the set of all possible motion vectors in the case of 1D patterns. The motion of a plaid pattern will thus result from the intersection of the two lines that correspond to the motions of the 1D components. A further and obvious benefit of the projective plane is that we can describe the motions in two dimensions instead of three.

The projective-plane representation is particularly useful for synthesis. The correspondence between motion patterns and the projective plane is bijective. By placing points and lines in the projective plane we can thus generate various motion patterns. The best way to learn about this possibility is to use our interactive tool. The interactive tool is described in Appendix B.

We now present two examples of how to use the interactive tool.

With two intersecting lines in the projective plane you see the coherent motion of the plaid that corresponds to the intersection. When adding a third grating, the coherent percept breaks down and three different combinations of one plaid and one grating can be perceived (Andrews & Schluppeck, 2000). This effect can easily be understood in the projective plane because the three lines intersect in three points, which correspond to the admissible plaid motions. The effect is strongest when the center of the projective plane is inside the triangle described by the three intersections. However, once you have all three lines intersecting in one point, only the one coherent motion corresponding to that point is seen. In general, the intersection points in the projective plane predict the motion percepts well. Moreover, the coherence of the perceived motion is higher for smaller triangles. Once you stop the motion of one layer, the set of admissible velocities is reduced to the single coherent motion of a plaid.

The second example is the *2D–1D entrainment effect* (Mota et al., 2004). Start with the default (circular) aperture, additive transparency, and one grating. You see the motion orthogonal to the orientation of the grating. Now place a noise pattern in the projective plane. You will see two transparent layers. The separation of the layers is better, the farther away the point is from the line. Once you place the point on the line, you see a coherent motion of the noise and the grating. However, when you move the point along the line, you will see that the motion of the noise pattern will always drag along the motion of the grating; this is the 2D–1D entrainment effect. As you change the type of superposition and the shape of the aperture you will notice that the effect is quite stable. The effect is simply explained by noting that the perceived motion is the intersection of the two layers in the projective plane (the common set of admissible velocities). However, at extremely elongated apertures, the barber-pole illusion (a motion of the grating seen along the aperture) will dominate and lead to a further interesting percept: the grating and the noise are seen as moving in different directions but no transparent layers are perceived. Again, transparency is determined by the distance between the layers in the projective plane.

7. Prediction of eye movements on overlaid movies

7.1. Introduction

We have seen in the section on confidence measures that motion estimation involves the determination of the appropriate motion model. In the simple case of one translation, *i0D* and *i1D* signals must be suppressed because motion can be estimated only for *i2D* signals. Generalizing this principle to more complex motion patterns requires more complex suppression mechanisms that, in our case, involve the invariants of the generalized structure tensors. The very same suppression mechanisms increase the efficiency of the representation, since the suppressed signals (of lower intrinsic dimension) exhibit increasingly complex regularities and by that increasingly complex redundancies.

We now make two assumptions: (i) the visual system does suppress signals with lower intrinsic dimensions (lower ranks of the structure tensors) and (ii) the resulting representations are somehow involved in the guidance of eye movements, in other words, eye movements tend to focus on less redundant features.

To test these hypotheses, we measure eye movements that subjects make on videos with overlaid motions and predict the eye movements based on different representations of the movies, using Machine Learning techniques to distinguish attended movie patches from control patches. The representations are exactly those that would arise from the suppression of increasingly complex regularities, i.e., we start from the invariants of J_1 and move on to the invariants of J_2, and for each invariant we use representations that would require an increasingly higher rank in order to be different from zero – see Table 2. More precisely, we use as representations the invariants H_1 (indicates that the rank of J_1 is at least one), S_1 (rank of J_1 is at least two), K_1 (rank of J_1 is three), H_2 (rank of J_2 is at least one), S_{22} (rank of J_2 is at least two), S_{23} (rank of J_2 is at least three), S_{24} (rank of J_2 is at least four), S_{25} (rank of J_2 is at least five), and K_2 (rank of J_2 is six). We shall see that as we move along with less redundant representations, the quality of the predictions tends to improve, and some of the improvements are highly significant.

