Project no.: IST-FP6-FET-16276-2

Project full title: Learning to emulate perception action cycles in a driving school scenario

Project Acronym: DRIVSCO

Deliverable no: D1.1

Title of the deliverable: Specification of the Components for the Micro-Chip

Date of Delivery:	14.6.2008
Organization name of lead contractor for this deliverable:	UGR
Author(s):	E.Ros (UGR), S. Sabatini (UGE), M. van Hulle (KUL), N. Krüger (SDU)
Participant(s):	UGR, UGE, KUL, SDU, BCCN
Work package contributing to the deliverable:	WP1
Nature:	R
Version:	2.0 (revised 10/06/2008)
Total number of pages:	81
Start date of project:	1 Feb. 2006 **Duration:** 42 months

Project Co-funded by the European Commission		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Summary: This deliverable defines all the modules to be integrated into the chip (low level vision hardware) and gives their specifications.

D1.1. Specification of the components for the micro-chip

DRIVSCO

(Deliverable due to Month 13: March 2007)

Brief outline of the deliverable

Abstract

This deliverable defines all the modules to be integrated into the chip (low level vision hardware) and gives the specifications of all these modules. In an attached *excel file* (*Appendix D*) we provide a tool that can be used to evaluate how changing the specifications of the different modules to be implemented on the chip will demand more or less temporal memory resources and data transmission bandwidth.

1. The whole system-on-chip

The whole system has different stages:

a. Spatio-temporal filters to obtain a harmonic representation of the visual signal. These modules process the rare images and produce responses tuned to the filter banks used. *Appendix B* describes the theoretical framework and the implementation issues related to the filter bank construction techniques (Gabor, Stereeable, monogenic transform, etc). It is important to define the filter bank that will be used. *Appendix A* and *Appendix C* address an accuracy vs. computation resources study about different alternatives. The size of this filter bank (at different orientations) will have a high impact on the actual hardware resources of the low level vision system.

b. Single modality vision modules. The responses of this bank of filters are combined to obtain motion, stereo or local spatial primitives (energy, phase and orientation). The target specifications of these vision modalities (number of computed scales, spatial and temporal resolutions) will have a high impact on the complete system resources requirement (this is outlined in *Appendix D*).

c. Cross-modality interactions. In *Appendix E* are outlined different interactions that are good candidates to be integrated on-chip. *Appendix F* includes some more concrete details about geometry issues related with motion-in-depth.

Appendices:

A. Joint paper: Sabatini et al. "Compact (and Accurate) Early Vision Processing in the Harmonic Space". Accepted in the VISAPP'07 conference (2nd Intern. Conf. on Computer Vision Theory and Applications), Barcelona 8-11 March 2007.
B. Filter design techniques. Technical report.

C. Filter evaluation: optic flow and disparity. Technical report.
D. *Excel file* comparing the impact on memory and bandwidth of different system specifications.
E. Cross-modality examples. Technical report.
F. Motion-in-depth. Geometrical considerations. Technical report.

Table of Contents

Appendix A

Joint paper: Sabatini et al. "Compact (and Accurate) Early Vision Processing in the Harmonic Space". Accepted in the VISAPP'07 conference (2nd Intern. Conf. on Computer Vision Theory and Applications), Barcelona 8-11 March 2007.

COMPACT (AND ACCURATE) EARLY VISION PROCESSING IN THE HARMONIC SPACE

Silvio P. Sabatini, Giulia Gastaldi, Fabio Solari
Dipartimento di Ingegneria Biofisica ed Elettronica, University of Genoa, via Opera Pia 11a, Genova, Italy
silvio.sabatini@unige.it, giulia@dibe.unige.it, fabio.solari@unige.it

Karl Pauwels, Marc van Hulle
Laboratorium voor Neuro- en Psychofysiologie, K.U.Leuven, Belgium
karl.pauwels@med.kuleuven.be, marc.vanhulle@med.kuleuven.ac.be

Javier Díaz, Eduardo Ros
Departamento de Arquitectura y Tecnologia de Computadores, University of Granada, Spain
jdiaz@atc.ugr.es, eduardo@atc.ugr.es

Nicolas Pugeault
School of Informatics, University of Edinburgh, U.K.
npugeaul@inf.ed.ac.uk

Norbert Krüger
The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
norbert@mip.sdu.dk

Keywords: Phase-based techniques, multidimensional signal processing, filter design, optic flow, stereo vision.

Abstract: The efficacy of anisotropic versus isotropic filtering is analyzed with respect to general phase-based metrics for early vision attributes. We verified that the spectral information content gathered through oriented frequency channels is characterized by high compactness and flexibility, since a wide range of visual attributes emerge from different hierarchical combinations of the same channels. We observed that it is preferable to construct a multichannel, multiorientation representation, rather than using a more compact representation based on an isotropic generalization of the analytic signal. The complete harmonic content is then combined in the phase-orientation space at the final stage, only, to come up with the ultimate perceptual decisions, thus avoiding an "early condensation" of basic features. The resulting algorithmic solutions reach high performance in real-world situations at an affordable computational cost.

1 INTRODUCTION

Although the basic ideas underlying early vision appear deceptively simple and their computational paradigms are known for a long time, early vision problems are difficult to quantify and solve. Moreover, in order to have high algorithmic performance in real-world situations, a large number of channels should be integrated with high efficiency. From a computational point of view, the visual signal should be processed in a "unifying" perspective that will allow us to share the maximum number of resources. From an implementation point of view, the resulting algorithms and architectures could fall short of their expectations when the high demand of computational resources for multichannel spatio-temporal filtering of high resolution images conflicts with real-time requirements. Several approaches and solutions have been proposed in the literature to accelerate the computation by means of dedicated hardware (e.g., see (Diaz et al., 2006; Kehtarnavaz and Gamadia, 2005)). Yet, the large number of products that must be computed to calculate each single pixel of each single frame for a couple of stereo images and at each time step still represents the main bottleneck. This is particularly true for stereo and motion problems to construct 3D representations of the world, for which establishing image correspondences in space and space-time is a prerequisite, but also their most challenging part.

In this paper, we propose (1) to define a systematic approach to obtain a “complete” harmonic analysis of the visual signal and (2) to integrate efficient multi-channel algorithmic solutions to obtain high performance in real-world situations, and, at the same time, an affordable computational load.

2 MULTICHANNEL BANDBASS REPRESENTATION

An efficient (internal) representation is necessary to guarantee all potential visual information can be made available for higher level analysis. At an early level, feature detection occurs through initial local *quantitative* measurements of basic image properties (e.g., edge, bar, orientation, movement, binocular disparity, colour) referable to spatial differential structure of the image luminance and its temporal evolution (cf. linear cortical cell responses). Later stages in vision can make use of these initial measurements by combining them in various ways, to come up with categorical *qualitative* descriptors, in which information is used in a non-local way to formulate more global spatial and temporal predictions. The receptive fields of the cells in the primary visual cortex have been interpreted as fuzzy differential operators (or local *jets* (Koenderink and van Doorn, 1987)) that provide regularized partial derivatives of the image luminance in the neighborhood of a given point $\mathbf{x} = (x,y)$, along different directions and at several levels of resolution, simultaneously. Given the 2D nature of the visual signal, the spatial direction of the derivative (i.e., the orientation of the corresponding local filter) is an important “parameter”. Within a local jet, the directionally biased receptive fields are represented by a set of similar filter profiles that merely differ in orientation.

Alternatively, considering the space/spatial-frequency duality (Daugman, 1985), the local jets can be described through a set of independent spatial-frequency channels, which are selectively sensitive to a different limited range of spatial frequencies. These spatial-frequency channels are equally apt as the spatial ones. From this perspective, it is formally possible to derive, on a local basis, a complete harmonic representation (phase, energy/amplitude, and orientation) of any visual stimulus, by defining the associated analytic signal in a combined space-frequency domain through filtering operations with complex-valued band-pass kernels. Formally, due to the impossibility of a direct definition of the analytic signal in two dimensions, a 2D spatial frequency filtering would require an association between spatial frequency and orientation channels. Basically, this association can be handled either (1) ‘separately’, for each orientation channel, by using Hilbert pairs of band-pass filters that display symmetry and antisymmetry about a steerable axis of orientation, or (2) ‘as-a-whole’, by introducing a 2D isotropic generalization of the analytic signal: the monogenic signal (Felsberg and Sommer, 2001), which allows us to build isotropic harmonic representations that are independent of the orientation (i.e., omnidirectional). By definition, the monogenic signal is a 3D phasor in spherical coordinates and provides a framework to obtain the harmonic representation of a signal respect to the dominant orientation of the image that becomes part of the representation itself.

In the first case, for each orientation channel θ, an image $I(\mathbf{x})$ is filtered with a complex-valued filter:

$$f_A^{\theta}(\mathbf{x}) = f^{\theta}(\mathbf{x}) - i\, f_{\mathcal{H}}^{\theta}(\mathbf{x}) \qquad (1)$$

where $f_{\mathcal{H}}^{\theta}(\mathbf{x})$ is the Hilbert transform of $f^{\theta}(\mathbf{x})$ with respect to the axis orthogonal to the filter’s orientation. This results in a complex-valued *analytic image*:

$$Q_A^{\theta}(\mathbf{x}) = I * f_A^{\theta}(\mathbf{x}) = C_{\theta}(\mathbf{x}) + iS_{\theta}(\mathbf{x}) \ , \qquad (2)$$

where $C_{\theta}(\mathbf{x})$ and $S_{\theta}(\mathbf{x})$ denote the responses of the quadrature filter pair. For each spatial location, the amplitude $\rho_{\theta} = \sqrt{C_{\theta}^2 + S_{\theta}^2}$ and the phase $\phi_{\theta} = \arctan(S_{\theta}/C_{\theta})$ envelopes measure the harmonic information content in a limited range of frequencies and orientations to which the channel is tuned.

In the second case, the image $I(\mathbf{x})$ is filtered with a *spherical quadrature filter* (SQF):

$$f_M(\mathbf{x}) = f(\mathbf{x}) - (i,j) \cdot \mathbf{f}_{\mathcal{R}}(\mathbf{x}) \qquad (3)$$

defined by a rotation invariant even $f(\mathbf{x})$ filter) and a vector-valued isotropic odd filter $\mathbf{f}_{\mathcal{R}}(\mathbf{x}) = (f_{\mathcal{R},1}(\mathbf{x}), f_{\mathcal{R},2}(\mathbf{x}))^T$, obtained by the Riesz transform of $f(\mathbf{x})$ (Felsberg and Sommer, 2001). This results in a *monogenic image*:

$$\begin{aligned} Q_M(\mathbf{x}) &= I * f_M(\mathbf{x}) = C(\mathbf{x}) + (i,j)\mathbf{S}(\mathbf{x}) \qquad (4) \\ &= C(\mathbf{x}) + iS_1(\mathbf{x}) + jS_2(\mathbf{x}) \end{aligned}$$

where, using the standard spherical coordinates,

$$\begin{aligned} C(\mathbf{x}) &= \rho(\mathbf{x}) \cos\varphi(\mathbf{x}) \\ S_1(\mathbf{x}) &= \rho(\mathbf{x}) \sin\varphi(\mathbf{x}) \cos\vartheta(\mathbf{x}) \\ S_2(\mathbf{x}) &= \rho(\mathbf{x}) \sin\varphi(\mathbf{x}) \sin\vartheta(\mathbf{x}) \ . \end{aligned}$$

The amplitude of the monogenic signal is the vector norm of f_M: $\rho = \sqrt{C^2 + S_1^2 + S_2^2}$, as in the case of the analytic signal, and, for an intrinsically one-dimensional signal, φ and ϑ are the dominant phase and the dominant orientation, respectively.

In this work, we want to analyze the efficacy of the two approaches in obtaining a complete and efficient representation of the visual signal. To this end, we consider, respectively, a discrete set of oriented (i.e., anisotropic) Gabor filters and a triplet of isotropic spherical quadrature filters defined on the basis of the monogenic signal. Moreover, as a choice in the middle between the two approaches, we will also take into consideration the classical steerable filter approach (Freeman and Adelson, 1991) that allows a continuous steerability of the filter respect to any orientation. In this case, the number of basis kernels to compute the oriented outputs of the filters depends on the derivative order (n) of a Gaussian function. The basis filters corresponding to $n = 2$ or $n = 4$ turned out as an acceptable compromise between the representation efficacy and the computational efficiency.

For all the filters considered, we chose the design parameters to have a good coverage of the space-frequency domain and to keep the spatial support (i.e., the number of taps) to a minimum, in order to cut down the computational cost. Therefore, we determined the smallest filter on the basis of the highest allowable frequency without aliasing, and we adopted a pyramidal technique (Adelson et al., 1984) as an economic and efficient way to achieve a multiresolution analysis (see also Section 3.2). Accordingly, we fixed the maximum radial peak frequency (ω_0) by considering the Nyquist condition and a constant relative bandwidth of one octave ($\beta = 1$), that allows us to cover the frequency domain without loss of information. For Gabor and steerable filters, we should also consider the minimum number of oriented filters to guarantee a uniform orientation coverage. This number still depends on the filter bandwidth and it is related to the desired orientation sensitivity of the filter (e.g., see (Daugman, 1985; Fleet and Jepson, 1990)); we verified that, under our assumptions, it is necessary to use at least eight orientations. To satisfy the quadrature requirement all the even symmetric filters have been "corrected" to cancel the DC sensitivity. The monogenic signal has been constructed from a radial bandpass filter obtained by summing the corrected bank of oriented even Gabor filters. All the filters have been normalized prior to their use in order to have constant unitary energy. A detailed description of the filters used can be found at `www.pspc.dibe.unige.it/VISAPP07/`.

3 PHASE-BASED EARLY VISION ATTRIBUTES

3.1 Basic principles

During the last decades, the phase from local bandpass filtering has gained increasing interest in the computer vision community and has led to the development of a wide number of phase-based feature detection algorithms in different application domains (Sanger, 1988; Fleet et al., 1991; Fleet and Jepson, 1990; Fleet and Jepson, 1993; Kovesi, 1999; Gautama and Van Hulle, 2002). Yet, to the best of our knowledge, a systematic analysis of the basic descriptive properties of the phase has never been done. One of the key contributions of this paper is the formulation a *single* unified representation framework for early vision grounded on a proper phase-based metrics. We verified that the resulting representation is characterized by high compactness and flexibility, since a wide range of visual attributes emerges from different hierarchical combinations of the same channels. The harmonic representation will be the base for a systematic phase-based interpretation of early vision processing, by defining perceptual features on measures of phase properties. From this perspective, edge and contour information can come from *phase-congruency*, motion information can be derived from the *phase-constancy* assumption, while matching operations, such as those used for disparity estimation, can be reduced to *phase-difference* measures. In this way, simple local relational operations capture signal features, which would be more complex and less stable if directly analysed in the spatio-temporal domain.

Contrast direction and orientation. Traditional gradient-based operators are used to detect sharp changes in image luminance (such as step edges), and hence are unable to properly detect and localize other feature types. As an alternative, phase information can be used to discriminate different features in a contrast independent way, by searching for patterns of order in the phase component of the Fourier transform (Owens, 1994). Abrupt luminance transitions, as in correspondence of step edges and line features are, indeed, points where the Fourier components are maximally in phase. Therefore, both they are signaled by peaks in the local energy, and the phase information can be used to discriminate among different kinds of contrast transition (Kovesi, 1999), e.g., a phase of $\pi/2$ corresponds to a dark-bright edge, whereas a phase of 0 corresponds to a bright line on dark background (see also (Krüger and Felsberg, 2003)) .

Binocular disparity. In a first approximation, the phase-based stereopsis defines the disparity $\delta(x)$ as the one-dimensional shift necessary to align, along the direction of the (horizontal) epipolar lines, the phase values of bandpass filtered versions of the stereo image pair $I^R(x)$ and $I^L(x) = I^R[x + \delta(x)]$ (Sanger, 1988). Formally,

$$\delta(x) = \frac{\lfloor \phi^L(x) - \phi^R(x) \rfloor_{2\pi}}{\omega(x)} = \frac{\lfloor \Delta\phi(x) \rfloor_{2\pi}}{\omega(x)} \qquad (5)$$

where $\omega(x)$ is the average instantaneous frequency of the bandpass signal, at point x, that, under a linear phase model, can be approximated by ω_0 (Fleet et al., 1991). Equivalently, the disparity can be directly obtained from the principal part of phase difference, without explicit manipulation of the left and right phase and without incurring the 'wrapping' effects on the resulting disparity map (Solari et al., 2001):

$$\lfloor \Delta\phi \rfloor_{2\pi} = \lfloor \arg(Q^L Q^{*R}) \rfloor_{2\pi} \qquad (6)$$

where Q^* denotes complex conjugate of Q.

Normal Flow. Considering the conservation property of local phase measurements (phase constancy), image velocities can be computed from the temporal evolution of equi-phase contours $\phi(\mathbf{x}, t) = c$ (Fleet et al., 1991). Differentiation with respect to t yields:

$$\nabla\phi \cdot \mathbf{v} + \phi_t = 0 \ , \qquad (7)$$

where $\nabla\phi = (\phi_x, \phi_y)$ is the spatial and ϕ_t is the temporal phase gradient. Note that, due to the aperture problem, only the velocity component along the spatial gradient of phase can be computed (normal flow). Under a linear phase model, the spatial phase gradient can be substituted by the radial frequency vector $\omega = (\omega_x, \omega_y)$. In this way, the component velocity $\mathbf{v}_c$ can be estimated directly from the temporal phase gradient:

$$\mathbf{v}_c = -\frac{\phi_t}{\omega_0} \cdot \frac{\omega}{|\omega|} \ . \qquad (8)$$

The temporal phase gradient can be obtained by fitting a linear model to the temporal sequence of spatial phases (using *e.g.* five subsequent frames) (Gautama and Van Hulle, 2002):

$$(\phi_t, p) = \underset{\phi_t, p}{\operatorname{argmin}} \sum_t \big((\phi_t \cdot t + p) - \phi(t) \big)^2 \ , \qquad (9)$$

where p is the intercept.

Motion-in-depth. The perception of motion in the 3D space relates to 2nd-order measures, which can be gained either by interocular velocity differences or temporal variations of binocular disparity (Harris and Watamaniuk, 1995). Recently (Sabatini et al., 2003), it has been proved that both cues provide the same information about motion-in-depth (MID), when the rate of change of retinal disparity is evaluated as a total temporal derivative of the disparity:

$$\frac{d\delta}{dt} \simeq \frac{\partial\delta}{\partial t} = \frac{\phi_t^L - \phi_t^R}{\omega_0} \simeq v^R - v^L \ , \qquad (10)$$

where v^R and v^L are the velocities along the epipolar lines. Through the chain rule in the evaluation of the temporal derivative of phases, we obtain information about MID directly from convolutions Q of stereo image pairs and by their temporal derivatives Q_t, eluding explicit calculation and differentiation of phase and the attendant problem of phase unwrapping:

$$\frac{\partial\delta}{\partial t} = \left[\frac{\mathrm{Im}[Q_t^L Q^{*L}]}{|Q^L|^2} - \frac{\mathrm{Im}[Q_t^R Q^{*R}]}{|Q^R|^2} \right] \frac{1}{\omega_0} \ . \qquad (11)$$

3.2 Channel interactions

The harmonic information made available by the different basis channels must be properly integrated across both multiple scales and multiple orientations to optimally detect and localise different features at different levels of resolution in the visual signal. In general, for what concerns the scale, a multiresolution analysis can be efficiently implemented by a coarse-to-fine strategy that helps us to deal with large features values, which are otherwise unmeasurables by the small filters we have to use in order to achieve real-time performance. Specifically, a coarse-to-fine Gaussian pyramid (Adelson et al., 1984) is constructed, where each layer is separate by an octave scale. Accordingly, the image is increasingly blurred with a Gaussian kernel $g(\mathbf{x})$ and subsampled:

$$I_k(\mathbf{x}) = (\mathcal{S}(g * I_{k-1}))(\mathbf{x}) \ . \qquad (12)$$

At each pyramid level k the subsampling operator $\mathcal{S}$ reduces to a half the image resolution respect to the previous level $k-1$. The filter response image Q_k at level k is computed by filtering the image I_k with the fixed kernel $f(\mathbf{x})$:

$$Q_k(\mathbf{x}) = (f * I_k)(\mathbf{x}) \ . \qquad (13)$$

For what concerns the interactions across the orientations a key distinction must be done according that one uses isotropic or anisotropic filtering.

Isotropic filtering. The monogenic signal directly provides a *single* harmonic content with respect to the dominant orientation:

$$\rho(\mathbf{x}) \overset{\text{def}}{=} \sqrt{C^2(\mathbf{x}) + |\mathbf{S}(\mathbf{x})|^2} = \mathcal{E}(\mathbf{x})$$

$$\theta(\mathbf{x}) \overset{\text{def}}{=} \operatorname{atan2}(S_2(\mathbf{x}), S_1(\mathbf{x})) = \vartheta(\mathbf{x})$$

$$\phi(\mathbf{x}) \overset{\text{def}}{=} \operatorname{sign}[\mathbf{S}(\mathbf{x}) \cdot \mathbf{n}_\vartheta(\mathbf{x})] \operatorname{atan2}(|\mathbf{S}(\mathbf{x})|, C(\mathbf{x})) = \varphi(\mathbf{x}),$$

$$\text{with} \quad \mathbf{n}_\vartheta(\mathbf{x}) = (\cos\vartheta(\mathbf{x}), \sin\vartheta(\mathbf{x})) \ .$$

Anisotropic filters. Basic feature interpolation mechanisms must be introduced. More specifically, if we name E_q and ϕ_q the "oriented" energy and the "oriented" phase extracted by the filter f_q steered to the angle $\theta_q = q\pi/K$, the harmonic features computed with this filter orientation are:

$$\begin{aligned}
\rho_q(\mathbf{x}) &= \sqrt{C_q^2(\mathbf{x}) + S_q^2(\mathbf{x})} = E_q(\mathbf{x}) \\
\theta_q(\mathbf{x}) &= \frac{q\pi}{K} \\
\phi_q(\mathbf{x}) &= \mathrm{atan2}\left(S_q(\mathbf{x}), C_q(\mathbf{x})\right) \ .
\end{aligned}$$

Under this circumstance, we require to interpolate the feature values computed by the filter banks in order to estimate the filter's output at the proper signal orientation. The strategies adopted for this interpolation are very different, and strictly depend on the 'computational theory' (in the Marr's sense (Marr, 1982)) of the specific early vision problem considered, as it will be detailed in the following.

Contrast direction and orientation. According to (Krüger and Felsberg, 2004) the phase is used to describe the local structure of i1D signals in an image. Therefore, we determine maxima of the local amplitude orthogonal to the main orientation with sub–pixel accuracy and compute orientation and phase information at this sub-pixel position using bi-linear interpolation in the phase–orientation space. Sub–pixel accuracy is achieved by computing the center of gravity in a window with size depending on the frequency level. For the bilinear interpolation we need to take care of the topology of the orientation–phase space that has the form of a half–torus. The precision of sub–pixel accuracy calculation as well as the precision of the phase estimate depending on the different harmonic representations is discussed in Section 4.

Binocular disparity. The disparity computation from Eq. (5) can be extended to two-dimensional filters at different orientations θ_q by projection on the epipolar line in the following way:

$$\delta_q(x) = \frac{\lfloor \phi_q^L(x) - \phi_q^R(x) \rfloor_{2\pi}}{\omega_0 \cos\theta_q} \ . \tag{14}$$

In this way, multiple disparity estimates are obtained at each location. These estimates can be combined by taking their median:

$$\delta(x) = \underset{q \in V(x)}{\mathrm{median}}\, \delta_q(x) \ , \tag{15}$$

where $V(x)$ is the set of orientations where valid component disparities have been obtained for pixel x. Validity can be measured by the filter energy.