7.2. Methods

7.2.1. Stimuli and experimental setup

For the experiment, we used 19 high-resolution (1280 by 720 pixels) overlaid movie clips of 17 s duration each. Each movie was created by superimposing pairs of two videos randomly selected from a set of 14 outdoor natural scene sequences (Dorr, Martinetz, Gegenfurtner, & Barth, 2010). Spatio-temporal frequency bands were equalized before the superposition to avoid that blending two movies with very different spatio-temporal spectral energy distribution would lead to the perceptual dominance of

Please cite this article in press as: Barth, E., et al. Efficient coding and multiple motions. *Vision Research* (2010), doi:10.1016/j.visres.2010.08.011

one component video in the blended result. To this end, movies were decomposed into an anisotropic spatio-temporal Laplacian pyramid with five spatial and five temporal levels, and blending weights were computed separately on each pyramid level as the reciprocal of its standard deviation. Samples of the resulting stimuli are shown at http://www.ebarth.de/demos/VRspecial.

The 19 resulting videos were shown in random order to ten volunteering subjects with normal or corrected to normal vision. The experimental setup consisted of an Iiyama Vision Master Pro 514, 22" display screen with an actual viewable diagonal of 20" and an SMI Hi-Speed eye tracker system running at 1250 Hz. The viewing distance was 50 cm. A nine-point array was used to calibrate the tracker. The displayed size of the videos was approximately 43° by 23° of visual angle (because the aspect ratios of the screen and the videos did not match, the videos were displayed in the "letterbox" format with black stripes at the bottom and top of the screen). The subjects were instructed to freely view the movies. After each movie clip, drift correction was performed and, halfway through each viewing session, the subject was offered a short break in order to relax. The second part of the viewing started with a full recalibration of the eye tracker.

The saccades were extracted from the collected gaze data using a velocity based procedure (Böhme, Dorr, Krause, Martinetz, & Barth, 2006). The resulting saccade data was further filtered so that samples rendered invalid because of blinks were removed from the data. About 10,000 saccades remained after filtering.

7.2.2. Feature extraction

Here we describe the computations required to determine the invariants and how we use the invariants to compute a feature vector for the subsequent classification stage.

7.2.2.1. First linear stage: Gaussian pyramids and derivatives. We have chosen the simplest way of estimating derivatives, namely to first convolve the image sequence with a Gaussian smoothing kernel, and to then compute the differences among nearby pixel values (e.g., f_x at position (x,y,t) would result as the difference between the values $f(x-1,y,t)$ and $f(x+1,y,t)$). Second order derivatives are obtained by iterated differentiation, e.g., f_{xx} is obtained by applying the above first order derivative to f_x instead of f.

In order to obtain representations with multiple spatio-temporal scales, we perform the smoothing operation with different levels of resolution. The details of the implementation and the values of the parameters are the same as in Vig, Dorr, and Barth (2009) but for the fact that we here use an anisotropic spatio-temporal Gaussian pyramid with five spatial and five temporal levels. In other words, the movie is successively blurred (with a Gaussian kernel with $\sigma = 1$ pixel) and subsampled, and the derivatives are then computed as pixel- and frame-wise differences on all the 25 scale levels.

7.2.2.2. First nonlinear stage: product terms. The nine product terms required to estimate the tensor J_1 are defined by Eq. (21), e.g., f_x^2 and $f_x f_y$. The 36 product terms in J_2 are defined by Eq. (15), i.e., all possible combinations of the components of L, e.g., f_{xx}^2, $f_{xx}f_{yy}$, and $f_{xx}f_{tt}$. Because the structure tensor is symmetric, only six and 21 product terms have to be computed in practice, respectively. Such multiplications are the key nonlinearity for the suppression of signals with lower intrinsic dimensions (Zetzsche & Barth, 1990).

7.2.2.3. Second linear stage: spatio-temporal pooling. As can be seen in Eqs. (3), (21), (15), the computation of the structure tensors involves a convolution of the nonlinear product terms with a smoothing kernel ω. Our kernel is a Gaussian with a width of $\sigma = 1$ pixel. For motion estimation, the size of ω should be adapted to the size of the spatio-temporal neighborhood for which one can assume the motion model to hold. In practice, however, the expected size is not known (but could be estimated). We have not systematically investigated the role of this parameter with respect to the prediction results.

7.2.2.4. Second nonlinear stage: invariants. The invariants of J_1 are defined by Eq. (18). We used the first part of the equations, i.e., we computed H_1 as the trace of J_1, S_1 as the sum of minors, and K_1 as the determinant. In addition, we equalized the variances of the invariants by computing H_1^6, S_1^3, and K_1^2. We did this because we wanted the invariants to differ only in terms of how they suppress different signal types and not in terms of how they weight the remaining signal types (because they comprise of products of one, two, and three eigenvalues, respectively). The invariants of J_2 are defined in Section 4.4. For a 6×6 matrix it is computationally more efficient to compute the invariants numerically by performing an eigenvalue analysis, and then using the sums and products of the resulting eigenvalues to compute the invariants. However, we used the geometric means, and not the products of the eigenvalues, in order to equalize the differences between the invariants with respect to how they vary over space and time.

Note that we compute the invariants in order to not depend on a particular choice of the coordinates, in our case (x,y,t) in horizontal, vertical and time directions. One intuitive interpretation is that we need a mechanism that determines locally the main directions of signal variance, and then we need nonlinearities that generate products and sums of products of these variances ($\sum\prod$-structures).

7.2.2.5. Final pooling stage: signal energy as feature vector. There are two reasons for not using the invariants, say $K_1(x,y,t)$, directly for making the predictions. The first reason is that both the eye tracker and the eye movements are not precise. Therefore, we have to consider a spatio-temporal neighborhood (window) around the fixations to allow for position uncertainty. The second reason is that using all pixels contained in a reasonably-sized window becomes computationally intractable. For a spatio-temporal patch of 64 by 64 pixels (about 2.5° by 2.5°), the dimensionality of the pixel space in which subsequent learning algorithms (see below) would have to operate is 4096. Hence, a representation is needed to reduce the dimensionality of the feature space.

We therefore decided to compute the local signal energy in a window around each saccade landing point (x,y) as

$$e_{s,t} = \sqrt{\frac{1}{W_s^2}\sum_{i,j=-W_s/2}^{W_s/2} I_{s,t}^2(x_s - i, y_s - j)}, \tag{20}$$

where $I_{s,t}$ is a frame of the sth spatial and tth temporal level of the saliency pyramid with I being one of the invariants $H_1,\ldots,K_2$, which had been computed before for every pixel. Because of the reduced resolution of higher spatial levels, the fixation point (x_s,y_s) on spatial level s is $(x/2^s, y/2^s)$. For the same reason, window size W_s was decreased by a factor of two per level in the spatial domain so that the spatial envelope was kept constant (W_0 was set to 64 by 64 pixels for all simulations, $W_s = W_0/2^s$). In time, one frame of a lower level corresponded to several frames on the original level, so that the time window was about half a second. Note that the dimensionality of the space in which the prediction takes place is thus independent of window size.

As a result of all the above steps, we obtained a 25-dimensional (five by five) feature vector with each component being the signal energy of a particular invariant at a particular scale, and in a window around the considered location.

7.2.3. Machine learning and classification

The locations can be of two types: attended and not attended locations. The set of attended locations consisted of the landing points of the approximately 10,000 saccades extracted from the recorded gaze data. For the non-attended class that contains image regions that were of relatively low saliency, we shuffled the movies and their corresponding scanpaths, so that the non-attended regions of a selected video were picked from the saccade landing points on a different, randomly chosen video and vice versa. This procedure might introduce some overlaps in the two classes, but removes artifacts due to the central fixation bias and assures that the negative examples are drawn from the specific distribution of human fixation locations.

In addition, the data set of all feature vectors was divided into two subsets: the training set, containing the fixations of two-thirds of all subjects (on all 19 movies), and a test set with the fixations of the remaining one-third of the subjects (also on all movies). Although we used saccade data from all movies, we made sure that a subject's scanpath on a given movie appeared only in one of the two subsets, i.e., we were predicting the behavior of new subjects.

On the first subset of data we learned how to separate the classes and on the second subset we tested how well we can predict the eye movements of unknown subjects by using the learned classifier. For learning we trained a soft-margin Support Vector Machine, with a Gaussian kernel (using the standard LIBSVM package). Twenty different training and test set realizations (random subdivisions of the data into the two subsets) were used to perform hypothesis testing. We performed this analysis on the basis of all invariants of J_1 and J_2. The above classification framework is the same as the one used in Vig et al. (2009) for analysis of the predictability of eye movements on natural movies.