A coarse-to-fine control scheme is used to integrate the estimates over the different pyramid levels (Bergen et al., 1992). A disparity map $\delta^k(x)$ is first computed at the coarsest level k. To be compatible with the next level, it must be upsampled, using an expansion operator $\mathcal{X}$, and multiplied by two:

$$d^k(x) = 2 \cdot \mathcal{X}\left(\delta^k(x)\right) \ . \tag{16}$$

This map is then used to reduce the disparity at level $k+1$, by warping the phase or filter outputs before computing the phase difference:

$$\delta_q^{k+1}(x) = \frac{\lfloor \phi^L(x) - \phi^R\left(x - d^k(x)\right) \rfloor_{2\pi}}{\omega_0 \cos\theta_q} + d^k(x) \ . \tag{17}$$

In this way, the remaining disparity is guaranteed to lie within the filter range. This procedure is repeated until the finest level is reached.

Optic flow. The reliability of each component velocity can be measured by the mean squared error (MSE) of the linear fit in Eq. (8) (Gautama and Van Hulle, 2002). Provided a minimal number of reliable component velocities are obtained (threshold on the MSE), an estimate of the full velocity can be computed for each pixel by integrating the valid component velocities (Gautama and Van Hulle, 2002):

$$\mathbf{v}(\mathbf{x}) = \underset{\mathbf{v}(\mathbf{x})}{\mathrm{argmin}} \sum_{q \in O(\mathbf{x})} \left(|\mathbf{v}_{c,q}(\mathbf{x})| - \mathbf{v}(\mathbf{x})^{\mathrm{T}} \frac{\mathbf{v}_{c,q}(\mathbf{x})}{|\mathbf{v}_{c,q}(\mathbf{x})|} \right)^2 \ , \tag{18}$$

where $O(\mathbf{x})$ is the set of orientations where valid component velocities have been obtained for pixel $\mathbf{x}$. A coarse-to-fine control scheme, similar to that of Section 3.2 is used to integrate the estimates over the different pyramid levels. Starting from the coarsest level k, the optic flow field $\mathbf{v}^k(\mathbf{x})$ is computed, expanded, and used to warp the phases or filter outputs at level $k+1$. For more details on this procedure we refer to (Pauwels and Van Hulle, 2006).

Motion-in-depth. Although the motion-in-depth is a 2nd-order measure, by exploiting the direct determination of the temporal derivative of the disparity (see Eq.11), the binocular velocity along the epipolar lines can be directly calculated for each orientation channel, and thence the motion-in-depth:

$$\mathrm{V_Z} = \underset{q \in W_L(x)}{\mathrm{median}}\, v_q^L(x) - \underset{q \in W_R(x)}{\mathrm{median}}\, v_q^R(x) \ , \tag{19}$$

where for each monocular sequence, $W(x)$ is the set of orientations for which valid components of velocities have been obtained for pixel x. As in the previous cases, a coarse-to-fine strategy is adopted to guarantee that the horizontal spatial shift between two consecutive frames lie within the filter range.

4 RESULTS

We are interested in computing different image features with the maximum accuracy and the lower processor requirements. The utilization of the different filtering approaches leads to different computing load requirements. Focusing on the convolutions operations on which the filters are based, we have analyzed each approach to evaluate their complexity. Spherical filters require three non-separable convolutions operations, which make this approach quite expensive in terms of the required computational resources. The eight oriented Gabor filters requires eight 2-D non separable convolution but they can be efficiently computed through a linear combination of separable kernels as it is indicated in (Nestares et al., 1998), thus significantly reducing the computational load. For steerable filters, quadrature oriented outputs are obtained from the filter bases composed of separable kernels. The higher is the Gaussian derivative order, the higher is the number of the basis filters. More specifically, the number of 1-D convolutions is given by $4n+6$ where n is the differentiation order.

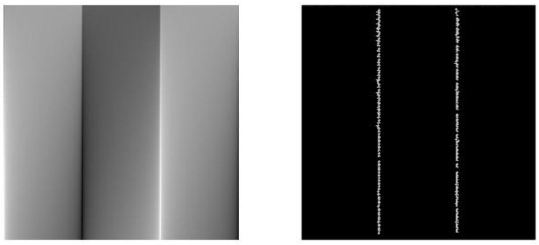
Summarizing, the complexity of computing the harmonic representation with the different set of filters is summarized in Table 1.

Table 1: Computational costs of convolution operators.

	# filters	# taps	products	sums
Gabor	24	11	264	240
s4	22	11	242	220
s2	14	11	154	140
SQF	3	11×11	363	360

The accuracy achieved by the different filters has been evaluated using synthetic images with well-known ground-truth feature.

Contrast direction and orientation. We have utilized a synthetic image (see Figure 1) where the feature type changes from a step edge to a line feature from top to bottom (Kovesi, 1999). By rotating the image by stepwise angles in $[0, 2\pi)$, we constructed a set of test images and measured the contour localization accuracy, phase and orientation with the different approaches, comparing the results with the ground-truth. In Table 2 the mean errors in localisation, orientation and phase and their standard deviations are reported. It is worth noting that the features were extracted with sub-pixel accuracy. We can see that Gabor and 4th-order steerable filters (s4) produce the most accurate results for phase and edge localization, with low variance. Second order steerable filters (s2) and SQFs seem very noisy in their phase estimation.



(a) (b)

Figure 1: (a) Test image representing a continuum of phases taking values in $(-\pi, \pi]$ corresponding to a continuum of oriented grey-level structures as expressed in a changing "circular" manifold (cf. (Kovesi, 1999)). The feature type changes progressively from a step edge to a line feature, while retaining perfect phase congruency. (b) Phase-based localization of contours obtained by the Gabor filters.

Table 2: Accuracy evaluation for localization, phase and orientation in the synthetic image of Figure 1. The localization error is expressed in pixels, whereas the orientation and phase errors are in radians.

	localization		orientation		phase	
	avg	std	avg	std	avg	std
Gabor	0.067	0.026	0.021	0.007	0.025	0.005
s4	0.072	0.027	0.022	0.008	0.032	0.006
s2	0.076	0.017	0.042	0.011	0.340	0.203
SQF	0.124	0.062	0.026	0.021	0.198	0.092

Binocular disparity. The *tsukuba*, *sawtooth* and *venus* stereo-pairs from the Middlebury stereo vision page (Scharstein and Szeliski, 2002) are used in the evaluation. Since we are interested in the precision of the filters we do not use the integer-based measures proposed there but instead compute the mean and standard deviation of the absolute disparity error. To prevent outliers to distort the results, the error is evaluated only at regions that are textured, non-occluded and continuous. The best results (see Table 3)are obtained with the Gabor filters. Slightly worse are the results with 4th-order steerable filters and the 2nd-order filters yield results about twice as bad as the 4th-order filters. The results obtained with SQFs are comparable with those obtained by the 2nd-order steerable filters. Figure 2 contains the left images of the stereo-pairs, the ground truth depth maps, and the depth maps obtained with the Gabor filters.

Optic flow. We have evaluated the different filters with respect to optic flow estimation on the *diverging tree* and *yosemite* sequences from (Barron et al., 1994), using the error measures presented there. The

Table 3: Average and standard deviation of the absolute errors in the disparity estimates (in pixels).

	tsukuba		sawtooth		venus	
	avg	std	avg	std	avg	std
Gabor	0.32	0.61	0.41	1.26	0.25	0.77
s4	0.36	0.68	0.50	1.86	0.40	1.30
s2	0.47	0.79	1.12	2.50	0.98	2.44
SQF	0.46	0.85	0.93	2.20	0.95	2.40

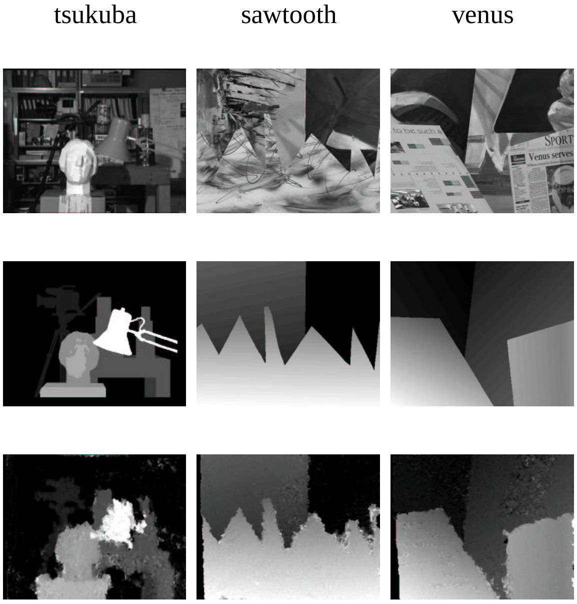


Figure 2: Left frame (top row), ground truth disparity (middle row), and estimated disparity using Gabor filters (bottom row).

cloud region was excluded from the *yosemite* sequence. The results are presented in Table 4 and similar conclusions can be drawn as in the previous Section. Gabor and 4th-order steerable filters yield comparable results whereas 2nd-order steerable filters score about twice as bad. The results obtained by SQFs are slightly worse, since the resulting optic flow have larger errors but a higher density. Figure 3 shows the center images, ground truth optic flow fields, and the optic flow fields computed with the Gabor filters.

Table 4: Average and standard deviation of the optic flow errors (in pixels) and optic flow density (in percent).

	diverging tree			yosemite (no cloud)		
	avg	std	dens	avg	std	dens
Gabor	2.05	2.28	95.6	2.15	3.12	81.8
s4	2.39	2.62	93.2	2.96	4.46	85.0
s2	4.20	4.58	90.6	6.51	9.23	81.9
SQF	12.9	13.4	95.1	18.7	17.8	99.1

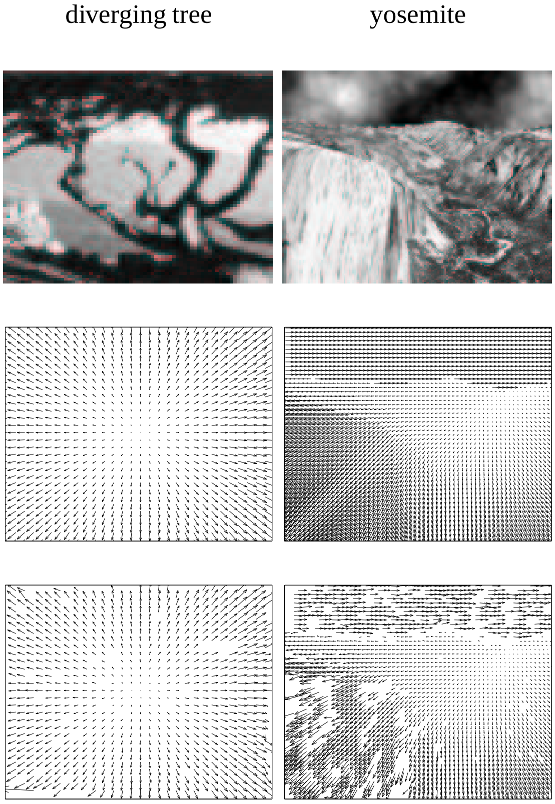


Figure 3: Center frame (top row), ground truth optic flow (middle row) and estimated optic flow obtained with Gabor filters (bottom row). All optic flow fields have been scaled and subsampled five times.

Motion-in-depth. Since binocular test sequences with the ground truth and a sufficiently high frame rate are not available, it has not been possible to make quantitative comparisons. However, considering that motion-in-depth is a ‘derived’ quantity, we expected, that the multichannel anisotropic filtering has the same advantages over isotropic filtering alike those observed for stereo and motion processing. Qualitative results obtained in real-world sequences preliminarily confirmed this conclusion.

5 CONCLUSIONS

Early vision processing can be reconducted to measuring the amount of a particular type of local structure with respect to a specific representation space. The choice for an early selection of features by adopying thresholding procedures, which depend on a specific and restricted environmental context, limits the possibility of building, on the ground of such representations, an artificial vision system with complex functionalities. Hence, it is more convenient to base further perceptual processes on a more general representation of the visual signal. The harmonic representation discussed in this paper is a reasonable representation of early vision process since it allows for an efficient and complete representation of (spatially and temporally) *localized* structures. It is characterized by: (1) compactness (i.e., minimal uncertainty of the band-pass channel); (2) coverage of the frequency domain; (3) robust correspondence between the harmonic descriptors and the perceptual 'substances' in the various modalities (edge, motion and stereo). Through a systematic analysis we investigated the advantages of anisotropic *vs* isotropic filtering approaches for a complete harmonic description of the visual signal. We observed that it is preferable to construct a multichannel, multiorientation representation, thus avoiding an "early condensation" of basic features. The harmonic content is then combined in the phase-orientation space at the final stage, only, to come up with the ultimate perceptual decisions. An analysis of possible advantages of the aggregation of the information in the monogenic image in mid- and high-level perceptual tasks (e.g., image classification) would require further investigation, and it is deferred to a future work.

ACKNOWLEDGEMENTS

This work results from a cross-collaborative effort within the EU Project IST-FET-16276-2 "DrivSco".

REFERENCES

Adelson, E., Anderson, C., Bergen, J., Burt, P., and Ogden, J. (1984). Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41.

Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *Int. J. of Comp. Vision*, 12:43–77.

Bergen, J., Anandan, P., Hanna, K., and Hingorani, R. (1992). Hierarchical model-based motion estimation. In *Proc. ECCV'92*, pages 237–252.

Daugman, J. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Amer. A*, A/2:1160–1169.

Diaz, J., Ros, E., Pelayo, F., Ortigosa, E., and Mota, S. (2006). FPGA based real-time optical-flow system. *IEEE Trans. Circuits and Systems for Video Technology*, 16(2):274–279.

Felsberg, M. and Sommer, G. (2001). The monogenic signal. *IEEE Trans. Signal Processing*, 48:3136–3144.

Fleet, D. and Jepson, A. (1993). Stability of phase information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(12):1253–1268.

Fleet, D., Jepson, A., and Jenkin, M. (1991). Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210.

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *Int. J. of Comp. Vision*, 1:77–104.

Freeman, W. and Adelson, E. (1991). The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906.

Gautama, T. and Van Hulle, M. (2002). A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Networks*, 13(5):1127–1136.

Harris, J. and Watamaniuk, S. N. (1995). Speed discrimination of motion-in-depth using binocular cues. *Vision Research*, 35(7):885–896.

Kehtarnavaz, N. and Gamadia, M. (2005). *Real-Time Image and Video Processing: From Research to Reality*. Morgan & Claypool Publishers.

Koenderink, J. and van Doorn, A. (1987). Representation of local geometry in the visual system. *Biol. Cybern.*, 55:367–375.

Kovesi, P. (1999). Image features from phase congruency. *Videre, MIT Press*, 1(3):1–26.

Krüger, N. and Felsberg, M. (2003). A continuous formulation of intrinsic dimension. In *Proc. British Machine Vision Conference*, Norwich, 9-11 September 2003.

Krüger, N. and Felsberg, M. (2004). An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863.

Marr, D. (1982). *Vision*. New York: Freeman.

Nestares, O., Navarro, R., Portilla, J., and Tabernero, A. (1998). Efficient spatial-domain implementation of a multiscale image representation based on Gabor functions. *J. of Electronic Imaging*, 7(1):166–173.

Owens, R. (1994). Feature-free images. *Pattern Recognition Letters*, 15:35–44.

Pauwels, K. and Van Hulle, M. (2006). Optic flow from unstable sequences containing unconstrained scenes through local velocity constancy maximization. In *Proc. British Machine Vision Conference*, Edinburgh, 4-7 September 2006.

Sabatini, S., Solari, F., Cavalleri, P., and Bisio, G. (2003). Phase-based binocular perception of motion in depth: Cortical-like operators and analog VLSI architectures. *EURASIP J. on Applied Signal Proc.*, 7:690–702.

Sanger, T. (1988). Stereo disparity computation using Gabor filters. *Biol. Cybern.*, 59:405–418.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. of Comp. Vision*, 47(1–3):7–42.

Solari, F., Sabatini, S., and Bisio, G. (2001). Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Elect. Letters*, 37:1382–1383.

Appendix B

Filter design techniques. Technical report.

DRIVSCO TECHNICAL REPORT TR2

Analysis of different filters design approaches for the Drivsco system

First version: Javier Díaz, spa. March 27, 2006.
Second version: Karl Pauwels, bel. March 29, 2006.
Third version: Silvio Sabatini,ita. May 3, 2006.
Fourth version: Javier Díaz, spa. June 27, 2006.
Fifth version: Javier Díaz, spa. January 19, 2007.

1. Overview and general requirements

In the framework of the E.U. project DRIVSCO, we are designing a real-time vision system for complex scene understanding. The system uses a dense phase-based stereo and optical flow module developed by Bel (see also [9]). Furthermore, local phase information as well as local orientation are used by other partners such as Den and Ita, to come up with higher visual descriptors and to compute motion-in-depth

In general, spatially localized phase measures can be obtained by filtering operations with complex-valued band-pass quadrature filters:

$$h(x;u_{peak},\sigma) = h_C(x;u_{peak},\sigma) + ih_S(x;u_{peak},\sigma) \qquad (1)$$

where u_{peak} is the peak frequency of the filter and σ determines its spatial extension. The resulting convolution with the image signal I (complex-valued analytic image) can be expressed as shown in equation (2):

$$Q(x) = I * h(x) = \rho(x)\exp\left(i\phi(x)\right) = C(x) + iS(x) \qquad (2)$$

where $\rho(x)$ and $\phi(x)$ denote their amplitude and phase components, and C(x) and S(x) are the responses of the quadrature filter pair.

Although, in principle, phase-based techniques are robust against typical variations in image formation (e.g., brightness, contrast, affine deformations, etc...), the estimation of phase is intrinsically noisy and depends critically on the choice of the quadrature filters.

In this document we summarize the filters types we have considered and include the basic designing stages.

We consider three kinds of filters:

1. Isotropic analytic filters (based on the monogenic signal).
2. Gabor filters.
3. Steerable filters.

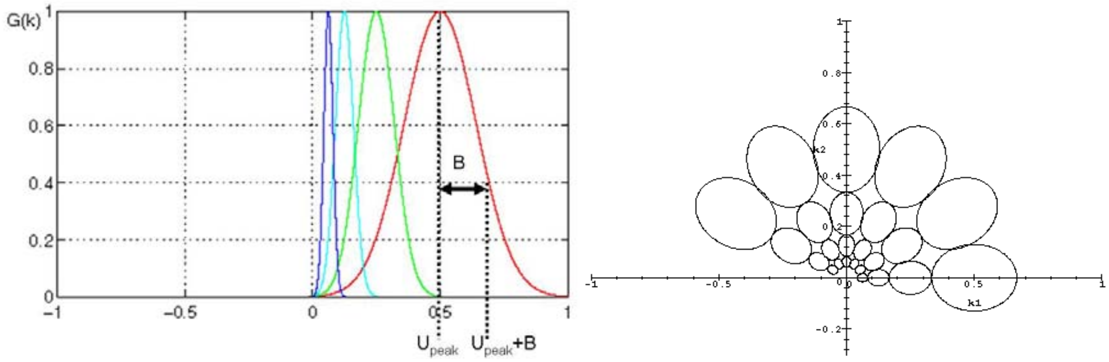


Figure 1. Bandpass filters covering different spatial frequencies and orientations (figures adapted from [10]. Left image represents the different filters spatial scales based on scaling by 2 of the main filter. The x-axe represents the normalized frequency values (f/$f_{Nyquist}$). Right image shows a polar representation of these scales across different orientations, using a logarithmic splitting of the frequency domain. Uniform coverage of the frequency domain allows properly decomposing the image signal on this domain and extracting multivalued local phase and energy information.

There are several conditions to fulfil due to implementation issues (see Figure 1 to easily identify parameters meanings):

1. Nyquist sampling condition: Using pixels as units, the sampled period is 1 pixel, which corresponds to 1 pixel^{-1} sampling frequency. The maximum bandwidth of the filter to avoid aliasing is 0.5 pixels^{-1}. Given B the filter bandwidth (defined at the cut-off frequency corresponding to half of the base-band amplitude spectrum), the maximum peak frequency of the filter can be derived from the following equation:

$$u_{peak} + B < 0.5 \qquad (3)$$

It is worthy to note that, since all the filters considered in this report are not bandlimited, some aliasing will occur regardless of the sampling density. In other words, by setting, by defining, a filter bandwidth we decide how much aliasing we tolerate.

2. Multiscale frequency space coverage. The distance between neighboring frequency "channels" is determined by the spatial frequency bandwidth. Since we work with multiscale representation based on power of two, the minimum B to cover the frequency domain without holes is:

$$B>=u_{peak}/3 \tag{4}$$

3. Uniform orientation coverage condition (only for Gabor and Steerable filters). Because we should cover the 2-D frequency domain for the different orientations, we need to consider a minimum number of oriented filters. This number depends on the filter bandwidth and is related to the desired orientation sensitivity of the filter.
   Accordingly, we can estimate the desired bandwidth using equation (5) as in [2]:

$$2\pi u_{peak} <= N_{orientation} * 2B \tag{5}$$

   and define the orientation bandwidth in the frequency domain as in equation (6) (cf. [9])[1]:

$$B_{\theta} = \tan^{-1}(B / u_{peak}) \tag{6}$$

   Spatial frequency bandwidths are constant in octaves, and orientation bandwidths are constant in degrees, but there is freedom to choose the absolute magnitudes of these bandwidths (provided that they respect , see condition 1).

4. The extent of the filter should not exceed the number of taps. For Gabor we check if the number of taps is larger than four times the spatial standard deviation.

2. Isotropic analytic filters

In general, direct extensions of the analytical-signal / Hilbert transform to two-dimensional signals is not straightforward, since they do not satisfy the isotropy property, which is necessary to obtain invariance with respect to orientation.

[1] We are considering 2D spatial filters with a circular support. We should perhaps compare system's performance with anisotropic filter envelopes (sigma_y not equal to sigma_x). (This would not be possible for Steerable filters).

Alternatively, the *Monogenic signal* has been introduced as a 2D isotropic analytic signal, based on the Riesz transform, which is used instead of the Hilbert transform [1]. Such transform is based on the generalization of the 1-D concept of phase for 2-D signals. Orientation is used as disambiguation information to extend this concept (see Figure).

The basic idea is to design an odd isotropic filter that is vector-valued rather than scalar-valued, and the resulting complex-valued monogenic image can be expressed as:

$$Q_M(x,y) = I * h_C(x,y) + [i,j]\cdot[I * h_{S1}(x,y), I * h_{S2}(x,y)]^T = C(x,y) + [i,j]\cdot[S_1(x,y), S_2(x,y)]^T \quad (7)$$

The image is convolved with a radial filter and two 2-D non-separable filters that allow the estimation of the phase, and magnitude of the signal for the main orientation (also computed).