7.3. Results

The box plot in Fig. 2 is constructed from the ROC (receiver-operator characteristic) scores obtained for the different invariants of J_1 and J_2 (horizontal axis of the plot) over all training and test set realizations. ROC curves illustrate the possible tradeoffs between true-positive and false-positive decisions that would be expected from a classifier. The smaller the Area Under the Curve of the ROC (ROC score or AUC) the more the predictor resembles a random classifier, which has an AUC of 0.5. An AUC of 1.0 means perfect discrimination.

The left part of the figure shows that the predictability of eye movements increases with the intrinsic dimension of the signal, i.e., the rank of J_1: invariants that extract features with higher intrinsic dimension are more predictive. The right part of the figure shows the same effect for the higher-order structure tensor J_2. More importantly, the predictions based on J_2 are better than those based on J_1 (see figure caption for details). This confirms our hypothesis that redundancies are suppressed even in the more complex case of overlaid motions.

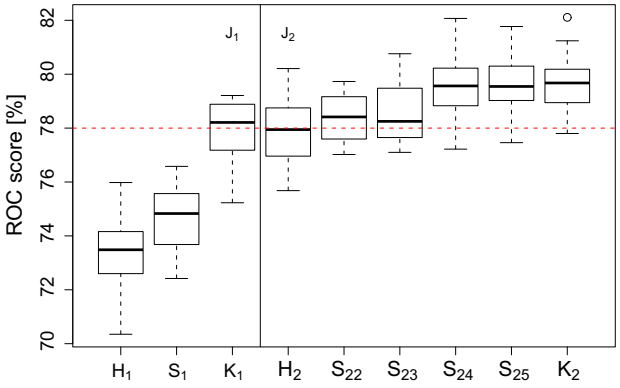

Fig. 2. Box plot comparing ROC scores of the invariants of J_1 and J_2 on overlaid movies over all 20 training/test set realizations (for a window size of about 2.5°). Horizontal lines indicate the median, the lower and upper quartiles, and the minimum and maximum values of the specific result set. Circles represent outliers. ROC scores were obtained for different invariants (horizontal axis). Comparison of the prediction performance was done by Wilcoxon's signed rank test. Note that when moving on the horizontal axis, there are steps where ROC score improves significantly; these are $p(H_1,S_1) = 19 \times 10^{-4}$, $p(S_1,K_1) = 0.89 \times 10^{-4}$, and $p(K_1,S_{24}) = 1.4 \times 10^{-4}$. Further significant differences are $p(H_1,H_2) = p(S_1,S_{22}) = p(S_1,S_{25}) = 0.89 \times 10^{-4}$, and $p(K_1,K_2) = 1.03 \times 10^{-4}$. Finally note that overall, all invariants of J_2 and the highest-order invariant of J_1 give a very high prediction rate (median 78%, indicated by the red dotted line, or higher).

8. Discussion

Neural systems are known to encode changes, a principle that can increase the efficiency of a visual representation. When the input is one-dimensional, encoding changes as deviations from a constant signal is straightforward. Multidimensional signals, however, can be constant in different ways, and therefore the problem of how to encode changes is more complex. One basic classification of how a multidimensional signal may change is due to its intrinsic dimension (Zetzsche & Barth, 1990). In Barth and Watson (2000) it has been shown how the principle of encoding only features with higher intrinsic dimensions relates to visual motion selectivity. Based on a generalization of the structure tensor, we have here presented an extension of these concepts to the case of multiple motions – see Table 2.

Moreover, we have here shown how to determine the basic types of overlaid motion signals based on the rank of the generalized structure tensor. To experience this, the reader can generate different motion patterns that correspond to the different lines of Table 2 by using the interactive tool. The more reddish and the less green the entries in that line are, the more undefined the motion is. The classical example is that of the aperture problem in the second line of Table 2. In this case, the motion percept is mainly determined by the shape of the aperture. We have presented the 2D–1D entrainment effect as an example of a higher-order aperture problem. Such problems occur when the generalized structure tensor is ill-conditioned.

Spatio-temporal scales are defined by the support of the derivatives. A further scale parameter is the support of the integration kernel ω, which should correspond to the region for which the motion model is valid. The convolution with ω is a linear pooling of nonlinear responses, which, in turn, are products of responses of linear filters, e.g., J_2 consists of terms such as $\omega * f_{xx} f_{tt}$. We have not addressed the relationship between the initial linear integration scale (support of the derivatives, size of V1-like receptive fields) and the nonlinear-linear pooling scale (the size over which the nonlinear terms are integrated).