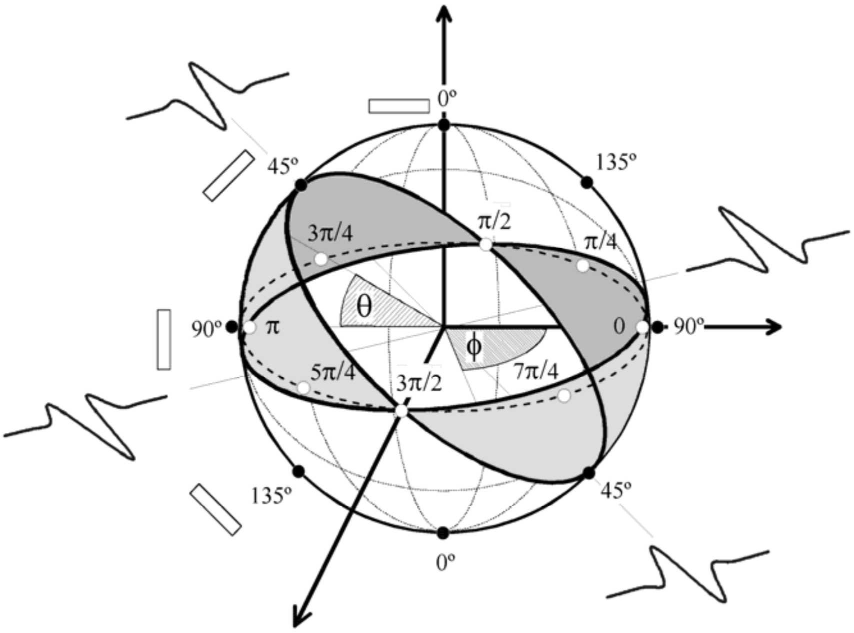


Figure 2. Orientation-phase sphere.

From the filters outputs, image features are computed as:

Local amplitude: $A = \sqrt{C^2 + S_1^2 + S_2^2}$ (8.a)

Local phase: $\phi = \arctan\left(\frac{\sqrt{S_1^2 + S_2^2}}{C}\right)$ (with sign correction, see below) (8.b)

Local orientation: $\theta = \arctan\left(\frac{S_2}{S_1}\right)$ (8.c)

Because only the information of the main direction is provided, this approach is only recommended for 1-D signals as edges or lines.2.1 Filter Implementation

We have applied this transform to the Difference of Poissons (DOP) functions. The resulting equations are:

a. Radial filter:

$$h_e(x) = \frac{s_1}{2\pi\left(x^2+y^2+s_1^2\right)^{3/2}} - \frac{s_2}{2\pi\left(x^2+y^2+s_2^2\right)^{3/2}} \tag{9.a}$$

$$H_e(u) = \exp(-2\pi|u|s_1) - \exp(-2\pi|u|s_2) \tag{9.b}$$

b. X-Y filters (real and imaginary parts)

$$h_o(x) = \frac{x+iy}{2\pi\left(x^2+y^2+s_1^2\right)^{3/2}} - \frac{x+iy}{2\pi\left(x^2+y^2+s_2^2\right)^{3/2}} \tag{10.a}$$

$$H_o(u) = \frac{y-ix}{|u|}\left[\exp(-2\pi|u|s_1) - \exp(-2\pi|u|s2)\right] \tag{10.b}$$

c. Systems relations

1. Filters parameters relation:

   $s_2=\lambda^k \bullet s_1$, according to [1], pp. 95. We consider k=1, $\lambda=2$. $s_2=2\bullet s_1$

2. Peak frequency:

$$u_{peak} = \frac{1}{2\pi\left(s_2 - s_1\right)}\ln\left(s_2 \big/ s_1\right) \tag{11}$$

3. Poisson bandwidth:

$$\frac{M(u_{peak})}{M(u_c)} = 2 \quad \Rightarrow u_c = \frac{\ln(2)}{2\pi s} \tag{12}$$

4. DOP bandwidth: Just for using $s_2=2\bullet s_1$, then we obtain:

$$\frac{M(u_{peak})}{M(u_c)} = 2 \quad \Rightarrow x^2 - x + M(u_{peak})/2 = 0, \quad x = \exp(-2\pi s_1 u)$$

$$B = \frac{1}{4\pi s_1}\left[\ln\frac{2}{1-\sqrt{1-2M(u_{peak})}} - \ln\frac{2}{1+\sqrt{1-2M(u_{peak})}}\right] \tag{13}$$

5. Spatial window extension [-s$_2$, s$_2$] for windowing.

6. DOP energy: ([1], pp. 181)

$$E=8\,\pi\,s_2 \qquad (14)$$

d. Implementation file: ***function[bp, cbp]=create_DOP(s1,s2,Ecut)*****,** note that s1 and s2 should be integer numbers to avoid interpolation problems. Basically the parameters to fix are s1 and s2. We are constrained by the DOP properties that not allow us to tuning high frequencies (wide filters compared with the others).

(**NOTE:** to make fairer the comparison with the Gabor filters, an even Gabor filter with radial symmetry should be used. The associated X-Y filters (not available analytically) would be computed numerically by FFT.)

a. Radial filter:

$$h_C(x,y) = const * \exp\left\{\frac{-(x^2+y^2)}{2\sigma^2}\right\} * \exp\left\{i2\pi u_0\left(x^2+y^2\right)^{1/2}\right\} \qquad (15.a)$$

$$H_C(u,v) = \exp\left[-\frac{\sigma^2}{2}\left(\sqrt{u^2+v^2}-u_0\right)^2\right] \qquad (15.b)$$

b. X-Y filters:

$$h_{S1}(x,y) = -\frac{x}{2\pi\left(x^2+y^2\right)^{3/2}} \otimes h_C(x,y) \qquad (16.a)$$

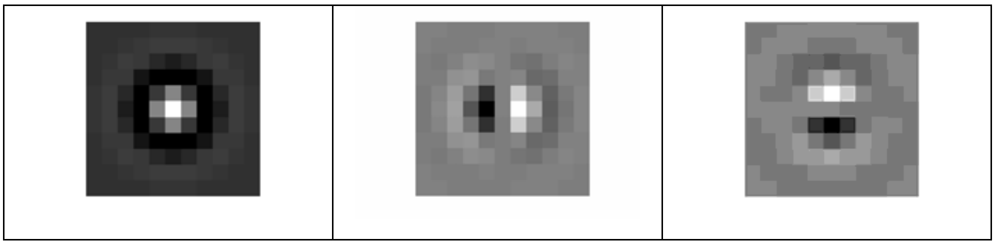
$$h_{S2}(x,y) = -\frac{y}{2\pi\left(x^2+y^2\right)^{3/2}} \otimes h_C(x,y) \qquad (16.b)$$

$$H_{S1}(u,v) = i\frac{u}{|u|}H_C(u,v) \qquad (16.c)$$

$$H_{S2}(u,v) = j\frac{v}{|u|}H_C(u,v) \qquad (16.d)$$

where ⊗ is the convolution operator.

Starting from the spectrum of a radially-symmetric Gabor filter (as defined in Section 4) the corresponding 11 tap spatial filters are shown below (it is worthy to note that the radially symmetric filter (hc) does not integrate to zero, hence a procedure similar to that presented in Section 5.1 should be applied).



hc	*hs1*	*hs2*

Figure 3. Monogenic signal filters numerically computed from a even Gabor filter with radial symmetry.

The corresponding frequency representation for three scales is represented below:

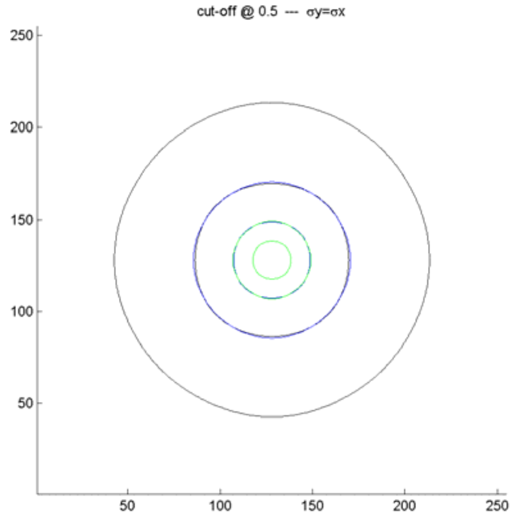


Figure 4. Monogenic signal frequency representation for several spatial scales (filters based on an even Gabor filter with radial symmetry).

It is worthy to note that, by definition, the band-pass channels associated with the filters derived by the monogenic signal are isotropic. Orientation parameterised families of filter responses can be obtained through the Radon transform, by projecting the spectrum energy onto a line with orientation θ.

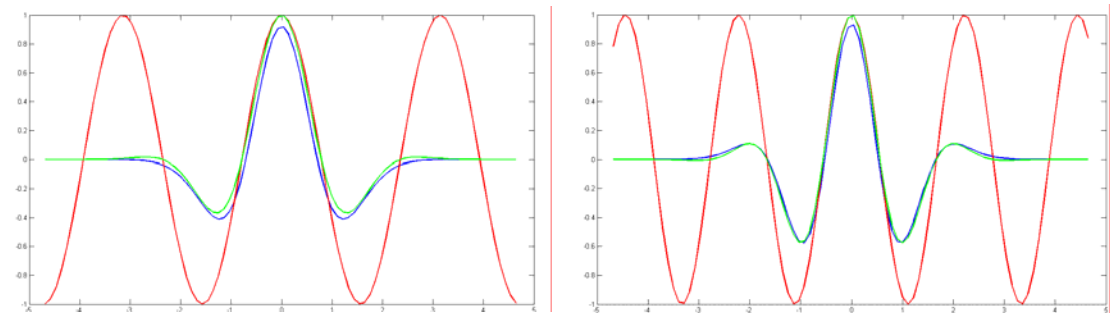
If we use non separable filters such as the presented in this document, they requires high computational resources. Computing K non separable 2-D convolutions of N taps has a complexity:

$$O_{no\text{-}sep}(K,N)=K*N^{2} \textit{ multiplications } + K*(N^{2}-1) \textit{ additions} \qquad (17)$$

It means that the computing resources for this approach using a 11x11 taps filter implementation requires 363 multiplications and 360 additions. It is then the more expensive approach and, its implementation only can be justified for the sake of accuracy.

3. Steerable filters based on Gaussian derivatives

Widely used on the literature, [4], [5], [6]. They allow the computation of an oriented filter based on basic separable set of convolution kernels that are properly weighted to get the desired oriented kernel. A key factor for its design is the derivative order; its properties are quite similar to Gabor for derivative order 4 or larger as demonstrated Figure 5.



(a) **(b)**

Figure 5. Comparison of Gabor (green), Gaussian derivatives (blue) and cosine (red) functions tuned to the same peak frequency. (a) Gaussian derivative of order 2 is close to Gabor filters but the difference is not completely negligible. (b) Gaussian derivative of order 4. This time the similarity is larger and the filter is very close to the Gabor approach. Note that the number of waves increase according to the Gaussian derivative order, corresponding to higher orientation selectivity.

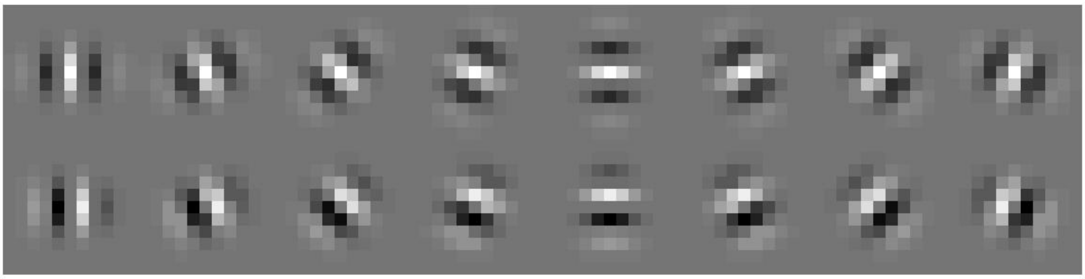


Figure 6. Example 4th order filters for u_{peak}=1/4, 8 orientations and using 11 taps are shown below.

For instance, the Second order Gaussian derivatives G_{xx}, G_{xy} and G_{yy} and their Hilbert transforms H_{xx}, H_{xy}, H_{yx} and H_{yy} are represented Figure 7. Their equations are the following:

$$G_{xx} = 0.9213 \cdot \left(2x^2 - 1\right) \cdot e^{-\left(x^2+y^2\right)} \tag{18.a}$$

$$G_{xy} = 1.843 \cdot x \cdot y \cdot \left(2x^2 - 1\right) \cdot e^{-\left(x^2+y^2\right)} \tag{18.b}$$

$$G_{xx} = 0.9213 \cdot \left(2y^2 - 1\right) \cdot e^{-\left(x^2+y^2\right)} \tag{18.c}$$

And for their Hilbert Transform:

$$H_{xx} = 0.9780 \cdot \left(-2.254x + x^3\right) \cdot e^{-\left(x^2+y^2\right)} \quad (19.a)$$

$$H_{xy} = 0.9780 \cdot \left(-0.7515 + x^2\right) \cdot \left(y\right) \cdot e^{-\left(x^2+y^2\right)} \quad (19.b)$$

$$H_{yx} = 0.9780 \cdot \left(-0.7515 + y^2\right) \cdot \left(x\right) \cdot e^{-\left(x^2+y^2\right)} \quad (19.c)$$

$$H_{yy} = 0.9780 \cdot \left(-2.254y + y^3\right) \cdot e^{-\left(x^2+y^2\right)} \quad (19.d)$$

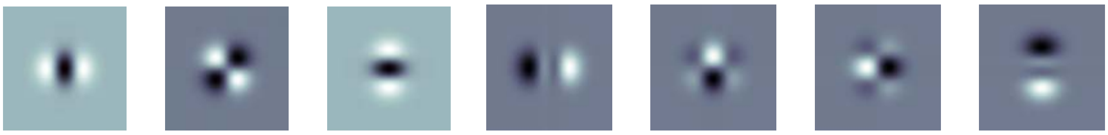


Figure 7. Second order Gaussian derivatives separable base set. The three first filters (from left to right) are the Gaussian derivatives and their Hilbert transforms are the four filters showed on their right. With this set of filters, we can estimate the output at any orientation just combining linearly the base set output. This allows building oriented quadrature filters banks as shows in Figure 2.3 but at any possible orientation.

The equations for the fourth order can be taken from [6]. The corresponding multichannel frequential representation (“rosettelike” or “daisy” diagram) is shown below (for three different scales) at figure 8:

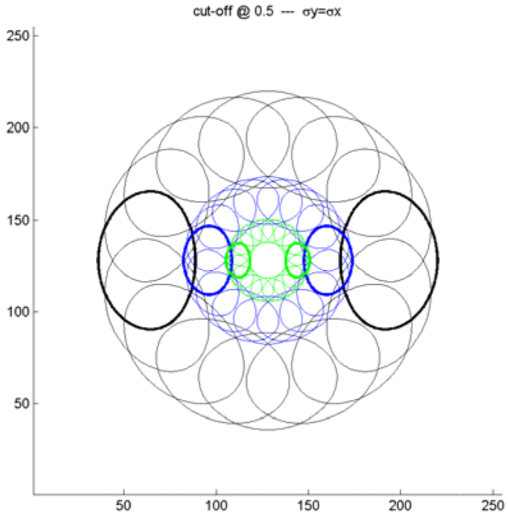


Figure 8. Multichannel frequential representation for the Fourth Order Gaussian derivative filters.

It is worthy to say that the orientation bandwidth is larger than that obtained with Gabor filters, thus resulting in a broad tuning for local orientation.

3.1 Filter implementation

a. The well known equations of a 1-D Gaussian and its derivatives are:

$$g_0(x) = e^{-\frac{x^2}{2\sigma^2}} \qquad g_n(x) = \frac{d^n}{dx^n} g_0(x) = P_{n,\sigma}(x) g_0(x) \tag{20}$$

This equation indicates that the n-th derivative of a Gaussian can be written as the product of a polynomial (generalized Hermite polynomial) by the original Gaussian. In the frequency domain Equation (10) can be expressed as:

$$G_0(f) = \sigma e^{-\frac{(2\pi\sigma)^2 f^2}{2}} \qquad G_n(f) = (j2\pi f)^n G_0(f) \tag{21}$$

Since our main concern is on the phase-based approaches, we need the quadrature pair of these filters. It can be obtained by using it's the Hilbert transform as described in [6]:

b. Spatial window extension [-2σ, 2σ] for windowing.

c. Variance-bandwidth relation. From [4] we get the asymptotic bandwidth:

$$B \to \frac{1}{4\pi\sqrt{2}\sigma} \tag{22}$$

d. The peak frequency is computed by derivation in the frequency domain as in [5]. From this we get the value presented at equation (23).

$$u_{peak} = \frac{1}{2\pi}\sqrt{n/\sigma^2} \tag{23}$$

e. Design steps:

1. Fix u_{peak}=1/4.
2. Fix number of taps=11.
3. Fix numbers of orientations.
4. Fix the derivative order=4. With these parameters we obtain B and σ.
5. Checking the conditions of section 1 to verify that they are satisfied.

f. Implementation file: ***function[??]=design_Steer_filter(??).***

Concerning the derivative order, we will study the properties of the 2 and 4 orders in Sections 2.3 and 2.4. The number of kernels to compute the oriented output of the filters *k*, depend on their derivative order *n*. We need *k'=2n+3* separable 2-D kernels or *k=4n+6* 1-D kernels.

The complexity for computing K separable convolutions using a kernel of N taps is given by equation (24):

$$O_{sep}(K,N)=K*N \textit{ multiplications } + K* N\textit{-1 additions} \tag{24}$$

This represent that the complexities for computing these convolutions depending on the derivative order and they are:

- n=2, k=14, 153 multiplications + 140 additions.
- n=3, k=18, 198 multiplications + 180 additions.
- n=4, k=22, 242 multiplications + 220 additions.

4. Gabor filters

Perhaps the most widely filters used in the literature, Gabor filters are defined by harmonic functions modulated by a Gaussian envelope. They main property is that they minimize the space-frequency uncertainty. An efficient filter implementation could be find in [3]. Please note that oriented Gabor filters are not x-y separable, but they can be computed as sums of separable filters as described in [3].

The bandwidth used in [3] equals $u_{peak}/3$ or 1 octave, which is the smallest one to properly cover scale space.

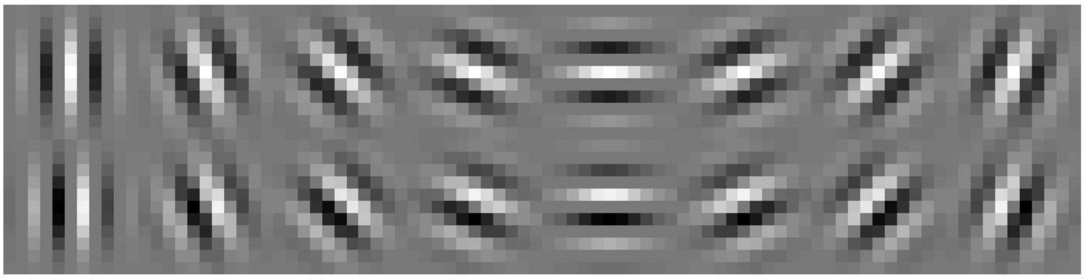


Figure 9. Resulting Gabor filters for u_{peak}=1/4 using 11 taps.

In [3], only 4 orientations are supported. We have extended this scheme to 8 orientations. By exploiting the symmetry, all 8 even and odd filters can be constructed on the basis of 24 1D convolutions. The block diagram from Figure 5 of [3] has been extended to include the orientations in between the horizontal, vertical and diagonal as shown in Figure 10.

Each block corresponds to one 1D convolution, so only 12 convolutions are required to compute even and odd responses for these 4 `in-between' orientations.

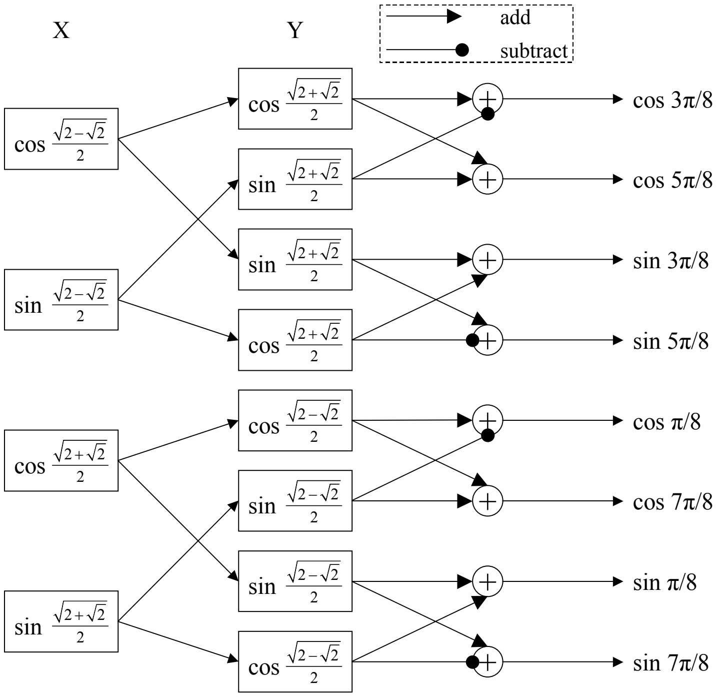

X
Y
add
subtract
cos $\frac{\sqrt{2-\sqrt{2}}}{2}$
sin $\frac{\sqrt{2-\sqrt{2}}}{2}$
cos $\frac{\sqrt{2+\sqrt{2}}}{2}$
sin $\frac{\sqrt{2+\sqrt{2}}}{2}$
cos 3π/8
cos 5π/8
sin 3π/8
sin 5π/8
cos π/8
cos 7π/8
sin π/8
sin 7π/8

Figure 10. Block diagram for extension of 4 orientation (from [3]) to a total of 8 orientations.

In each block, cos or sin refers to an even or odd gabor respectively. The frequency of these elementary Gabor filters equals the peak frequency multiplied by the factor in the block.

The corresponding multichannel frequential representation ("rosettelike" or "daisy" diagram) is shown below (for three different scales):

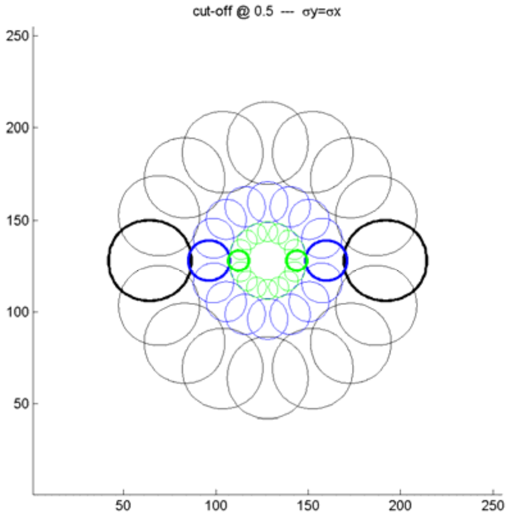


Figure 11. Multichannel frequential representation for Gabor filters. Note that at difference with Gaussian derivatives, this approach present circular symmetry around the peak frequency.

4.1 Filter implementation

a. Basic equation for even and odd filters, with u_{peak} the peak frequency, θ the main orientation and σ the Gaussian variance.

$$G(x, y, \theta) = \exp\left\{\frac{-(x^2 + y^2)}{2\sigma^2}\right\} * \exp\left\{i2\pi U_{peak}\left(x\cos(\theta) + ysen(\theta)\right)\right\} \tag{25}$$

b. Spatial window extension [-2σ, 2σ] for windowing.

c. Variance-bandwidth relation. The bandwidth of the Gabor filters is equal to the bandwidth of its associate Gaussian. It is:

$$\frac{\hat{G}(0)}{\hat{G}(B)} = 2 \quad \Rightarrow B = \frac{\sqrt{2 * \ln(2)}}{2\pi\,\sigma} \tag{26}$$

where $\hat{G}$ is the Fourier transform of G

d. Design steps:

1. Fix u_{peak}=1/4.

2. Fix number of taps=11.

3. Fix numbers of orientations. Conditions 2 and 3 are related, and then we should check the conditions of section 1 to verify that they are satisfied.

4. Compute Gaussian variance. It is done by given a desired bandwidth or using computing the value such as the Gabor and Gaussian derivative have the same bandwidth. This relation is:

$$B_{Gabor} = B_{Steerable} \Rightarrow \sigma_{Gabor} = 2\sqrt{\ln 2}\,\sigma_{Steerabler} \qquad (26)$$

e. Implementation file: ***function[??]=design_Gabor_filter(??)*.**

According to equation 24, the Gabor filter bank, using 24 total filters (odd and even) of 11 taps each of them requires 264 multiplications and 240 additions.

5. Filter Implementation issues

This section discusses some technical details required in the practical implementation of the filters.

5.1. Quadrature constraint

In general, due to the sampling and windowing operations, a bandpass filter pair is not in quadrature as it has non-zero negative frequencies, and a non-zero DC component (even filters do not integrate to zero)[2]. This degrades the system outputs and a DC removal procedures must be used.

In Monogenic filters, it can be eliminated using normalization over the two Poissons to make them equal weight after sampling.

For the Gaussian derivatives and Gabor filters, several approaches are possible:

1. Convolve the image with a kernel: I-I_{mean} where the mean value is computed on a square window of size [-σ, σ].
2. Convolve the kernel itself with a kernel (-1 2 1)/4 [7] padding zeros on the filters and cut to get the original size.
3. Numerical optimization. Using the function ***fmincon*** of the Matlab optimization toolbox. The idea is to get zero DC without degrade the oriented filters.
    a. Using this procedure, the 1D filters are modified by enforcing zero DC on all 2D filters in which they take part and minimizing the difference with the theoretical 2D filters.

Specific care should be paid to to adjust the coefficients of each filter function so that the even and odd symmetry is respected.

[2] . In order that the even and odd filters approximate a Hilbert transform pair, the amplitude of their spatial frequency response should be negligible for negative frequencies; e.g., the power spectrum should fall below several dBs from peak value at $f = 0$. It is worthy to note that the negative power and the DC component are functions of the frequency bandwidth (the DC component reduces as the number of oscillation periods increases).

5.2. Filters values quantization

The real-time implementation using customized hardware requires integer data representation from [-255.255] (9 bits signed data). We need to evaluate the two strategies: scaling and rounding data or integer optimization.

For the Gabor filters the same strategy as Nestares has been implemented. Nestares removes DC-component by optimizing (Section 5.1 3), multiplying by 256 and rounding, in case a DC-component remains in the 1D filters after rounding, a correction is made :

- if DC is one, in the central point of the filter
- if DC is two, symmetrical at location with largest rounding error
- other options not yet implemented

5.3. Energy symmetry for odd and even filters

The sampling and windowing operation could unbalance the energy of the odd and even filters. We need to rescale its energy values to have the same energy.

Similar to DC-removal, energy balancing could be included as a constraint in the optimisation (Section 5.1 3). However, this can no longer be guaranteed after the integer rounding process.

Two alternative normalisation conditions have been used most frequently in the literature:

(1) maximum condition ($\max_f \hat{G}(f) = 1$); for 1D Gabor function the corresponding normalization constant is $\frac{1}{\sqrt{2\pi}\sigma}$

(2) constant energy condition ($\int_{-\infty}^{+\infty} G^*(x)G(x)dx = 1$) for 1D Gabor function the corresponding normalization constant is $(\sqrt{\pi}\sigma)^{-1/2}$

We would suggest to use the second condition.

6. Local Phase and orientation based on a oriented bank of filters.

The monogenic signal extracts directly this information for the dominant orientation using the equations:

$$E_{local} = \sqrt{C^2 + S_1^2 + S_2^2} \tag{28.a}$$

$$\theta_{local} = \arg\left(\vec{S}\right) \quad \mathrm{mod}(\pi) \tag{28.b}$$

$$\phi_{local} = sign\left(\zeta\left\{\vec{S}\right\}\right) * \arg\left(C + i\left|\vec{S}\right|\right), \quad \zeta\left\{\vec{S}\right\} = S_1 * \cos\theta + S_2 * \sin\theta, \quad \vec{S} = (S_1, S_2) \tag{28.c}$$

Gabor and Steerable filters provide a much more rich information because information of several orientations are considered, allowing the description of complex structures such as textures where several orientations are available. If we consider N orientations, we will note E$_i$ and ϕ_i to the energy and phase of the filter oriented with angle θ$_i$=i*π/N. If only the dominant orientation information is required, there are several methods for its estimation:

1. Winner- take-all. We will take for each pixel the phase, energy and orientation of the filter with maximum energy.

$$E_{local} = E_{\max} \quad \phi_{local} = \phi_{E_{\max}} \quad \theta_{local} = \theta_{E_{\max}} \tag{29}$$

2. Weighted-average: (we consider linear case, though the energy can be power to different orders).

$$E_{local} = \frac{\sum_i E_i}{N} \quad \phi_{local} = \frac{\sum_i E_i \phi_i}{\sum_i E_i} \quad \theta_{local} = \frac{\sum_i E_i \theta_i}{\sum_i E_i} \tag{30}$$

3. Some-winner-take-all: only values over a threshold (typically the mean) are considered:

$$E_{local} = \frac{\sum_{i\in\Omega} E_i}{N} \quad \phi_{local} = \frac{\sum_{i\in\Omega} E_i \phi_i}{\sum_{i\in\Omega} E_i} \quad \theta_{local} = \frac{\sum_{i\in\Omega} E_i \theta_i}{\sum_{i\in\Omega} E_i} \quad \Omega = i \,/\, E_i > E_{mean} \tag{31}$$

4. Tensor-based method [7], [8]. Based on a local tensor that projects the different orientations, information can be computed as:

$$E_{local} = \frac{\sum_{i\in\Omega} E_i}{N} \quad \theta_{local} = \frac{1}{2}\arg\left(\sum_i \sqrt{C_i^2 + S_i^2}\, \exp 2\theta_i\right) \quad \phi_{local} = \arctan\left(\frac{\tilde{S}}{\tilde{C}}\right) \tag{32}$$

Where C_i and S_i correspond to the even and odd filters output at orientation i. Three different ways can be used for phase computation which are illustrated in equation 33. The approaches (33.a) and (33.c) coming from [8] and (33.b) from [7]. In our experiment we have used (33.a) because this is the hardware friendly approach. The differences between choices, as indicated by [8], are negligible. Furthermore, in the case of a one-dimensional signal there will be no difference at all between the methods. We will use the first method because is more hardware friendly since it does not suffer from features dependencies (we do not require to compute orientation in advance to estimate the phase).

$$\tilde{C} = \sum_i C_i \quad \tilde{S} = \sum_i S_i \tag{33.a}$$

$$\tilde{C} = \sum_i C_i \left|\cos(\theta_i - \theta_{local})\right| \quad \tilde{S} = \sum_i S_i \cos(\theta_i - \theta_{local}) \tag{33.b}$$

$$\tilde{C} = \sum_i c_i \cos^2(\theta_i - \theta_{local}) \quad \tilde{S} = \sum_i S_i \bullet sign(\cos(\theta_i - \theta_{local})) \bullet \cos^2(\theta_i - \theta_{local}) \tag{33.c}$$

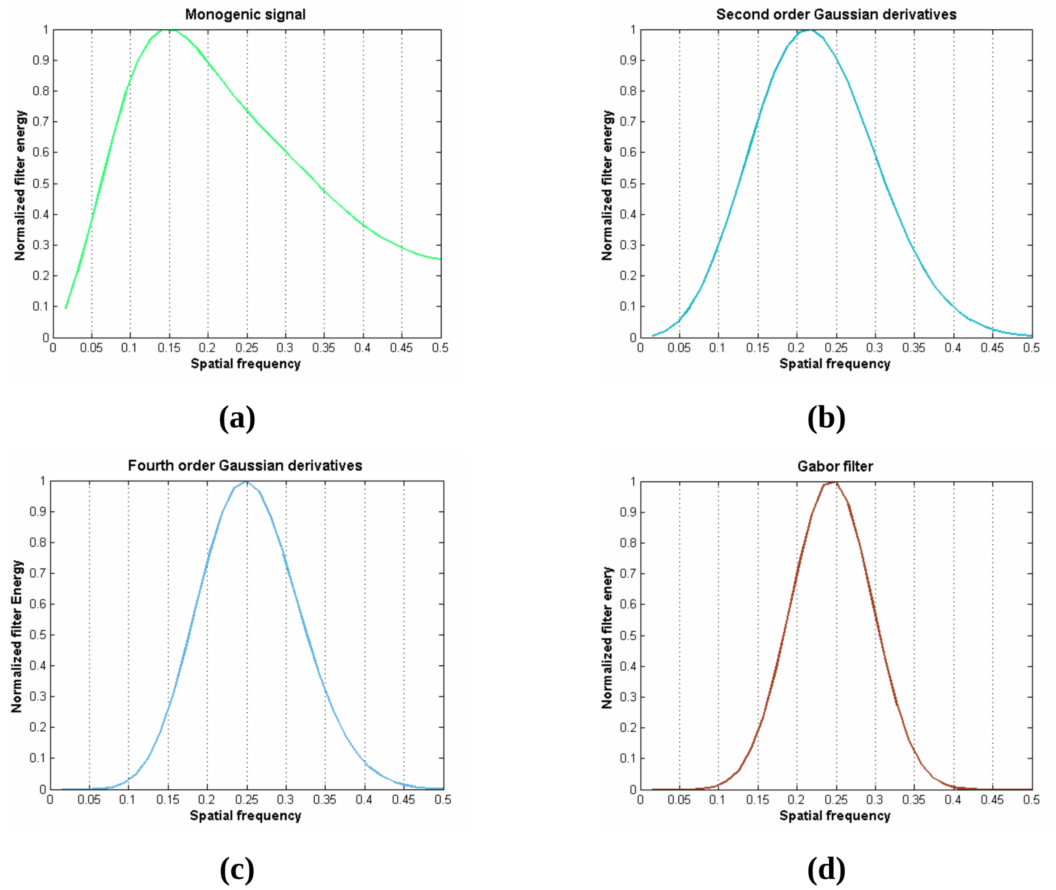
7. Quantitative analysis of the accuracy for feature estimation of the presented approaches.

In Sections 2-4 we study the computing complexity of the different filters and conclude that Gaussian derivatives with low derivative order are the less computational load approach and monogenic signal the most expensive one. Section 5 shows the underlying equations and methods that we can use to estimate the local image features from each filter type. Now we are going to evaluate the accuracy of the different approaches using the equations presented on previous section.

Comparing the different approaches is a hard task to do due to the large number of variables to consider for filtering design. Furthermore, the parameter choice can significantly bias the results, possibly leading to wrong conclusions. Because of that, we will focus on a most affordable task; we will use some fixed filter parameters that exploit each signal type properties. For instance, Gabor and 4-order Gaussian derivatives allow very fine filter tuning capabilities and orientation selectivity. As in [3], our designed filter will have a high frequency of f_0=0.25 pixels^{-1} and bandwidth β= f_0/3=0.083 pixels^{-1}. Monogenic signals and second order Gaussian derivatives have broad bandwidth and therefore, peak frequency should be lower to fulfil equation (2.3). We use f_0=0.21 pixels^{-1} and bandwidth β= 0.1 pixels^{-1} for the Second order Gaussian derivative as in [6]. For the Monogenic signal, the design values are S_1=1 and S_2=2, which gives us the higher frequency filter based on this approach. It gives a peak frequency of f_0=0.11 pixels^{-1} and mean bandwidth β= 0.14 pixels^{-1} (bandwidth curve is not symmetric and therefore we only provide its mean value). All these values have been computed using the equations described on Section 2.2.

In order to test the different approaches, we use two different kinds of signals. First, a set of synthetic sinusoidal gratings with different orientations and spatial scales is used. For this stimulus image features are known and we can numerically test the accuracy of the filters. Second, we also have used real images to get some qualitative results.

From the sinusoidal gratings set, the experimental energy values of the different filters responses across the scales is represented on Figure 12 (note that these are the experimental results, which consider for example quantization problems or finite spatial kernel size). It confirms our numerical bandwidth values and shows that for our design, Gabor filters have the narrowest bandwidth and Monogenic signal the widest one.



(a)

(b)

(c)

(d)

Figure 12. Normalized energy distribution across the spatial frequency scales, experimental results using a

sinusoidal gratins test (x-axe use units on pixels^{-1}) for the Monogenic signal (a), Second order Gaussian derivatives (b), Fourth order Gaussian derivatives (c) and Gabor filters (d). Filters bandwidth decrease from (a) to (d).

Our goal is to compare the different alternatives accuracy taking into account on their hardware implementation feasibility. We have three features to evaluate but we will focus on the local orientation estimation accuracy and bandwidth tuning here.

Local energy is valuable as reference to discriminate areas with low or high contrast and therefore, its numerical value is not important but rather its relative value compared with closer areas. This allows to evalutate the localization properties of the different filters as addressed in [13].

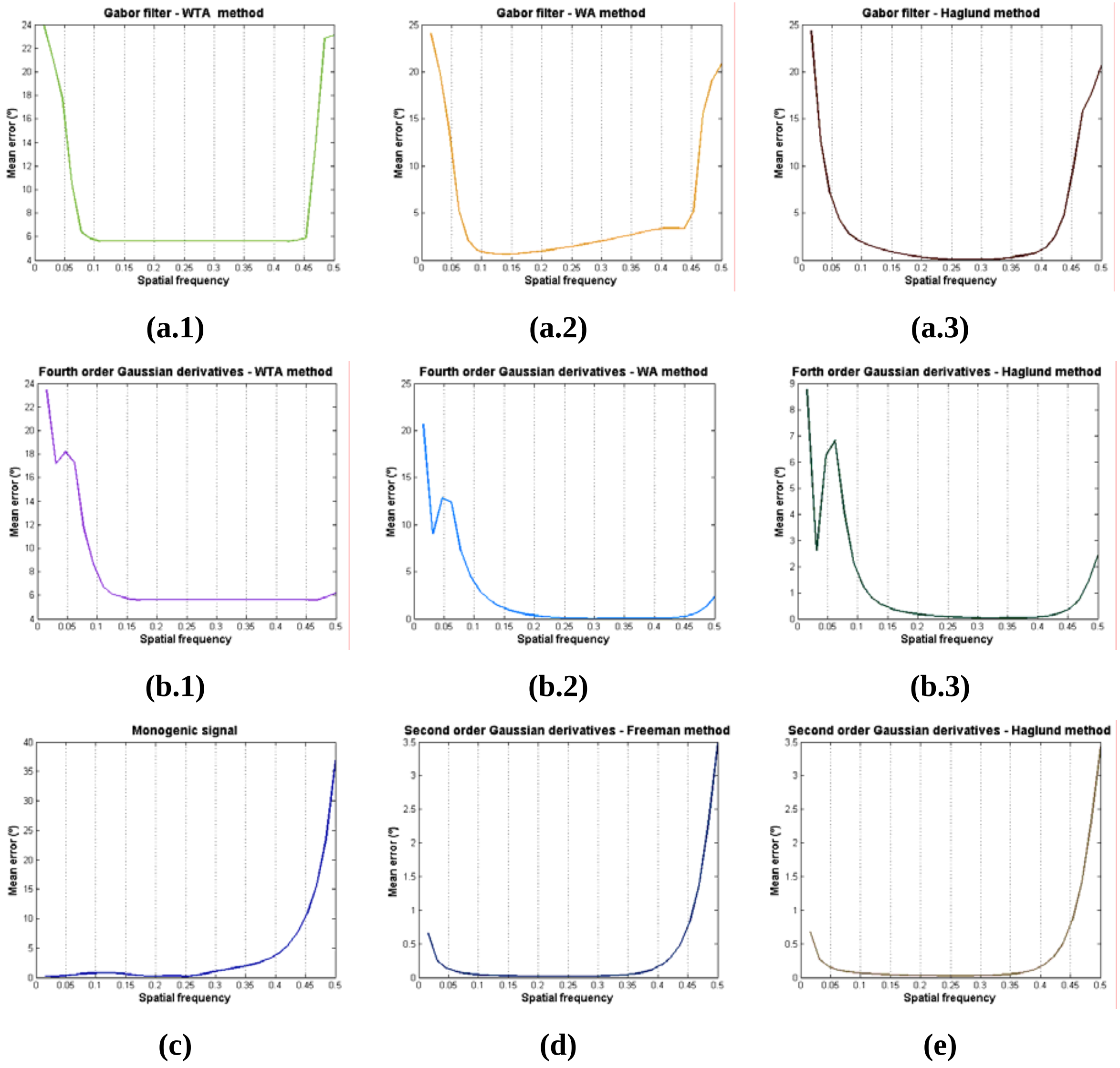
Local orientation is necessary to compute phase and therefore, error or bias on its estimation significantly can degrade the phase accuracy and their study is carried out in this section and in [13].

The phase information is related with the filter shape and therefore numerical evaluation is addressed in the framework of task oriented analysis in [11] and [13].

Given the previous discussion and using the reports [11] and [13] as complementary material, we focus on orientation selectivity to decide between the different approaches.

In the Figure 13 we measure the mean error vs. sinusoidal grating spatial scale. Data outputs are unthresholded and therefore, large errors are not significant if the filter energy value is close to zero. For approaches that need of filter responses interpolation, the tree methods presented in section 2.3, Winner-take-all, weighted-average and Haglund tensor are compared. Several conclusions can be extracted from these figures.

1. The best interpolation method is the Haglund approach (note that we have implemented the interpolation method of equation (33.a) for phase. It produces the smaller error on the filter frequency band.
2. All the filters have high accuracy for orientation estimation, less that 1° of error.
3. The filters that cover the wider range are the Monogenic signals and the second order Gaussian derivatives. This confirms the theoretical analysis in Section 2-4 relative to its bandwidths.
4. For the second order Gaussian derivatives, Haglund tensor approach and the Freeman Fourier series expansion (see [6]) produce equivalent results.



(a.1) **(a.2)** **(a.3)**

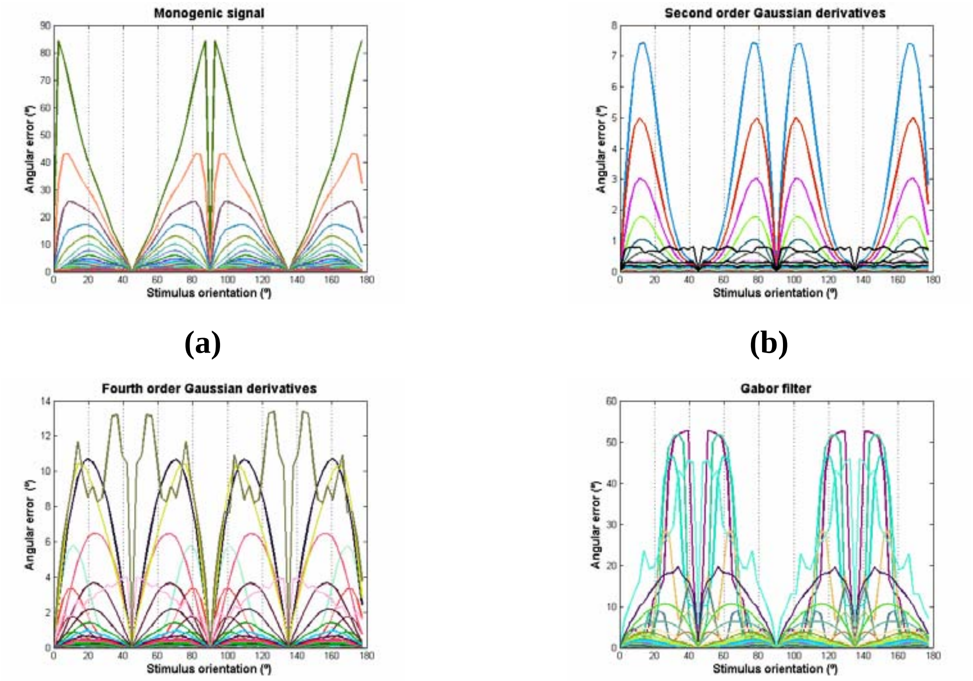
(b.1) **(b.2)** **(b.3)**

(c) **(d)** **(e)**

Figure 13. Mean orientation error measured for each grating spatial frequency. We have used sinusoidal gratings as input, oriented to 64 different angles and with spatial scales from 0.5 pixels^{-1} to 0.0078 pixels^{-1}. Note that, for the sake of clarity, we use different y-axe scales but error values are quite different for the different approaches.(a) Gabor filters are computed at 8 orientations and three interpolation methods are used: (a.1) Winner take all (*WTA)*, (a.2) Weighted average (*WA*) and (a.3) Haglund approach. (b) The same methods are utilized using the Fourth order Gaussian derivatives. (c) Monogenic signal results. (d) Second order Gaussian derivatives, orientation computed using the Freeman and Adelson approach [6]. (e) Second order Gaussian derivatives, orientation computed using eight oriented filter and the Haglund [8] interpolation method.

We also have measure the different filter behaviours against several noise types (multiplicative, Gaussian white noise and salt and pepper). As expected, the error grows approximately linear by all the approaches and therefore, robusness to noise will not drive and affect significantly the decision between the different filters.

The error presented at each orientation and scale is showed in Figure 14, using Haglund interpolation filter approach when required. There are 32 different scales from 0.5 to 0.0078 pixels^{-1} that makes difficult to use colors legends to mark each case. Therefore, they are only used for qualitative error hints, where large error is presented at scales far away from the filter tuning peak frequency. Smooth error curves descend indicating gratings from high to lower spatial frequencies. For the filter tuning frequency, error is close to 0 and therefore it is not visible on the graphics. These figures also show error curves with flat responses or multiple narrow peaks. It happens when we pass from the tuning scale to coarse scales where the filter response confidence is quite low (energy is close to 0) but this reponses can be easily filtered using energy thresholds.



(a) **(b)**

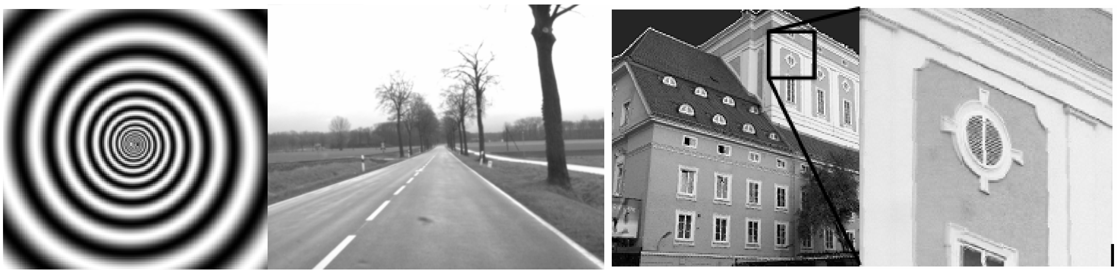
(c) **(d)**

Figure 14. Error evolution across the different stimulus orientations. We have used sinusoidal gratings as input, oriented to 64 different angles and with spatial scales from 0.5 pixels^{-1} to 0.0078 pixels^{-1}. Each spatial frequency filter output is represented on a different colour (there are 32 curves). (a) Monogenic signals results, error decreases with the spatial frequency in each plot. For low spatial frequencies, the filter provides quite accurate orientation estimations but it gets worse with high frequency gratings. (b) Second order Gaussian derivatives. There is a frequency range where the filter properly matches the stimulus orientation. For low frequency patterns, the error increases as represented on black lines at the bottom of the plot. (c) Fourth order Gaussian derivatives and (d) Gabor filters have a small bandwidth. Thus, in this case stimulus with spatial contexts far from the filter tuning frequency are prone to high orientation errors. Graphics (b), (c) and (d) use the Haglund approach for computing orientations based on a set of 8 oriented quadrature filters. Note that results

are unthresholded. Large errors appear in zones of almost zero energy but this feature can easily be used as confidence parameter for tuning the filter to the best spatial scale.