A further limitation is that we do not deal with color, but we have shown elsewhere (Mota, Stuke, & Barth, 2006) how the approach can be extended to deal with multispectral images, and how color can help to solve the problem of estimating multiple orientations and motions.

In order to better understand the relationship between the individual motion layers and the global motion of multiple transparent layers, we have used the projective-plane representation of motion. Single points in the projective plane represent well-defined motions. The motion layers are lines or points in the projective plane. If their intersection defines a single point, this point will be the single coherent motion that we see. Multiple points generate transparent motions, and lines generate incoherent motions in the sense that all motions that correspond to points on a line are admissible. An obvious limitation of the projective-plane

representation of motion is that it is, like the Fourier-domain representation, a global representation of motion patterns and therefore does not include a description of apertures. Moreover, we have designed and implemented an interactive tool for generating motion patterns. We believe that the interactive tool can be helpful for better understanding multiple-motion stimuli and their associated percepts.

Finally, our experimental results show that eye movements on overlaid natural movies can be predicted with high accuracy by using the invariants of the generalized structure tensors J_1 and J_2 (prediction rates of over 70%). However, the higher order invariants of J_2 (the tensor for two motions) are significantly more predictive (see Fig. 2). This demonstrates that our approach can be successfully applied to complex natural stimuli, and that it can yield very good predictions of the observed behavior with just a small number of standard parameters.

Overall, we hope to have contributed to the understanding of what kinds of nonlinearities are needed to deal with the basic types of signal variations in the visual input, such as to obtain a more efficient representation, where higher-order redundancies are suppressed.

Acknowledgments

Our work was supported by the Deutsche Forschungsgemeinschaft (DFG) under Ba 1176/7, by the DAAD and by the Graduate School for Computing in Medicine and Life Sciences (which is funded by Germany's Excellence Initiative [DFG GSC 235/1]). Also, our research has received funding from the European Commission within the GazeCom project (IST-C-033816) of the FP6. All views herein are those of the authors alone; the European Commission is not liable for any use made of the information. We thank Martin Haker for implementing parts of the interactive tool.

Appendix A. The spatio-temporal structure tensor

For an image sequence $f(x,y,t)$, the structure tensor is defined as

$$J = J_1 = \omega * \begin{pmatrix} f_x^2 & f_x f_y & f_x f_t \\ f_x f_y & f_y^2 & f_y f_t \\ f_x f_t & f_y f_t & f_t^2 \end{pmatrix}. \quad (21)$$

If the trace of J differs from zero, the intrinsic dimension is at least one but could be higher; thus the trace detects all kinds of spatio-temporal changes that are not further differentiated. The determinant detects non-stationary 2D features such as corners and dots. As shown in Barth (2000), the minors of J encode the motion: if f is $f = (x - vt)$, the motion vector v is

$$v = \left(\frac{M_{31}}{M_{11}}, -\frac{M_{21}}{M_{11}}\right). \quad (22)$$

The minors M_{ij} are the determinants of the matrices obtained from J by eliminating row $4-i$ and column $4-j$, e.g., $M_{11} = \omega * f_x \cdot \omega * f_y - \omega * f_x f_y \cdot \omega * f_x f_y$.

Appendix B. How to use the interactive tool

The interactive tool (IT) displays the synthesized motion in the left panel and the projective plane in the right panel. In addition it has a graphical user interface (GUI) with menus and buttons. The right panel and the GUI are used to define the motion layers. The left panel and the GUI are used to define the aperture and view the resulting motion sequence. The intensities are displayed directly without any correction of possible monitor nonlinearities.

B.1. Projective-plane window (right panel)

The spatial patterns in the motion layers can be of two types: either a sinusoidal grating (spatial 1D pattern) or a spatial noise pattern (spatial 2D pattern).

A grating is defined by a line in the projective plane. After activating the "Grating" button (bottom left), the user draws the line by clicking and dragging in the right panel. The first left-button mouse click defines one point and the release of the button a second point of the desired line. A line will appear extending from the point where the mouse button was first pressed to the point the mouse was released at. The two points continue to be marked with anchors that can be dragged any time in order to modify the line. The distance between the anchors defines the spatial frequency of the grating. The distance of the line from the center of the panel defines the slowest admissible velocity. The orientation of the line equals the orientation of the grating.