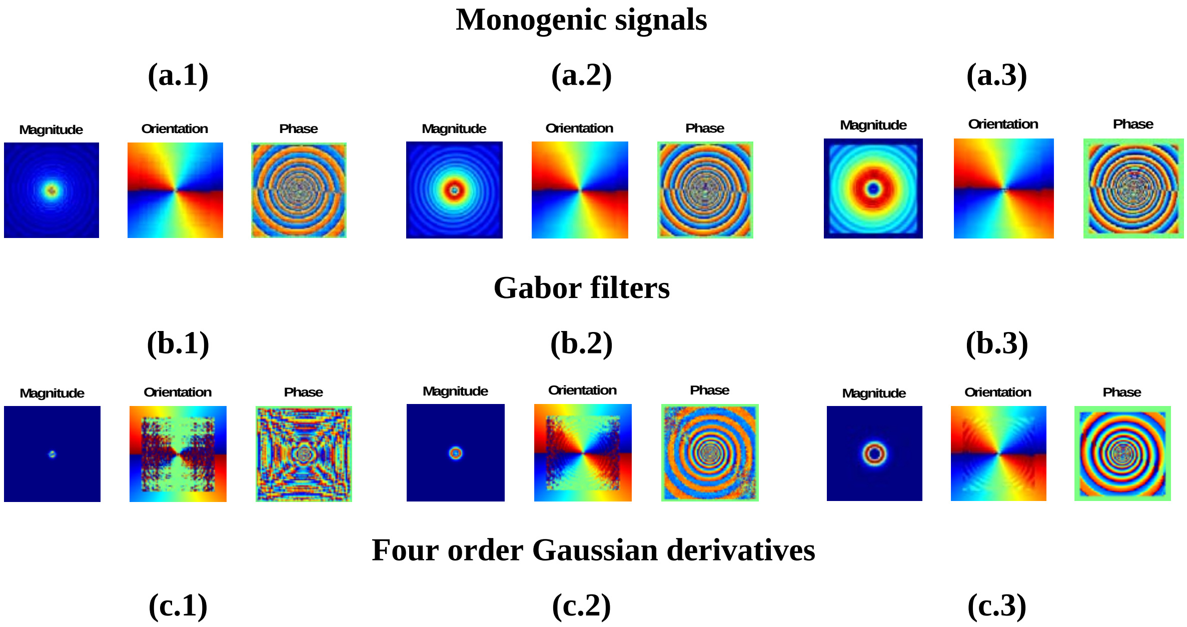
The qualitative results of features computation for images in Figure 15 are illustrated on Figures 16, 17 and 18. Three examples are shown:

1. Synthetic spiral image (Figure 15.a) that covers all the orientation as well as different spatial scales. Results presented in Figure 16.
2. Forward view of a road (Figure 15.b) with a well defined structure that allows easy identifications of the image features. Two different image scales are shown, the original one and the image reduced by a factor 4 in Figure 17.
3. Finally, real image of a house is used. We focus on the details of a circular skylight (Figure 15.c) and the extracted primitives are computed and shown in Figure 18.



(a) (b) (c)

Figure 15. Original images used for qualitative evaluation of the different approaches.



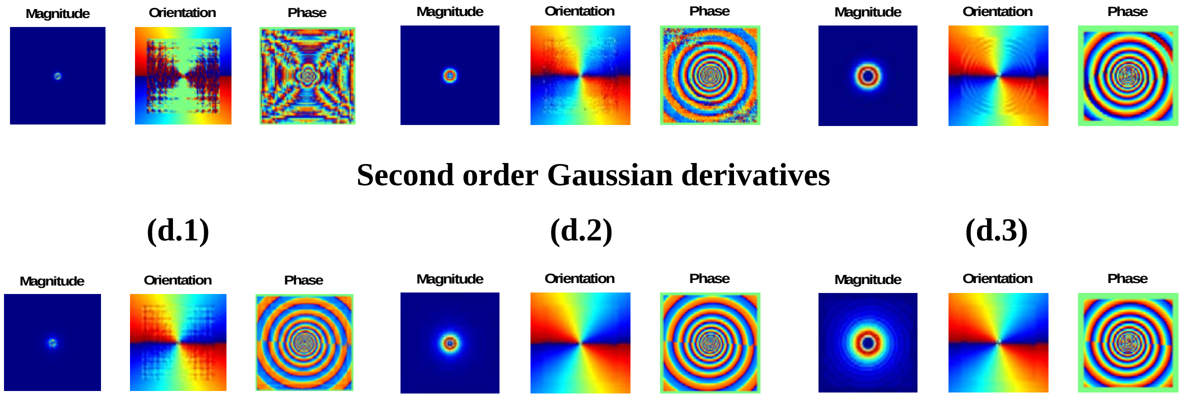
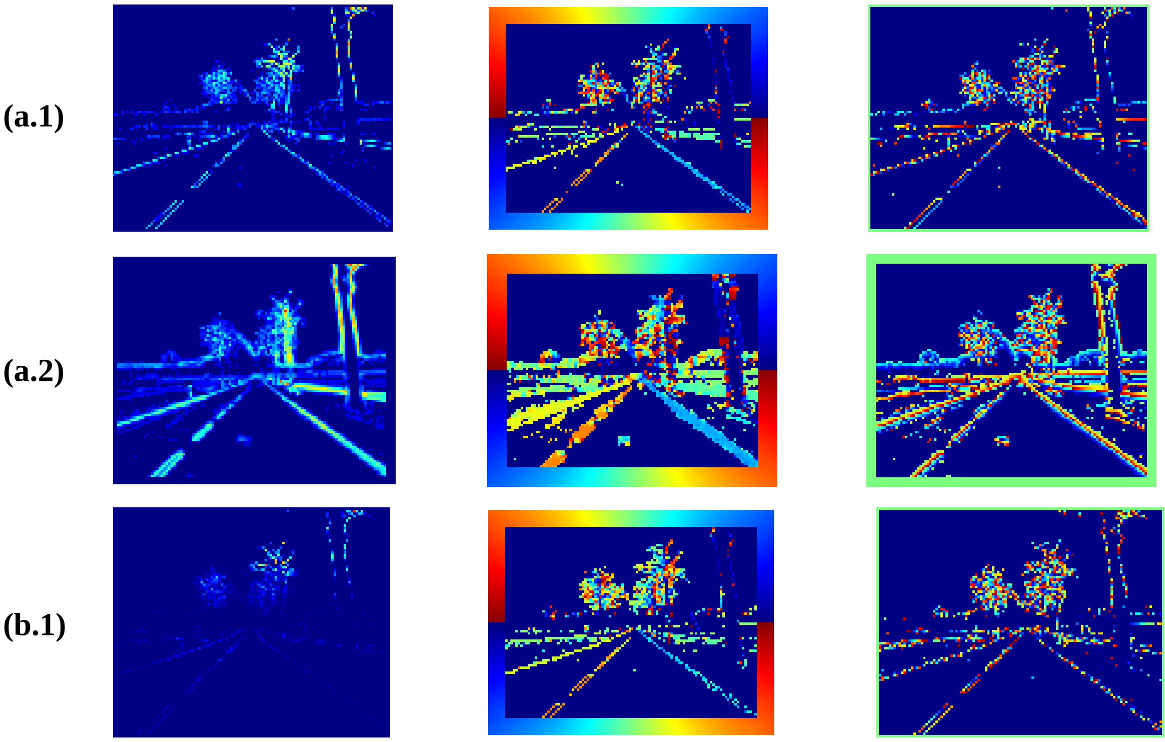


Figure 16. Orientation estimation for the image in Figure 15.a, using unthresholding results. Row (a) represents the results for the Monogenic signals, row (b) results of the Gabor filters, row (c) the fourth order Gaussian derivatives and (d) output from the second order Gaussian derivatives. Each column represent one scale, column 1 is the fine scale, column 2 represents a image resolution divided by 2 and column 3 represents the image with original resolution divided by 4. The results show that high resolution images are properly tuned only at the centre for the fourth order Gaussian derivatives and Gabor filters. The tuning region grows for lower resolution areas because the peak frequency is better tuned at these scales, as can be seen from the energy response images. Note that second order Gaussian derivatives and Monogenic signals, thanks to the wider bandwidth, allow the primitives computation at larger areas. An orientation frame is utilized for these images encoding with colours the different orientations. Note that we use the direction normal to the line (the filter axe) as orientation direction for the colormap.



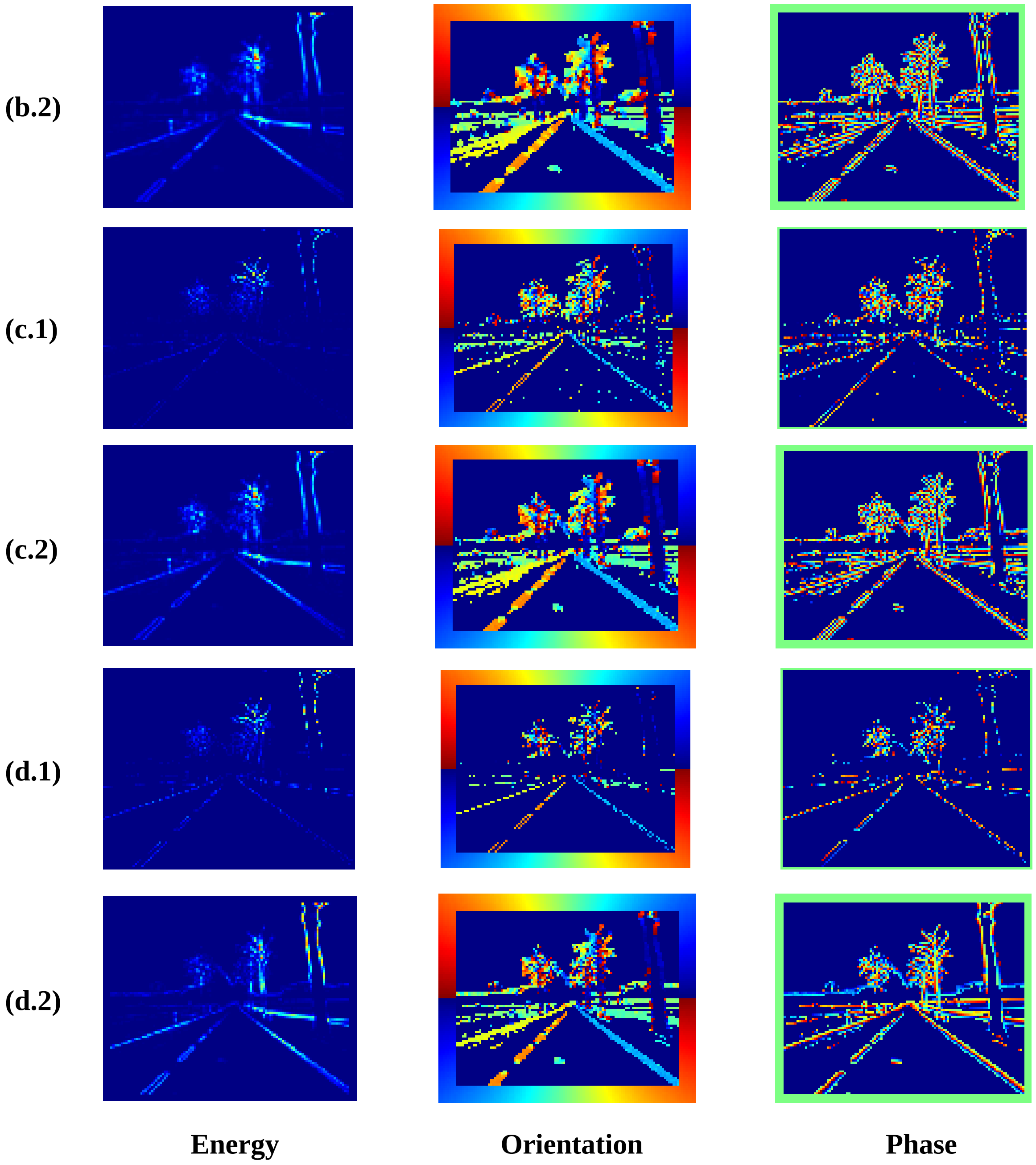


Figure 17. Image features computed the for road scene in Figure 15.b with a energy confidence threshold of $5e^{-3}$ times the maximum output. We use the previous filters: (a) Monogenic signals, (b) Gabor filters, (c) Fourth order Gaussian derivatives and (d) Second order Gaussian derivatives. For each filter, the image is computed at 2 scales, the original one and other resolution divided by four which is represented by the subindices x.1 and x.2 where x stand for a, b, c or d.

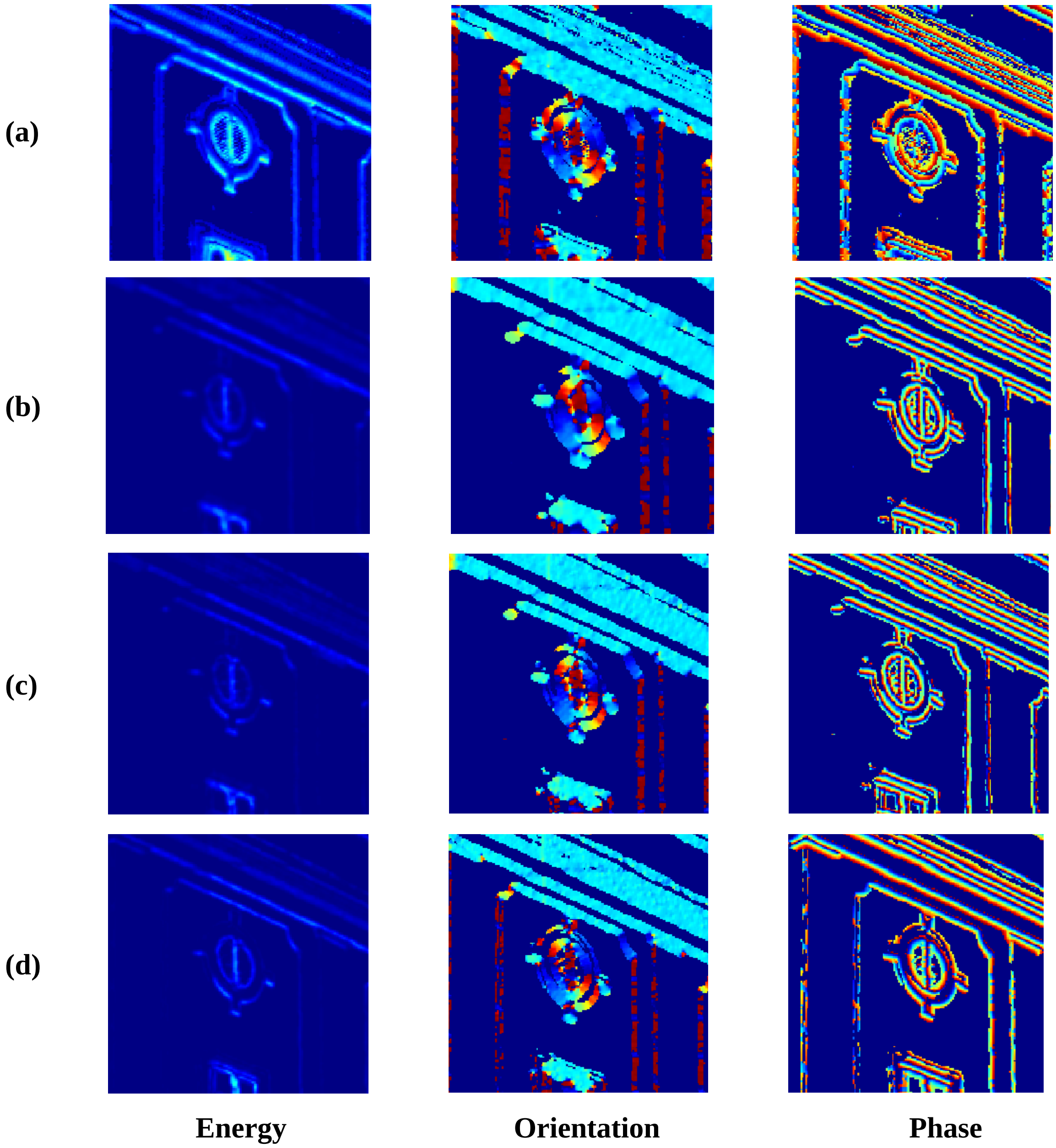


Figure 18. Image features computed for the circular skylight of Figure 15.c. We use the same filters: (a) Monogenic signals, (b) Gabor filters, (c) Fourth order Gaussian derivatives and (d) Second order Gaussian derivatives. We can appreciate that the phase information blur from (a) with the higher value to (b) and (c) with the lower value. A trade-off between these alternatives is represented by the (d) case based on the second order Gaussian derivatives.

7. Conclusions

From the previous analysis we conclude that all the filters responses have high accuracy for spatial scales close to their peak frequency. First, note that the bandwidth of the different approaches has a critical effect on feature computation at each scale.

Multiscale approaches can benefit from narrow tuning filter as in [11] but, for general applications with only one scale, this is a significant drawback. On [11], a multiscale algorithm for optical flow and stereo computation highlight the Gabor filter as the filter approach which produces the higher accuracy maps compared with the other alternatives. From section 6, a monoscale approach benefits from a wider bandwidth.

Several interpolation methods have been presented for these filters with the Haglund approach showing the best performance. Furthermore, as significantly different to the Monogenic signal, these filters provide multivalue responses at each orientation that open the utilization of these results for texture segmentation, intrinsic dimension analysis or other 2-D image structure as corners or junctions.

There is not an unique best approach. Thought in [11] Gabor is the best approach, in [14] Monogenic Signal provides better results. From the hardware implementation point of view, second order Gaussian derivatives could be the best option because:

1. They require the minimum number of resources. Only seven 2-D separable convolutions are required.
2. They are a good trade-off between spatial resolution and spatial scales range. Fine resolution is extracted compared to Monogenic signals although is coarser than Gabor or higher order Gaussian derivatives.
3. Their orientation accuracy is quite high (less than one degree of error) and therefore it fulfils the requirements of most applications.
4. Arbitrary orientations can be computed from the basic set of filters just changing the filters interpolation coefficients because they are Steerable filters.
5. It provide a good score in [14].

There are also some important conclusions related with the different filters for optical flow and stereo computation. According to [15], the main conclusions are:

- Gabor filters could be the best choice. Indeed, (a) isotropic filters lack of x-y separability, and (b) steerable filters have a broader orientation tuning. A more complete comparison (respect to their “final” performances) is still under study.
- Tensor-based methods are the best choice to extract local amplitude+phase+orientation, but the single phase-value “averaged over orientations” cannot be efficiently used for the estimation of disparity and

motion (it is better to avoid an early merge of information coming from orientation channels and to keep the phase information ϕ_i separate till the final decision).

Appendix I. Some hardware resources consumption estimations

The current situation is that the real-time system is now ready based on the Second order Gaussian derivatives approach. Thought maybe this is not the more accurate filter base, his implementation share most of the logic with the other approaches and give us an estimation of the required resources for the different options.

The platform used is the stand-alone one and therefore is not ready to be used as coprocessing system jet. Nevertheless, this is our next stage in order to make accessible this system to the entire DRIVSCO group.

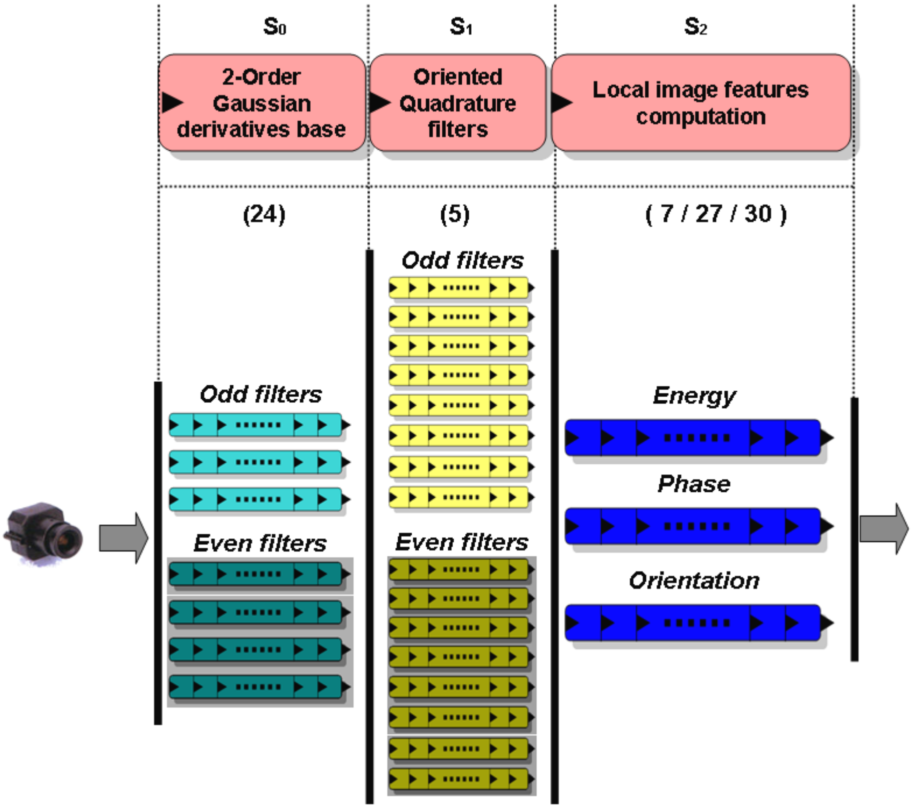


Figure A.1. Image features processing core. Coarse pipeline stages are represented at the top and superpipelined scalar units at the bottom. The number of parallel datapaths increase based on the algorithm structure. The whole system has more than 59 pipelined stages (without counting other interfacing hardware controllers such as memory or video input/output interfaces). This allows computing

the three image features at one estimation per clock cycle. The number of substages for each coarse-pipeline stage is indicated in brackets in the upper part of the figure.

The main stages of the system are schematically represented in Figure A.1, further details can are presented at [14]. The filter interpolation methods use the Haglund approach with phase estimation obtained by equation (33.a).

An estimation of the system resources for the different stages is included in table A.1 and the fully working system resources are presented in table A.2. The differences coming from the hardware controller (VGA for visualization, video input, memory management units, etc…).

Note that the convolution stage, S_0, requires about 12% of the whole system resources using Second Order Gaussian Derivatives. For example a Gabor filter based approach multiply by 24/14=1.7 the resources consumption and therefore their utilization must be strongly supported by a high numerical accuracy improvement compared to the presented filters (this represent to use the 21% of the whole system resources).

Table A.1.. Partial system resources required on a Virtex II XC2V6000-4 for the coarse pipeline stages described for this circuit. (*EMBS* stands for embedded memory blocks). The differences between the sum of partial subsystems and the whole core are explained in the text.

	Circuit stage	**Slices / (%)**	**EMBS / (%)**	**Embedded multipliers / (%)**	**f_{clk} (MHz)**
S_0	*Gaussian base convolutions*	4,170 (12)	8 (5)	50 (34)	85
S_1	*Oriented quadrature filters*	1,057 (3)	0	0	69
S_2	*Features Energy, Phase and Orientation computation*	2,963 (8)	0	6 (4)	89
	Whole processing core	7627 (22)	8 (5)	65 (45)	58.8

Table A.2. Complete system resources required for the local image features computing circuit. The circuits have been implemented on the RC300 prototyping board [CEL06d]. The only computing element is the Xilinx FPGA Virtex II XC2V6000-4. The system includes the image features processing unit, memory management unit, camera Frame-grabber, VGA signal output generation and user configuration interface.

(*Mpps*: *mega-pixels per second* at the maximum system processing clock frequency, *EMBS*: embedded memory blocks).

Slices / (%)	**EMBS / (%)**	**Embedded multipliers / (%)**	**Mpps**	**Image Resolution**	**Fps**
9135 (27%)	8 (5%)	65 (45%)	56.5	1000x1000	56.5

The figure A.2 shows the comparison between software and hardware results. Note that hardware quality is quite high and that the differences mainly appear due to the threshold limit (his values are quantized in the hardware system which modifies the saliency map).

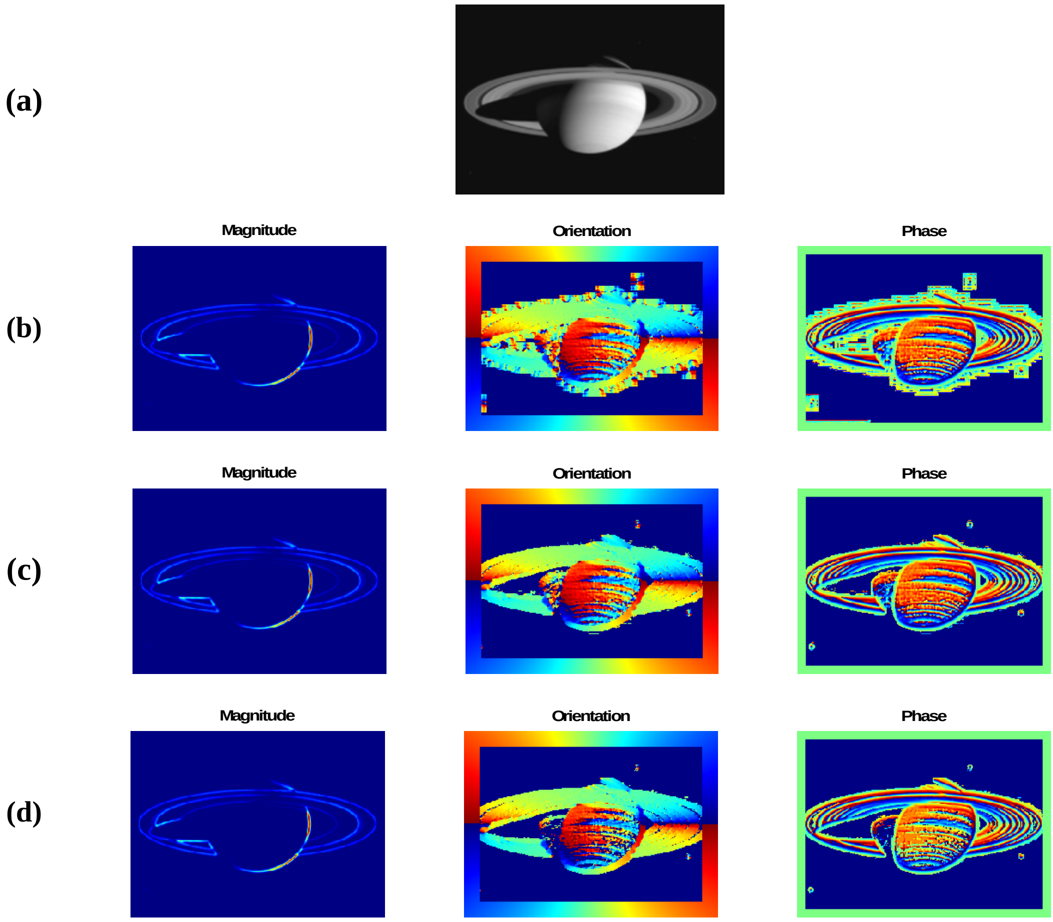


Figure A.2. Images features computation with several data representations and bit-widths. (a) Real image of Jupiter captured in the Cassini-Huygens space mission. (b) Image primitives computed using software with double precision floating point representation. (c) Image primitives computed using fixed point data representation with bit-widths choices of configuration B in Table 5.1. (d) Image primitives computed using fixed point data representation with bit-widths choices of configuration A in Table 5.1. This time we use an energy threshold that rejects pixels with energy bellow the maximum energy • 10^{-5}. This allows seeing small differences between floating point and fixed point approaches. The threshold rejects more pixels for fixed point that for floating point. This can easily be justified because of the energy quantization effect. Despite that, the results are quite similar and the differences only reject unreliable data estimations. Although not all of them are rejected because we consider a very low restrictive threshold.

Next figures A.3 and A.4 finally show two qualitative examples of the hardware system to evaluate system accuracy.

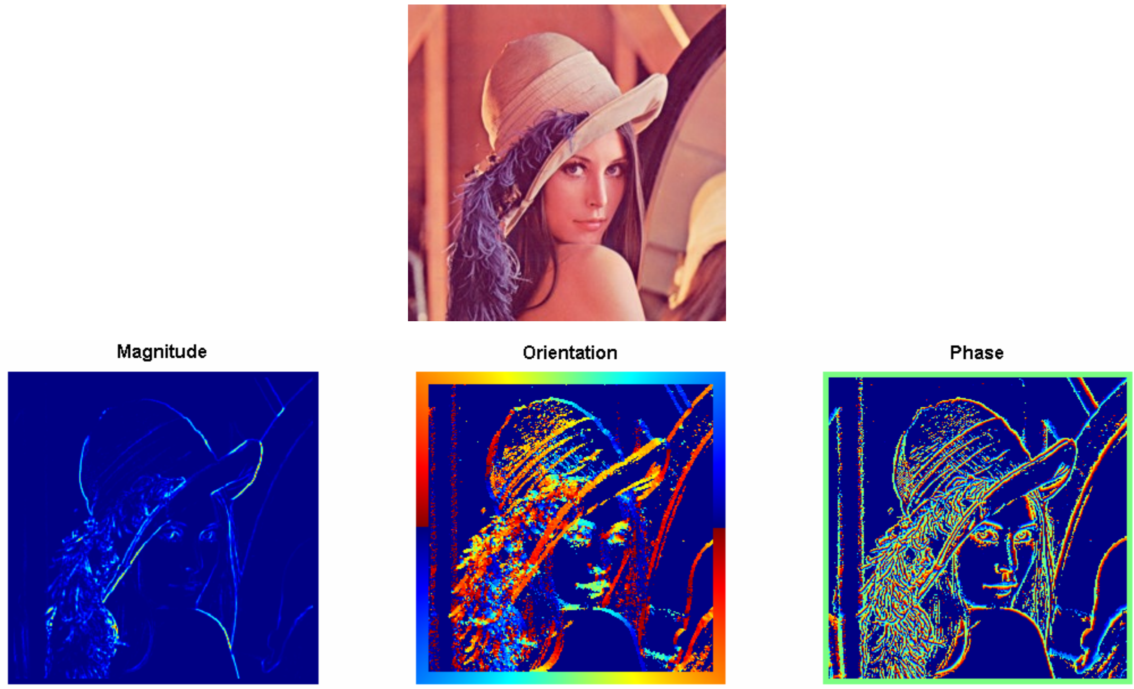
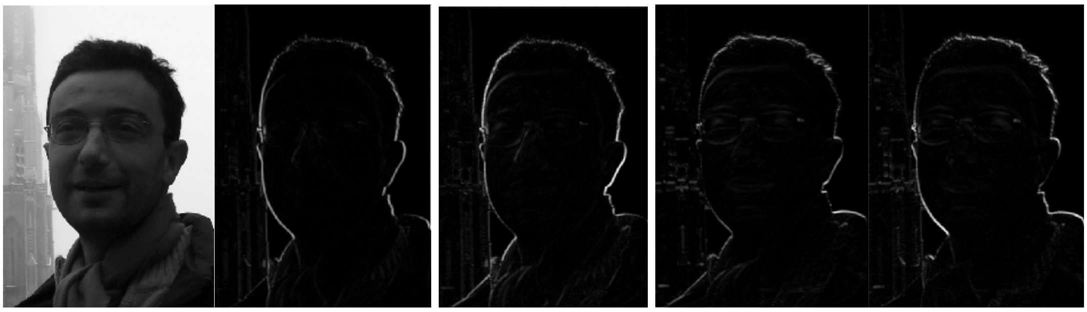


Figure A.3. System results for the known image of Lena (left image). The three right pictures present the computed features for this image. Note that although restricted fixed-point arithmetic is used, the quality of the features is quite high as can be seen looking on small details such as presented on hat's plumes.



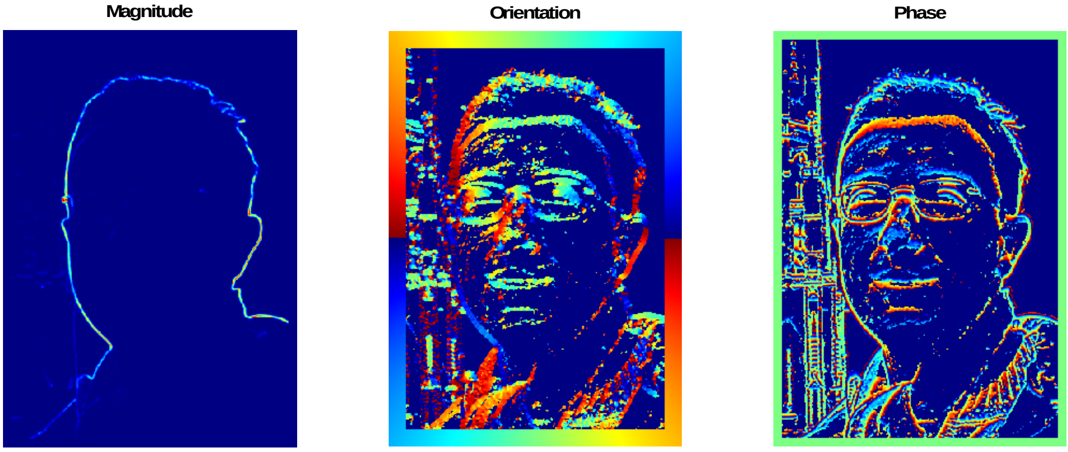


Figure A.4. System results for a real image. First row, left side, represent a photograph of a close face with a tower in the left side. The tower is blurred due to the fog and therefore has low contrast. Four images to the right represent the quadrature filters energy output at 4 defined orientations (0, π/4, π/2 and 3π/4). The second row represents the computed features for that image. Note that at the energy image, the tower is almost invisible but not for the orientation and phase images. This illustrates that the structure based features have not got contrast dependency, and therefore the tower appears clearly visible.

8. Bibliogaphy

[1] M. Felsberg, Low-Level Image Processing with the Structure Multivector, Ph.D. thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel, 2002.

[2] D.J. Fleet and A.D. Jepson: "Computation of component image velocity from local phase information", Int. Journal of Comp. Vision 5, 1990.

[3] Efficient Spatial-Domain Implementation of a Multiscale Image Representation Based on Gabor Functions. O. Nestares, R. Navarro, J. Portilla and A. Tabernero. Journal of Electronic Imaging, 7(1), pp. 166-173 (1998).

[4] Koenderink J.J. and van Doom A.J.: Representation of local geometry in the visual system, Biological Cybernetics, 55(6) (1987), 367-375.

[5] Bloom J.A. and Reed T.R.: A Gaussian derivative-based transform, IEEE Transactions on Image Processing, 5(3) (1996), 551-553.

[6] Freeman W. T. and Adelson E. H.: The design and use of steerable filters. IEEE Pattern Analysis and Machine Intelligence, 13(9) (1991), 891–906.

[7] Silvio Sabatini, DRIVSCO Meeting in Copenhagen, March 2006.

[8] Adaptive Multidimensional Filtering. Leif Haglund, PhD Dissertation no. 284, October, 1992

[9] A.C. Bovik, M. Clark and W.S. Geisler. Multichannel texture analysis using localized spatial filters. IEEE Trans. PAMI, 12(1), 55–73, 1990.

[10] Gabor filter introduction from The Computer Vision Lab at GET, University of Paderborn, Department of Electrical Engineering. Online resource available at: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TRAPP1/filter.html (Last access, May 06).

[11] K. Pauwels, "Filter Evaluation: Optic Flow and Disparity", *DRIVSCO Technical report,* June 2006.

[12] G. Gastaldi, S. P. Sabatini, F. Solari, "1-D vs 2-D Gabor filters for Phase-based Disparity Estimation", *DRIVSCO Technical report*, May 2006.

[13] N. Pugeault, "Evaluation of the sub–pixel accuracy of the primitive extraction for different filters", *DRIVSCO Technical report*, June 2006.

[14] J. Díaz, "Multimodal bio-inspired vision system. High performance motion and stereo processing architecture", PhD Thesis dissertation. University of Granada, 2006.

[15] Karl Pauwels and Marc M. Van Hulle, "Filter Evaluation: Optic Flow and Disparity", DRIVSCO Technical report, 14th December 2006.

Appendix C

Filter evaluation: optic flow and disparity. Technical report.

Filter Evaluation: Optic Flow and Disparity

Karl Pauwels and Marc M. Van Hulle

14th December 2006

1 Filter Specification

A wide variety of filters have been tested and the precise specifications are encoded in the filter name, which has the following form:

```
{1}t{2}(B)(o){3}f{4}{5}
```

The parameters (between brackets) are:

{1} the filter type, either G for a Gabor or S for a steerable filter

{2} the tap size of the filter

{3} B: the filter bandwidth for Gabor filters. For steerable filters this is not specified since a fixed relation exists with the derivative order. Instead, the derivative order is specified with o

{4} the peak frequency (f_0) of the filter

{5} additional settings related to the design procedure. The possibilities are:

- I: the filter has been rounded to integer values
- D: **no** DC correction has been performed
- E: **no** no energy balancing has been performed

and combinations thereof.

Energy balancing has been included as an additional constraint in the optimization, namely by enforcing that the sum of squares of the even filter equals that of the odd. Note that this is no longer enforced after rounding to integer values.

Our main design choice is to use 11 taps and a peak frequency $f_0 = 1/4$, but for comparison, some other combinations have also been included. The additional filters have been taken from Feeman and Adelson [1991] and are a 13 taps fourth order steerable filter with $f_0 = 0.23$ and a 9 taps second order steerable filter with $f_0 = 0.21$. Gabor filters have been included that match the frequency and bandwidth of these filters:

- `Gt11B0.0833f0.25`: 1 octave bandwidth (Nestares et al. [1998])
- `Gt11B0.0884f0.25` matches `St11o4f0.25`
- `Gt11B0.0711f0.21` matches `St9o2f0.21`
- `Gt11B0.0750f0.23` matches `St13o4f0.23`

2 Optic Flow

The optic flow algorithm is a coarse-to-fine implementation of the phase-based algorithm by Gautama and Van Hulle [2002]. The algorithm starts at the lowest resolution and optic flow is computed only after compensating (warping) for the lower resolution estimates. To avoid overly smooth flow fields, only estimates that are reliable at the highest resolution are retained. In this evaluation, the same settings are used at every scale, an adaptive threshold has not yet been implemented.

The different filters are evaluated using the (realistic) synthetic sequences from Barron et al. [1994] for which ground truth optic flow is available. The three sequences used are *translating tree*, *diverging tree* and *yosemite*. The center frame of each sequence and an example flow field are shown in the top and middle row of Fig. 1.

Table 2 contains the optic flow errors (average and standard deviation of angular error) and density (percentage reliable flow vectors), evaluated using the performance measures suggested by [Barron et al., 1994], for the different filters. In this evaluation, `thres_lin` = 0.05 and `nc_min` = 4. A border region of five pixels has been excluded from the evaluation. For *yosemite*, the cloud region has also been left out of the evaluation.

The same evaluation has been performed using `Gt11B0.0833f0.25` but for different values of `thres_lin`. The results are shown in Table 2. It is clear from this table that the reliability measure behaves well. If the linearity threshold is lowered, the average and standard deviation of the errors decrease, as does the optic flow density.

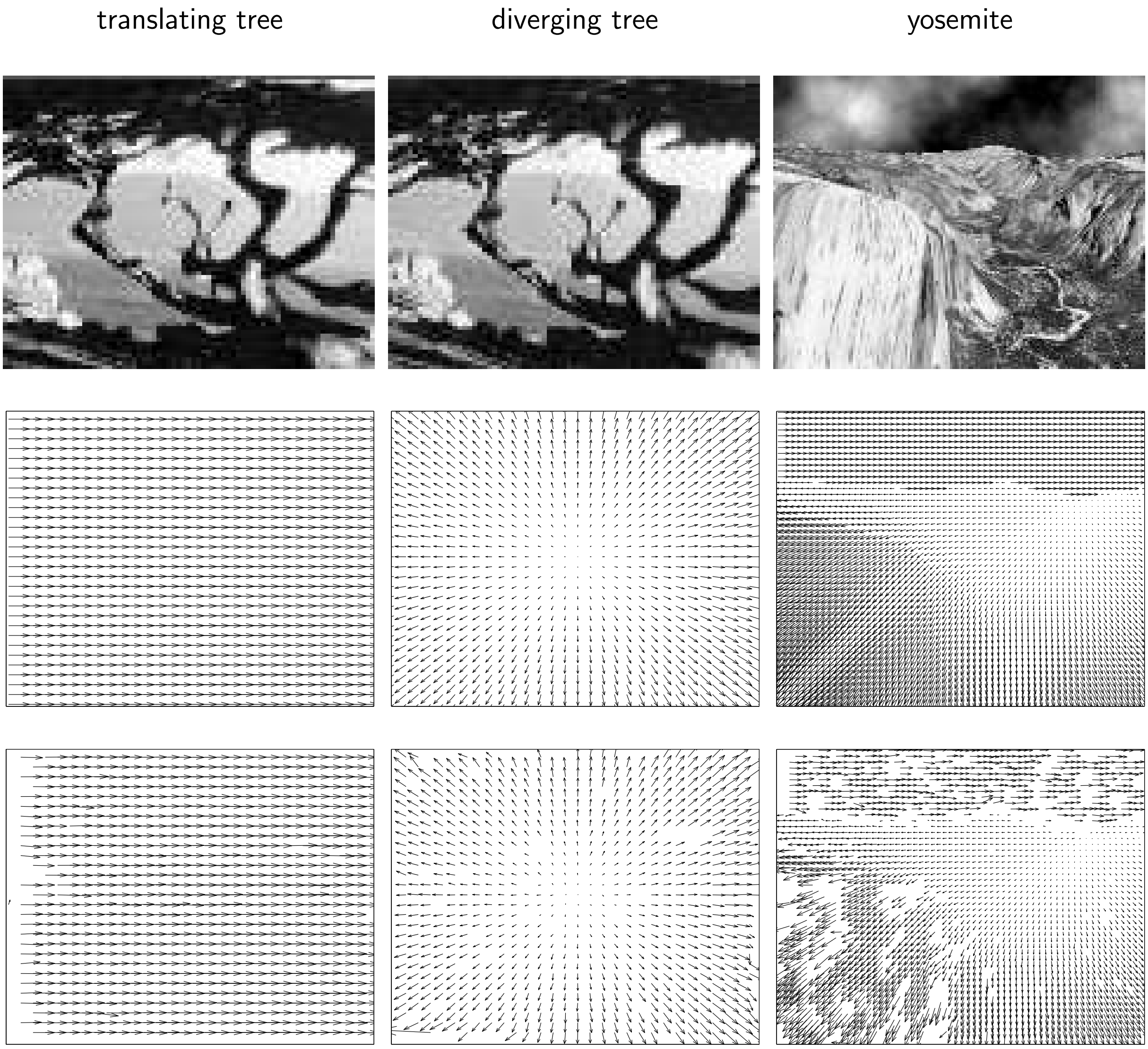
3 Disparity

The coarse-to-fine disparity algorithm uses the same filter outputs as the optic flow algorithm. For each orientation θ, the phase difference $\Delta\phi_\theta$ between left and right filter outputs is computed using the technique of Solari et al. [2001]. This phase difference relates to the 'component' disparity (projected on the orientation orthogonal to the filter orientation) in the following way:

$$\delta_\theta(\mathbf{x}) \cos\theta = \frac{\Delta\phi_\theta(\mathbf{x})}{2\pi f_0} \ , \tag{1}$$

where f_0 is the peak frequency. Since different estimates are available at each pixel, the median is used to robustly merge these into a single disparity value.

$$\delta(\mathbf{x}) = \operatorname*{median}_{\theta \in V(\mathbf{x})} \delta_\theta(\mathbf{x}) \ , \tag{2}$$



translating tree

diverging tree

yosemite

Figure 1: Center frame and corresponding ground truth flow fields and estimated flow fields obtained with `Gt11B0.0833f0.25` using 4 scales, `thres_lin` = 0.05 and `nc_min` = 4. All flow fields have been scaled and subsampled 5 times.

Table 1: Optic flow errors obtained with `thres_lin` = 0.05 and `nc_min` = 4

	translating tree			diverging tree			yosemite (no cloud)		
	avg	std	dens	avg	std	dens	avg	std	dens
one scale									
`Gt11B0.0833f0.25`	10.50	15.70	20.60	6.61	4.57	84.14	7.71	8.56	48.11
`Gt11B0.0833f0.25DE`	44.03	25.27	20.64	15.17	12.61	79.91	21.74	23.79	55.58
`Gt11B0.0833f0.25E`	10.57	15.85	20.52	6.63	4.56	84.10	7.73	8.55	48.07
`Gt11B0.0833f0.25I`	10.44	15.49	20.43	6.63	4.59	83.97	7.72	8.61	48.06
`Gt11B0.0833f0.25IE`	10.44	15.46	20.57	6.61	4.52	84.17	7.74	8.58	48.10
`Gt11B0.0884f0.25`	12.04	16.35	16.59	7.36	5.23	82.50	8.60	9.15	47.34
`Gt11B0.0711f0.21`	8.25	6.31	36.84	7.81	6.19	89.06	8.26	8.76	55.38
`Gt11B0.0750f0.23`	8.29	7.91	31.78	7.23	5.47	88.09	7.99	8.70	53.09
`St11o4f0.25`	24.75	23.86	6.81	10.33	8.23	69.65	13.42	14.07	43.18
`St11o4f0.25DE`	27.31	24.14	6.71	11.18	9.14	70.19	13.49	14.08	43.09
`St11o4f0.25E`	24.63	23.58	6.81	10.34	8.20	69.59	13.43	14.08	43.14
`St11o4f0.25IDE`	24.81	23.74	6.71	10.36	8.23	69.49	13.41	14.05	43.06
`St13o4f0.23`	20.23	18.58	12.68	10.31	8.25	77.35	13.52	14.39	47.96
`St9o2f0.21`	43.70	17.27	14.72	18.33	11.45	66.94	19.90	16.22	44.10
`St9o4f0.25`	25.94	24.31	6.45	10.44	8.29	68.32	13.46	14.07	42.83
four scales									
`Gt11B0.0833f0.25`	0.61	0.73	97.72	2.05	2.28	95.60	2.15	3.12	81.81
`Gt11B0.0833f0.25DE`	1.71	3.21	99.09	3.57	4.20	97.99	9.47	14.06	90.71
`Gt11B0.0833f0.25E`	0.62	0.73	97.74	2.05	2.25	95.60	2.14	3.10	81.90
`Gt11B0.0833f0.25I`	0.62	0.72	97.72	2.05	2.27	95.55	2.14	3.09	81.82
`Gt11B0.0833f0.25IE`	0.62	0.73	97.75	2.04	2.30	95.57	2.14	3.08	81.87
`Gt11B0.0884f0.25`	0.65	0.71	97.77	2.04	2.14	95.20	2.19	2.96	83.02
`Gt11B0.0711f0.21`	0.62	0.90	98.41	2.35	2.61	97.43	2.19	2.89	88.69
`Gt11B0.0750f0.23`	0.62	0.84	98.22	2.19	2.28	96.90	2.14	2.91	86.79
`St11o4f0.25`	0.97	1.27	96.60	2.39	2.62	93.21	2.96	4.46	85.04
`St11o4f0.25DE`	1.03	1.54	97.08	2.58	2.85	94.54	3.05	4.44	85.28
`St11o4f0.25E`	0.98	1.36	96.58	2.40	2.59	93.20	2.98	4.46	85.09
`St11o4f0.25IDE`	0.97	1.29	96.56	2.39	2.56	93.17	2.98	4.47	85.06
`St13o4f0.23`	0.93	1.46	97.32	2.52	3.00	95.09	2.92	4.38	88.77
`St9o2f0.21`	3.75	5.00	93.91	3.97	4.48	94.07	5.86	8.54	86.62
`St9o4f0.25`	0.98	1.28	96.69	2.38	2.43	93.02	3.00	4.42	85.10

Table 2: Optic flow errors obtained with Gt11B0.0833f0.25 for different settings of thres_lin

thres_lin	translating tree			diverging tree			yosemite (no cloud)		
	avg	std	dens	avg	std	dens	avg	std	dens
one scale									
0.01	8.63	15.67	1.12	6.37	3.85	43.39	6.04	4.39	23.93
0.05	10.50	15.70	20.60	6.61	4.57	84.14	7.71	8.56	48.11
0.10	11.30	16.35	44.61	6.71	4.70	94.88	9.54	12.33	58.73
four scales									
0.01	0.50	0.45	87.19	1.71	1.65	69.67	1.77	2.06	44.07
0.05	0.61	0.73	97.72	2.05	2.28	95.60	2.15	3.12	81.81
0.10	0.68	0.91	99.05	2.17	2.46	98.88	2.52	4.33	90.60

where $V(\mathbf{x})$ is the set of valid component disparities. The current implementation includes an energy-based validity measure thres_e.