To define a noise pattern, the user should first select the "Noise" button (below the "Grating" button) and then click on the right panel once (again by using the left mouse button). The position of that click defines the motion of the layer (remember: translating 2D spatial patterns are points in the projective plane). The speed is defined by the distance to the center and the direction of motion by the direction vector from the point to the center. Thus, as one moves the point towards the center, the motion will slow down. The circle drawn around the dot indicates the variance of the noise pattern. To increase the speed of the grating or the noise pattern, move the line or dot away from the center. To change the orientation of the grating, rotate the line by dragging one or both anchors. To change the variance of the noise pattern, increase the radius of the circle drawn around the point by selecting and dragging the circle line.

B.2. Multiple layers

To add another layer, repeat the procedures above. The IT is currently limited to three motion layers. The overlay can be either additive or multiplicative. The type of overlay can be selected by the two radio buttons "Additive" and "Multiplicative".

B.3. Aperture and movie window (left panel)

In addition to displaying the resulting movie, the left panel can be used to draw different apertures. The default aperture is a circle. Other apertures can be selected from the "Aperture" pull-down menu. Furthermore, the apertures can be translated and rotated. Finally, the apertures can be reproduced by the "Edit Copy Paste" feature and also deleted. Different types of apertures can be overlaid. Each new aperture needs to be first selected from the pull-down menu. After a selected aperture has been drawn, the mouse cursor switches to the selection mode, so that the user can select one or more apertures, e.g., for rotation or deletion.

B.4. Remaining buttons and features

The "Start" button will start the movie in the left panel. The same button will also stop the movie once it runs. The "Clear" button will delete the selected motion layer (grating or noise) in the right panel, and "Clear All" will delete all the motion layers. The apertures can only be deleted by selection and the "Delete" option in the "Edit" menu. The "Detach" button will detach the applet from the browser so that one can easily place the IT window in a preferred position. One can stop (or restart) the motion of just one layer by a double click on the anchor of a line or a point in the projective plane.

References

Aach, T., Mota, C., Stuke, I., Mühlich, M., & Barth, E. (2006). Analysis of superimposed oriented patterns. *IEEE Transactions on Image Processing, 15*(12), 3690–3700.

Adelson, E. H., & Bergen, J. R. (1991). The plenoptic function and the elements of early vision. In M. S. Landy & J. A. Movshon (Eds.), *Computational models of visual processing* (pp. 3–20). Cambridge, MA: MIT Press.

Andrews, T. J., & Schluppeck, D. (2000). Ambiguity in the perception of moving stimuli is resolved in favour of the cardinal axes. *Vision Research, 40*, 3485–3493.

Barth, E. (2000). The minors of the structure tensor. In G. Sommer (Ed.), *Mustererkennung 2000* (pp. 221–228). Berlin: Springer.

Barth, E., Stuke, I., Aach, T., & Mota, C. (2003). *Spatio-temporal motion estimation for transparency and occlusion. Proceedings of the IEEE International Conference on Image Processing* (Vol. III). Barcelona, Spain: IEEE Signal Processing Soc..

Barth, E., & Watson, A. B. (2000). A geometric framework for nonlinear visual coding. *Optics Express, 7*(4), 155–165.

Bergen, J. R., Burt, P. J., Hingorani, R., & Peleg, S. (1992). A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(9), 886–896.

Böhme, M., Dorr, M., Krause, C., Martinetz, T., & Barth, E. (2006). Eye movement predictions on natural videos. *Neurocomputing, 69*(16–18), 1996–2004.

Born, R. T., & Bradley, C. D. (2005). Structure and function of visual area MT. *Annual Reviews in Neuroscience, 28*, 157–189.

Braddick, O. (1993). Segmentation versus integration in visual motion processing. *TINS, 16*(7), 263–268.

Dorr, M., Martinetz, T., Gegenfurtner, K., & Barth, E. (2010). Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision, 10*(10), 1–7.

Braddick, O., & Qian, N. (2001). The organization of global motion and transparency. In J. M. Zanker & J. Zeil (Eds.), *Motion vision – computational neural and ecological constraints* (pp. 86–111). Berlin, Heidelberg, New York: Springer Verlag.