The control scheme is similar to that of the optic flow algorithm; starting at the lowest resolution and computing disparity only after compensating (warping) for the lower resolution estimates.

We use the *tsukuba*, *sawtooth* and *venus* stereo-pairs from Scharstein and Szeliski [2002] to evaluate the different filters. Since we are interested in the precision of the filters we do not use the integer-based measures used there but instead compute the mean and standard deviation of the absolute disparity error. In order not to distort the results with large errors, the error is evaluated only at the regions that are textured, non-occluded and continuous.

Figure 2 contains the left frame, ground truth and an example estimated disparity (using Gt11B0.0833f0.25) for these stereo-pairs. Note that the disparity in these pairs is quite large (up to 20 pixels).

Table 3 contains the results for all filters. Since we did not employ the validity measure, the density is close to 100% on all occasions.

4 Conclusions

- Gabor filters outperform steerable filters on all occasions
- concerning optic flow, the difference between Gabor and steerable is larger in the single scale setting, this is most likely due to the multiple corrections that occur in the multiscale setting
- removal of the DC component significantly improves the results
- rounding to integer values and energy balancing have little effect on the results
- for the Gabor filters, of the bandwidths tested, one octave (the same as in Nestares et al. [1998]) results in the best results

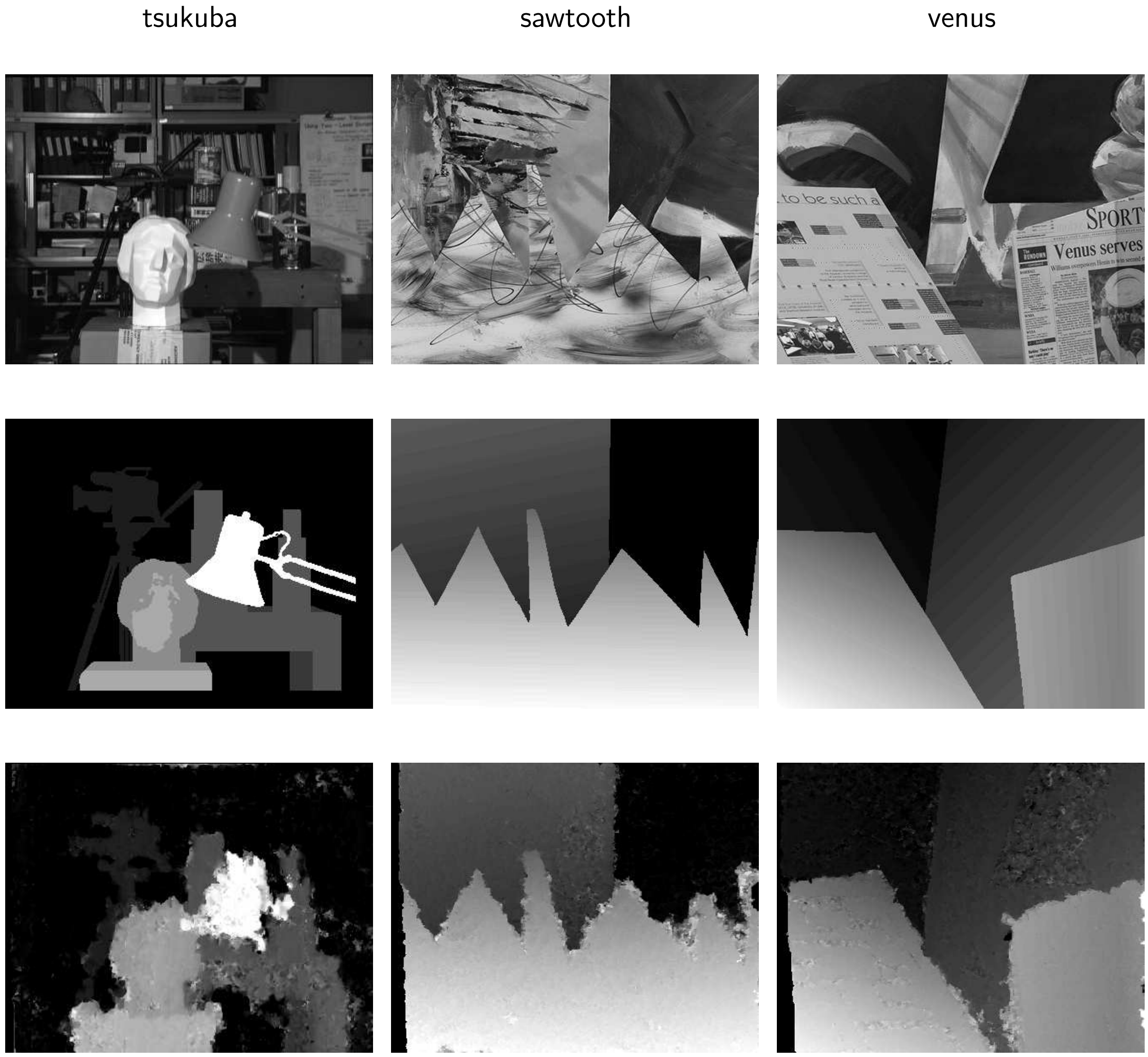


Figure 2: Left frame (top row), ground truth disparity (middle row), and estimated disparity using `Gt11B0.0833f0.25` with `thres_e` = 0. (bottom row).

Table 3: energy threshold = 0

	tsukuba			sawtooth			venus		
	avg	std	dens	avg	std	dens	avg	std	dens
Gt11B0.0833f0.25	0.32	0.61	100.00	0.41	1.26	99.64	0.25	0.77	99.65
Gt11B0.0833f0.25DE	0.35	0.65	100.00	0.67	1.42	99.59	0.55	1.17	99.79
Gt11B0.0833f0.25E	0.32	0.61	100.00	0.41	1.26	99.66	0.26	0.77	99.65
Gt11B0.0833f0.25I	0.32	0.61	100.00	0.41	1.26	99.67	0.26	0.79	99.65
Gt11B0.0833f0.25IE	0.32	0.61	100.00	0.41	1.25	99.64	0.26	0.79	99.65
Gt11B0.0711f0.21	0.33	0.69	100.00	0.42	1.33	99.47	0.30	1.08	99.53
Gt11B0.0750f0.23	0.32	0.65	100.00	0.42	1.52	99.40	0.30	1.10	99.58
Gt11B0.0884f0.25	0.32	0.60	100.00	0.40	1.23	99.62	0.26	0.83	99.66
St11o4f0.25	0.36	0.68	100.00	0.50	1.86	99.44	0.40	1.30	99.65
St11o4f0.25DE	0.36	0.69	100.00	0.46	1.57	99.56	0.40	1.28	99.63
St11o4f0.25E	0.36	0.68	100.00	0.50	1.88	99.48	0.41	1.34	99.64
St11o4f0.25IDE	0.36	0.68	100.00	0.47	1.67	99.49	0.41	1.35	99.64
St13o4f0.23	0.36	0.71	100.00	0.56	2.45	99.27	0.45	1.62	99.48
St9o2f0.21	0.44	0.79	100.00	0.88	1.99	99.37	0.89	2.34	99.19
St9o4f0.25	0.36	0.68	100.00	0.44	1.42	99.58	0.40	1.30	99.64

References

J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

W.T. Feeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

T. Gautama and M.M. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Trans. Neural Networks*, 13(5):1127–1136, 2002.

O. Nestares, R. Navarro, J. Portilla, and A. Tabernero. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, 1998.

D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1–3):7–42, 2002.

F. Solari, S.P. Sabatini, and G.M. Bisio. Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Electronics Letters*, 37(23):1382–1383, 2001.

Appendix D

Excel file comparing the impact on memory and bandwidth of different system specifications.

DRIVSCO SYSTEM REQUIREMENTS

System specifications

Stereo views (left and right images)	2
Horizontal resolution	1024
Vertical resolution	1024
Input data bit-width	8
Frame-rate	25
Number of pyramic levels	5
Number of orientations	8
Oriented filters and magnitud bit-width	9
Orientation bit-width	9
Phase bit-width	9
Optical flow bit-width	24
Stereo bit-width	12

Extra parameters

Pyramid size	1396736
An	4096
Multiscale factor	1,3320313
FPGA buffers	2
Temporal filter size for optical flow	5
High level vision signal loops (input like image warping or local thresholds)	1

Downloading band-width (PC--> FPGA)

629,1456	**Mb/s**
78,6432	**MB/s**

Uploading bandwidth (FPGA-->PC)

Bandwidth for oriented filters

10056,5	**Mb/s**
1257,062	**MB/s**

Bandwidth for primitives (magnitud, orientation and phase)

1885,594	**Mb/s**
235,6992	**MB/s**

Bandwidth for stereo and motion

943,7184	**Mb/s**
117,9648	**MB/s**

System configuration bandwith requirements

input images + primitives + oriented filters

12571,24	**Mb/s**
1571,405	**MB/s**

input images + primitives + stereo + optical flow

3458,458	**Mb/s**
432,3072	**MB/s**

Internal memory storage requirements

Input images pyramid

7,992188	**MB**

Oriented filters pyramid

47,95313	**MB**

Primitives: magnitud, orientation and phase (x temporal filter size)

(We consider that only left image optical flow is computed)

23,97656	**MB**

Stereo + optical flow outputs

9	**MB**

TOTAL INTERNAL MEMORY REQUIREMENTS

88,92188	**MB**

Standard interfaces banwidths peak

PCIe (1 lane)	256 MB/s
Basic PCI	133 MB/s
Ethernet 1 Gbits	128 MB/s
Firewire	50 MB/s
USB 2.0	60 MB/s

DRIVSCO SYSTEM REQUIREMENTS

System specifications

Stereo views (left and right images)	2
Horizontal resolution	800
Vertical resolution	600
Input data bit-width	8
Frame-rate	25
Number of pyramic levels	4
Number of orientations	8
Oriented filters and magnitud bit-width	8
Orientation bit-width	8
Phase bit-width	8
Optical flow bit-width	24
Stereo bit-width	12

Extra parameters

Pyramid size	637500
An	7500
Multiscale factor	1,328125
FPGA buffers	2
Temporal filter size for optical flow	5
High level vision signal loops (input like image warping or local thresholds)	1

Downloading band-width (PC--> FPGA)

288	**Mb/s**
36	**MB/s**

Uploading bandwidth (FPGA-->PC)

Bandwidth for oriented filters

4080	**Mb/s**
510	**MB/s**

Bandwidth for primitives (magnitud, orientation and phase)

765	**Mb/s**
95,625	**MB/s**

Bandwidth for stereo and motion

432	**Mb/s**
54	**MB/s**

System configuration bandwith requirements

input images + primitives + oriented filters

5133	**Mb/s**
641,625	**MB/s**

input images + primitives + stereo + optical flow

1485	**Mb/s**
185,625	**MB/s**

Internal memory storage requirements

Input images pyramid

3,647804	**MB**

Oriented filters pyramid

19,45496	**MB**

Primitives: magnitud, orientation and phase (x temporal filter size)

(We consider that only left image optical flow is computed)

9,727478	**MB**

Stereo + optical flow outputs

4,119873	**MB**

TOTAL INTERNAL MEMORY REQUIREMENTS

36,95011	**MB**

Standard interfaces banwidths peak

PCIe (1 lane)	256 MB/s
Basic PCI	133 MB/s
Ethernet 1 Gbits	128 MB/s
Firewire	50 MB/s
USB 2.0	60 MB/s

System specifications

Stereo views (left and right images)	2
Horizontal resolution	512
Vertical resolution	512
Input data bit-width	8
Frame-rate	25
Number of pyramic levels	4
Number of orientations	8
Oriented filters and magnitud bit-width	8
Orientation bit-width	8
Phase bit-width	8
Optical flow bit-width	20
Stereo bit-width	10

Extra parameters

Pyramid size	348160
An	4096
Multiscale factor	1,328125
FPGA buffers	2
Temporal filter size for optical flow	5
High level vision signal loops (input like image warping or local thresholds)	1

Downloading band-width (PC--> FPGA)

157,2864	**Mb/s**
19,6608	**MB/s**

Uploading bandwidth (FPGA-->PC)

Bandwidth for oriented filters

2228,224	**Mb/s**
278,528	**MB/s**

Bandwidth for primitives (magnitud, orientation and phase)

417,792	**Mb/s**
52,224	**MB/s**

Bandwidth for stereo and motion

196,608	**Mb/s**
24,576	**MB/s**

System configuration bandwith requirements

input images + primitives + oriented filters

2803,302	**Mb/s**
350,4128	**MB/s**

input images + primitives + stereo + optical flow

771,6864	**Mb/s**
96,4608	**MB/s**

Internal memory storage requirements

Input images pyramid

1,992188	**MB**

Oriented filters pyramid

10,625	**MB**

Primitives: magnitud, orientation and phase (x temporal filter size)

(We consider that only left image optical flow is computed)

5,3125	**MB**

Stereo + optical flow outputs

1,875	**MB**

TOTAL INTERNAL MEMORY REQUIREMENTS

19,80469	**MB**

Standard interfaces banwidths peak

PCIe (1 lane)	256 MB/s
Basic PCI	133 MB/s
Ethernet 1 Gbits	128 MB/s
Firewire	50 MB/s
USB 2.0	60 MB/s

MEMORY REQUERIMENTS (MB)

	Optical flow	Stereo	Features	WHOLE SYSTEM
VHA	6	3	23,98	88,92
HA	2,75	1,37	9,72	36,95
MA	1,25	0,625	5,31	19,8

BANDWIDTH REQUERIMENTS (MHz)

	Optical flow	Stereo	Features	WHOLE SYSTEM
VHA	78,6	39,3	235,1	432,3
HA	36	18	95,6	185,6
MA	16,4	8,2	52,2	96,4

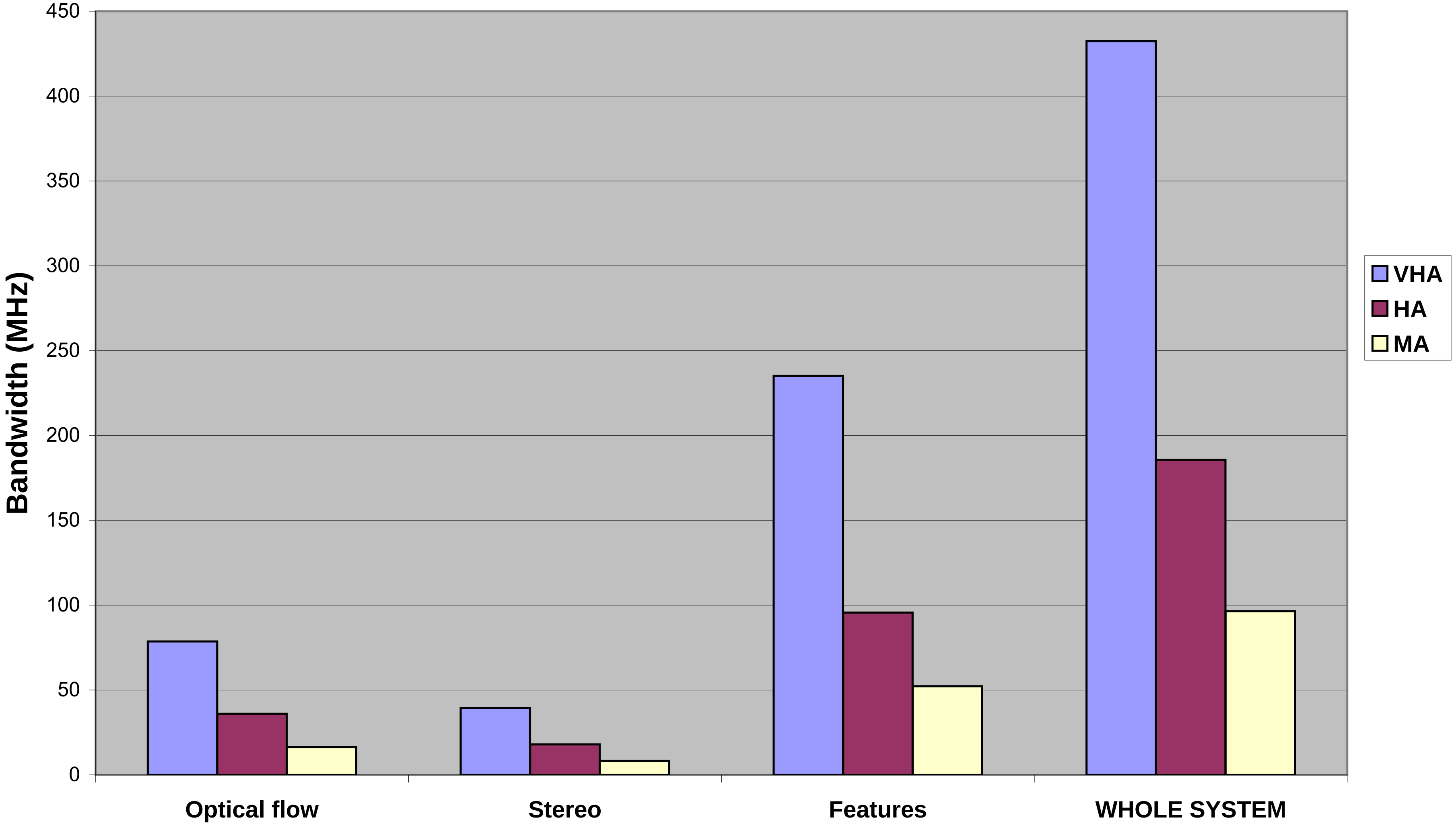
IMPORTANT NOTES: (PLEASE, TAKE THIS UNDER CONSIDERATION!)

- Optical flow is computed only for the left image
- We consider eight filters orientations
- Input images resolution are:

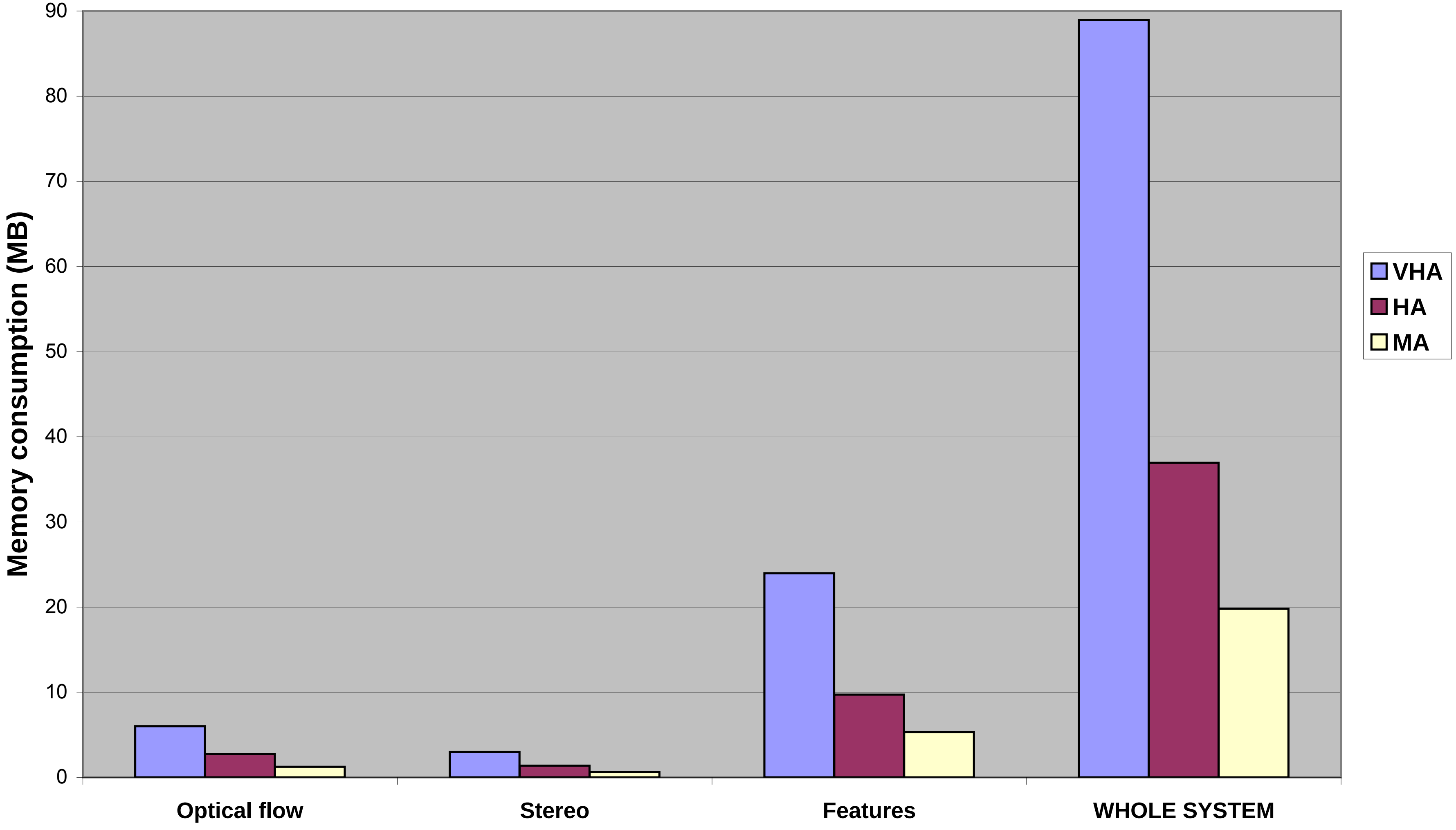
  1 Mpixels for VHA
  800x600 pixels for HA
  512x512 pixels for MA

- Frame-rate: 25 f 800x600 pixels for HA
- Features include 512x512 pixels for MA
- Total Memory resources consumption includes internal processing memory storage as well as memory to allowing double buffer computation
- Bandwidth requeriments include data transmission from PC to the FPGA

Early vision communication bandwith



Early vision memory requirements



Appendix E

Cross-modality examples. Technical report.

Crossmodal interactions

Eduardo Ros (9 of February 2007)
With material of different partners (BCCN, KUL, UGE, SDU)

Abstract: The crossmodal interactions are well defined mechanisms that allow efficient combination of single modality cues (motion, stereo, colour, etc) to obtain information based on two or more of these types of cues. In this short report we give some examples of such crossmodal interactions that are investigated in the framework of DRIVSCO.