Field, D. J. (1987). Relations between the statistics of natural images and the response profiles of cortical cells. *Journal of the Optical Society of America A, 4*, 2379–2394.

Jähne, B., Mester, R., Barth, E., & Scharr, H. (Eds.). (2007). *Complex motion, first international workshop, IWCM 2004, Günzburg, Germany, October 12–14, 2004. Revised Papers. Lecture Notes in Computer Science* (Vol. 3417). Springer.

Labusch, K., Barth, E., & Martinetz, T. (2009). Sparse coding neural gas: Learning of overcomplete data representations. *Neurocomputing, 72*(7–9), 1547–1555.

Mota, C., & Barth, E. (2000). On the uniqueness of curvature features. In G. Baratoff & H. Neumann (Eds.), *Dynamische perzeption. Proceedings in artificial intelligence* (Vol. 9, pp. 175–178). Infix Verlag.

Mota, C., Dorr, M., Stuke, I., & Barth, E. (2004). Categorization of transparent-motion patterns using the projective plane. *International Journal of Computer and Information Science, 5*(2), 129–140.

Mota, C., Stuke, I., & Barth, E. (2001). Analytic solutions for multiple motions. *Proceedings of the IEEE international conference on image processing* (Vol. II, pp. 917–920). Thessaloniki, Greece: IEEE Signal Processing Soc..

Mota, C., Stuke, I., & Barth, E., 2006. The intrinsic dimension of multispectral images. In: *MICCAI workshop on biophotonics imaging for diagnostics and treatment* (pp. 93–100).

Mulligan, J. B. (1992). Motion transparency is restricted to two planes. *Investigative Ophthalmology & Visual Science, 33*(Suppl.), 1049.

Mulligan, J. B. (1993). Nonlinear combination rules and the perception of visual motion transparency. *Vision Research, 33*(14), 2021–2030.

Olshausen, B. A. (2009). *Personal communication*.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381*, 607–609.

Pack, C. C., Gartland, A. J., & Born, R. T. (2004). Integration of contour and terminator signals in visual area MT of alert macaque. *The Journal of Neuroscience, 24*(13), 3268–3280.

Pack, C. C., Livingstone, M. S., Duffy, K. R., & Born, R. T. (2003). End-stopping and the aperture problem: Two-dimensional motion signals in macaque V1. *Neuron, 39*, 671–680.

Rust, N. C., Mante, V., Simoncelli, E. P., & Movshon, J. A. (2006). How MT cells analyse the motion of visual patterns. *Nature Neuroscience, 9*(11), 1421–1431.

Shizawa, M., & Mase, K. (1990). Simultaneous multiple optical flow estimation. In *Proceedings of the IEEE conference computer vision and pattern recognition, Atlantic City* (pp. 274–278).

Srinivasan, M. V. (1990). Generalized gradient schemes for the measurement of two-dimensional image motion. *Biol Cybernetics, 63*, 421–431.

Stuke, I., Barth, E., & Mota, C. (2006). Estimation of multiple orientations and multiple motions in multi-dimensional signals. In *IEEE XIX Brazilian symposium on computer graphics and image processing (SIBGRAPI'06)* (pp. 341–348).

Vig, E., Dorr, M., & Barth, E. (2009). Efficient visual coding and the predictability of eye movements on natural movies. *Spatial Vision, 22*(5), 397–408.

Watson, A. B., & Ahumada, A. J. (1983). *A look at motion in the frequency domain. Motion: Perception and representation* (Vol. 2). New York: Association for Computing Machinery. pp. 1–10.

Wuerger, S., Shapley, R., & Rubin, N. (1996). On the visually perceived direction of motion by Hans Wallach: 60 years later. *Perception, 25*, 1317–1367.

Zetzsche, C., & Barth, E. (1990). Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research, 30*, 1111–1117.

Zetzsche, C., Barth, E., & Wegmann, B. (1993). The importance of intrinsically two-dimensional image features in biological vision and picture coding. In A. B. Watson (Ed.), *Digital images and human vision* (pp. 109–138). MIT Press.

Please cite this article in press as: Barth, E., et al. Efficient coding and multiple motions. *Vision Research* (2010), doi:10.1016/j.visres.2010.08.011