1. Introduction

In an early vision system we can still distinguish different stages: pre-processing (band-pass spatial filtering), single modality extraction (motion, stereo, etc) and cross-modal interactions (motion-in-depth, 3D segmentation, etc). The scheme of this structure is illustrated in Fig. 1.

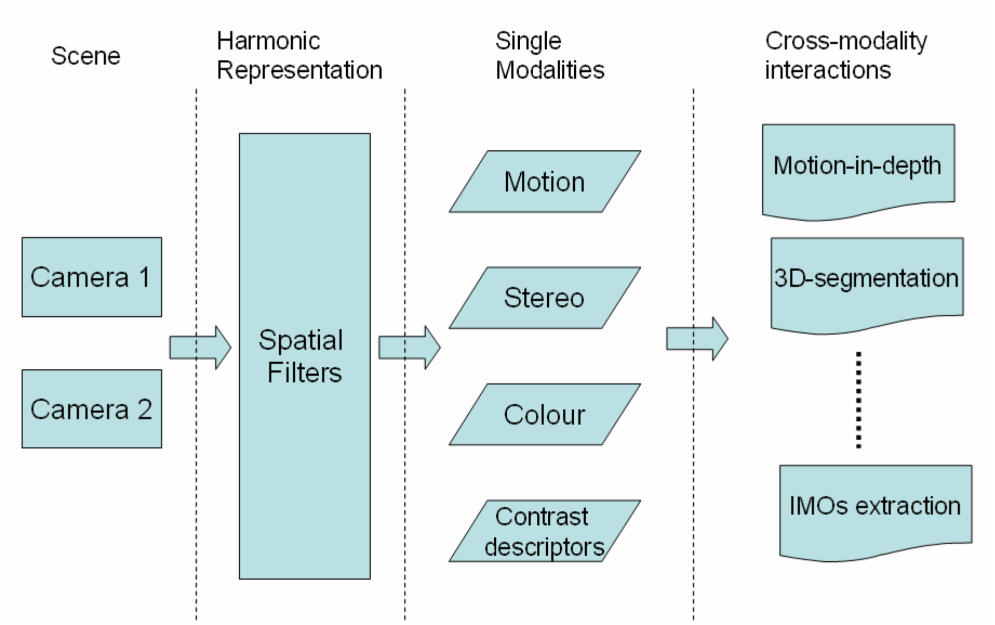


Fig. 1. Here we illustrate the different stages of the early vision system. The information obtained by the cameras is band-pass filtered leading to a harmonic representation where the outputs of the different spatial filters can be combined to extract different single modalities. These single modality cues can be merged towards cross-modality information.

This short report enumerates and briefly describes examples of these crossmodal interactions that are under investigation in the framework of DRIVSCO:

- Motion-in-depth. This is obtained combining disparity and motion cues.
- Clustering in 3D. Combining disparity and grey levels (or colours) we can cluster cues of the scene in the 3 dimensional space.
- Independent-Moving-Objects (IMOs). This can be obtained also combining motion and stereo or extracted directly from motion outliers.

- Phase for colour interpretation. Combining phase and colour helps to distinguish between lines and steps.

The next sections give illustrative examples of these kinds of crosmodal interactions.

2. Motion-in-depth

Considering jointly the binocular spatiotemporal constraints posed by moving objects in 3-D space, the disparity assigned to a point as a function of time is related to the trajectories in the right and left monocular images of the corresponding point in the 3-D scene. Therefore, dynamic stereopsis, implies the knowledge of the position of objects in the scene as a function of time. In general, the solutions to these problems rely upon a global analysis of the optic flow or on token matching techniques, which combine stereo correspondence and visual tracking. On the basis of the modelling work conducted by [Sabatini and Solari, 2004; Sabatini et al., 2002] UGE have demonstrated that dense motion-in-depth estimation can be obtained, without tracking, from binocular local measurements, provided that such measures are characterized by high significance and robustness. An analysis of the geometrical constraints influencing the reliability of motion-in-depth estimates has been conducted.

Motion-in-depth information can be directly extracted by using efficiently the harmonic representation (combining the outputs of spatial filters) or merging the motion and disparity cues. If we choose this second option we can calculate the temporal differentiation of the disparity or we can subtract right and left flows. These two options are being studied by UGE, investigating the design of local operators capable of providing a “full” description of 3D motion event (e.g., a spatially extended object moving in 3D), by projecting it into a subspace of elemental features (motion, disparity and orientation). Such description relies upon space and time phase information gathered from a band-pass spatio-temporal transformation of the binocular visual signal. Coherent stereo-motion correspondence constraints will be directly embedded in the structure of such visual operators, rather than being considered at a higher semantic level of data fusion.

Fig. 2 illustrates the proposed motion-in-depth information extraction scheme and the results obtained in a controlled lab sequence.

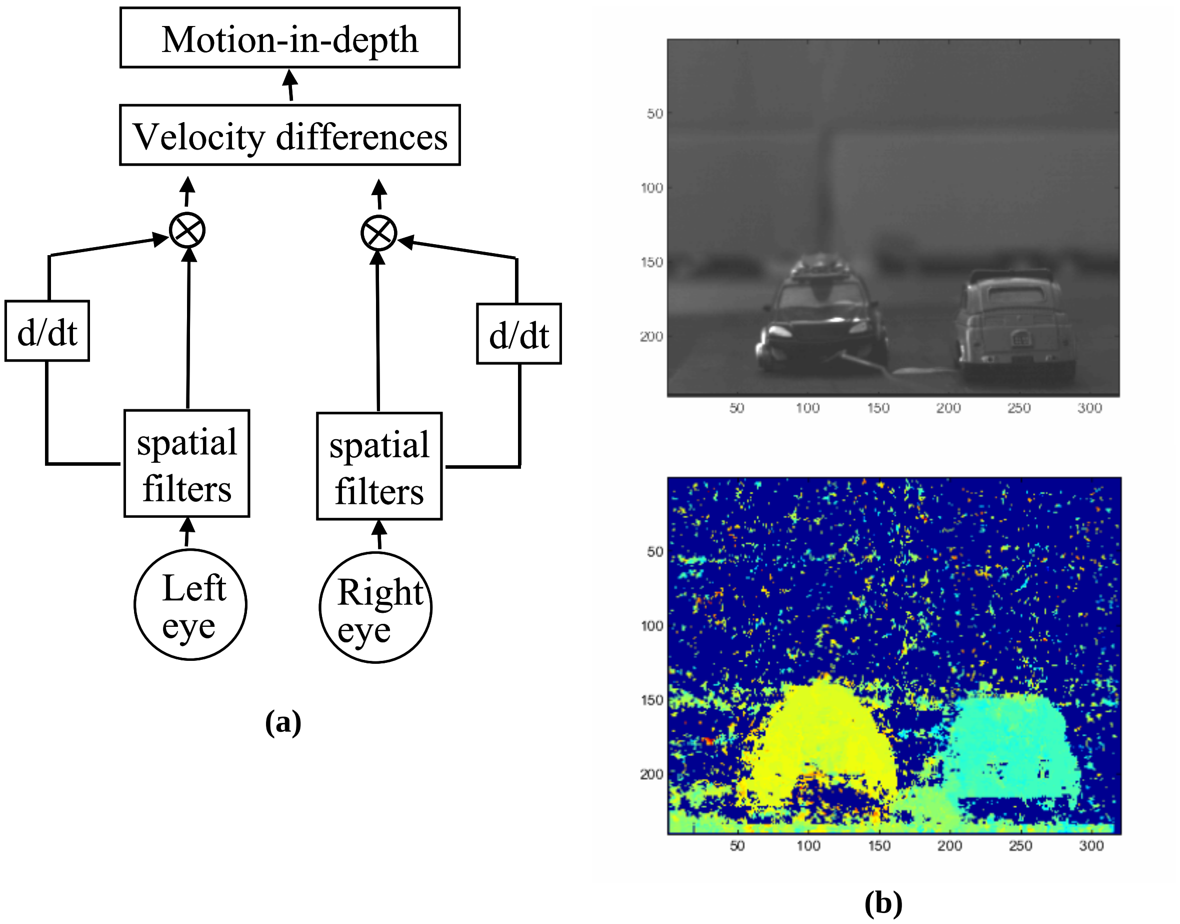


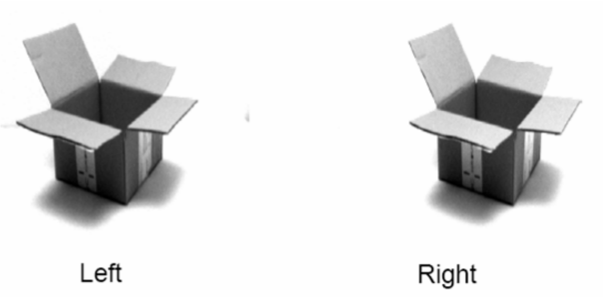
Fig. 2. Motion-in-depth extraction.(a) The basic architecture: information about interocular velocity difference is obtained directly by combining spatial convolutions of stereo image pairs and their temporal derivatives (calculated by linear interpolation of the temporal sequence). (b) Motion-in-depth extracted from a real "lab" sequence". Two toy cars are moving in opposite directions respect to the observer. The hue levels in the MID maps encode the motion-in-depth of the two cars: the yellowish region represents the car moving towards the observer, whereas the bluish region represents the car moving away. The background (dark blue) represents points discarded according to the confidence measure.

3. Clustering in 3D for surface segmentation

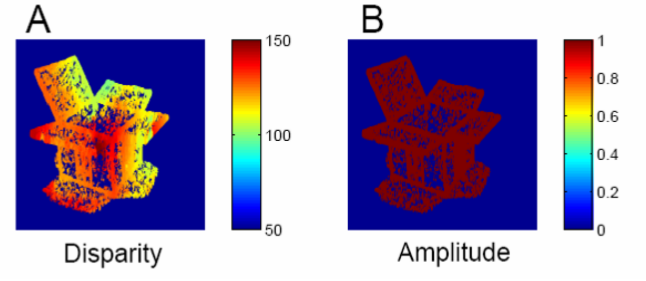
The identification and segmentation of surfaces or objects requires combined information from different visual attributes, such as binoculuar disparity, optic flow, texture, shape, and similarity. BCCN is investigating how to extract surfaces from stereo image pairs using superparamagnetic clustering [Opara and Wörgötter, 1998]. The method of superparamagnetic clustering represents image pixels by a Potts model of spins which interact such that neighbouring spins corresponding to similar pixels tend to align, given an appropriate similarity measure. Then, image segments are identified as clusters of aligned spins. We have extended this method to 3D images, i.e., stereo pairs and image sequences, by allowing spins belonging to different frames to interact.

Here, we use this method to segment stereo image pairs. The technique uses both information about gray-value pixel similarity and disparity information obtained from sparse and/or dense stereo algorithms. The disparities of the pixels are needed to localize the neighbours of pixels in other frames. By this mechanism, correspondences between frames can be established and stereo clusters can be conformed. Usually, sufficiently accurate disparity values are not available for all pixels. At these points, clustering is merely driven by spin interaction within a single frame. In this way, homogenous image regions for which no disparity information is given can be filled in using disparity information from the bounding edges.

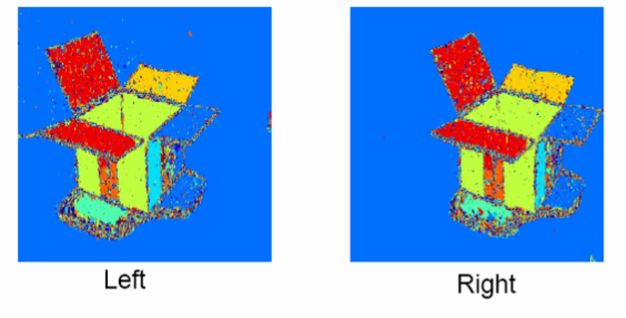
BCCN is studying this technique for a real stereo images, which shows a paper box from two different viewing positions, i.e., left and right (Fig. 3.a). The disparity map was computed using a dense stereo algorithm provided by KUL. We only consider those disparity values for which the corresponding amplitude value exceeds a certain threshold. The resulting amplitude map is given in (Fig. 3.b). In the clustering algorithm, spins are only allowed to interact with spins in the other image if their amplitude is equal to one, otherwise only interactions within a single frame are allowed. This has the advantage that (i) reliable disparity information can be used to establish correspondences between the stereo images, and (ii) homogeneous image regions for which the amplitude is zero can be filled in using 2D interactions. The resulting spin states of the box stereo pair are given in Fig. 3.c. The 3D surfaces could be extracted despite incomplete stereo information (Fig. 3.b). The salt and pepper noise visible in the segmented images is largely caused by erroneous disparity estimates, which lead to wrong correspondences. This effect is particularly strong at the edges. BCCN is also investigating how to incorporate highly accurate but sparse disparity information from primitives to improve image segmentation at the edges.



(a)



(b)

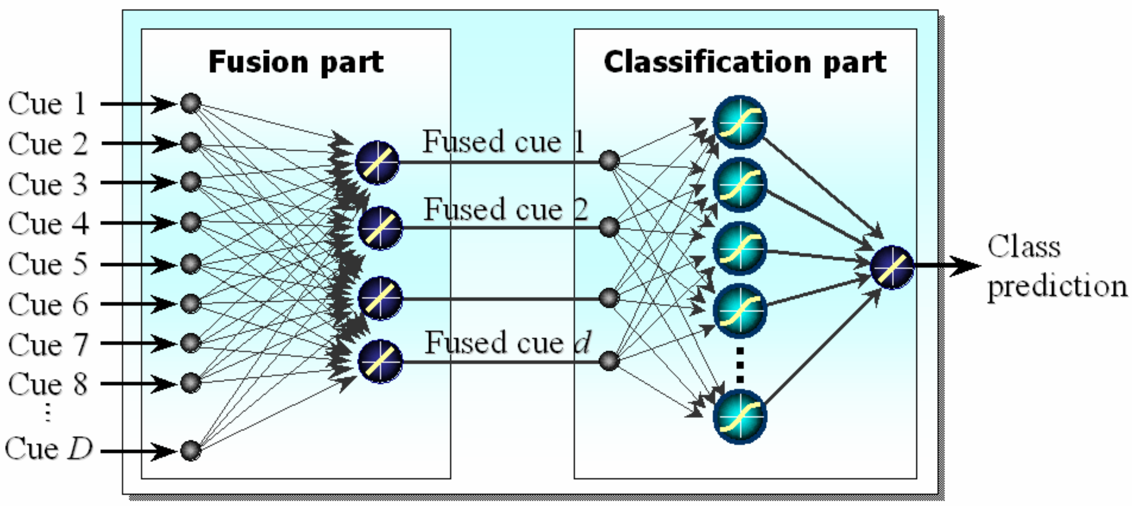


(c)

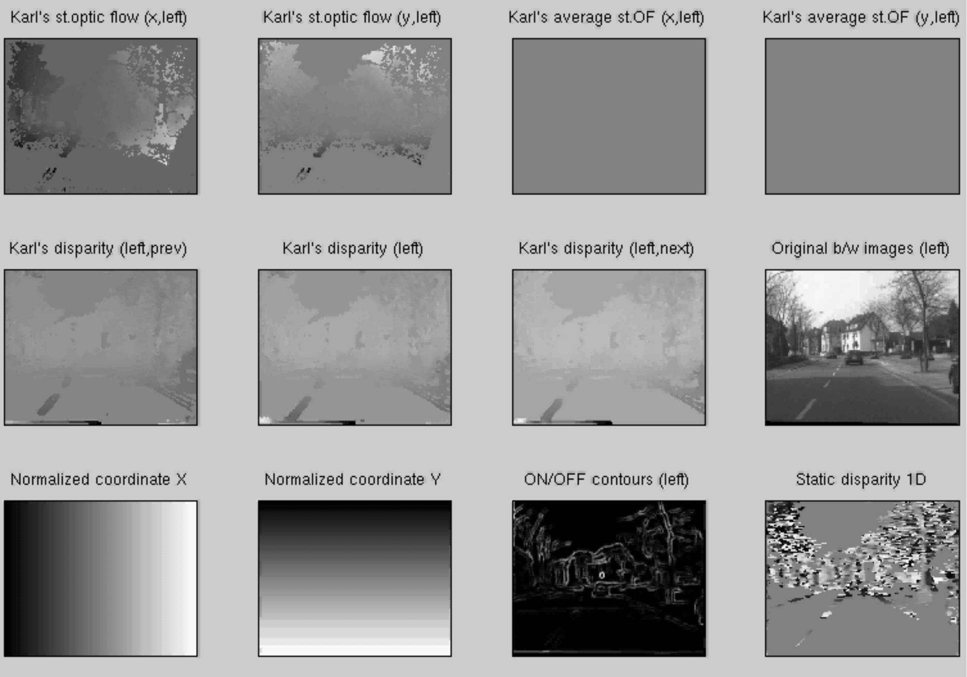
Fig. 3. Clustering in 3D for surface segmentation. a) Stereo image of an open box. b) Disparity and confidence measure of each estimation. c) Segmented surfaces.

4. Independent Moving Objects (IMOs)

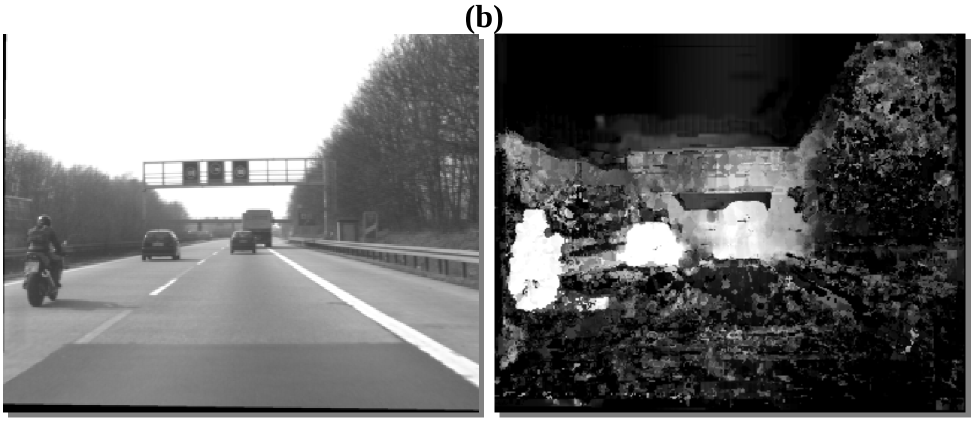
Independent moving objects represent very interesting entities in wide variety of applications (for instance automobile applications). The group of KUL is investigating how to use learning to directly learn efficient cues fusion (crossmodality interactions) for IMOs detection.



(a)


Karl's st.optic flow (x,left)
Karl's st.optic flow (y,left)
Karl's average st.OF (x,left)
Karl's average st.OF (y,left)
Karl's disparity (left,prev)
Karl's disparity (left)
Karl's disparity (left,next)
Original b/w images (left)
Normalized coordinate X
Normalized coordinate Y
ON/OFF contours (left)
Static disparity 1D

(b)

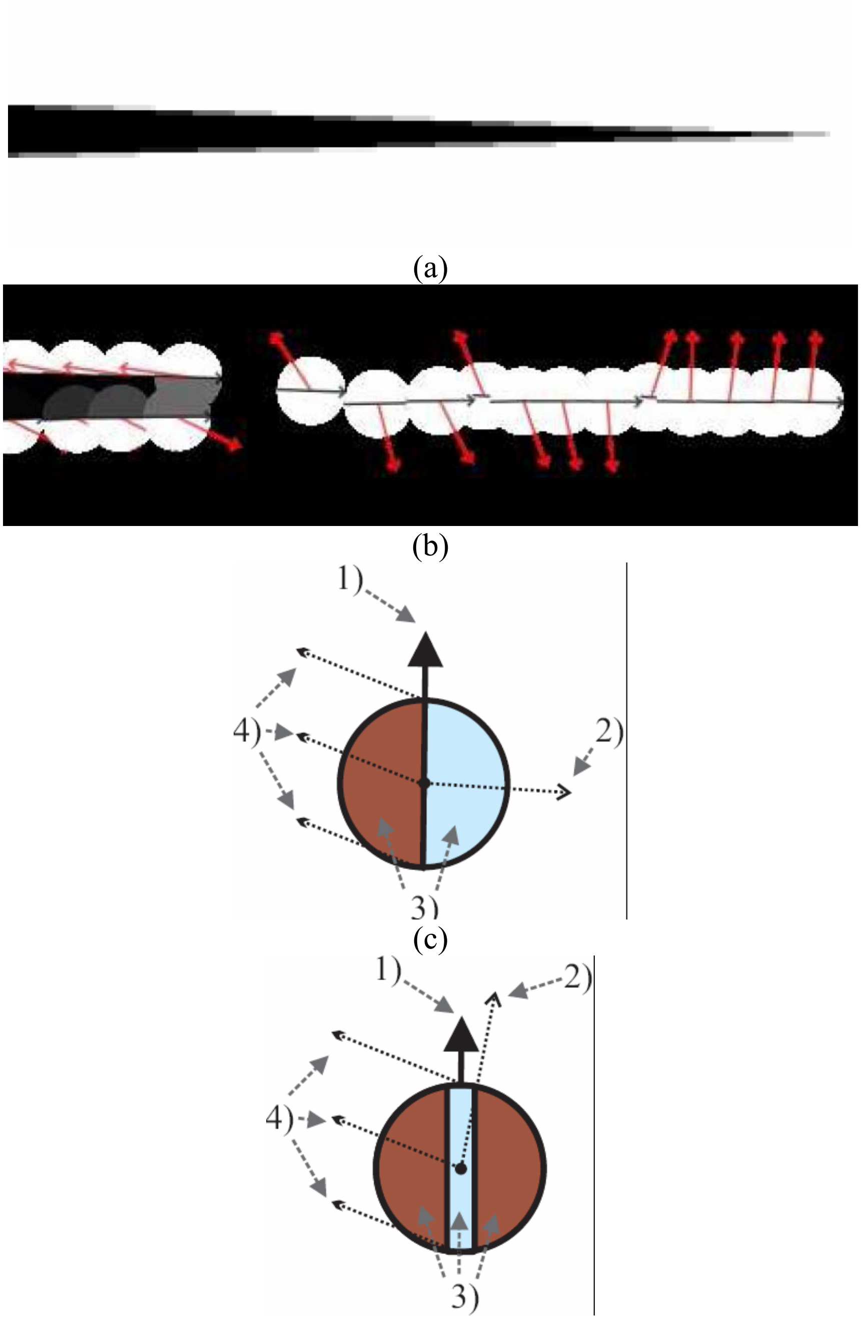


(c)

Fig. 4. Detecting IMOs by learning cues fusion. Learning crossmodality interactions. a) Neural network used for learning cues fusion. b) Different candidate cues. c) An example of IMO detection based on cues fusion.

5. Phase for colour interpretation

Phase analysis helps to distinguish between lines and steps and therefore helps in the colour interpretation of different areas of the scene. A phase change of π represents a step (which can have different colours in the two sides) while a π/2 change may represent a line and therefore it may be a feature onto the top of background uniform colour.



(a)

(b)

(c)

(d)

Fig. 5. Phase analysis for colour interpretation. A) Original figure, b) primitives extraction, c) Illustration of the symbolic representation of a primitive for a 1D interpretation, for a bright-to-dark step-edge (phase (φ ≠0)) edge primitive, d) Illustration of the symbolic representation of a primitive for a 1D a bright line on dark background (phase (φ≠π/2)), line primitive. 1) represents the orientation of the primitive, 2) the phase, 3) the colour and 4) the optic flow.

An important aspect of the condensation achieved by the primitives is that all main parameters can be derived from one property of the basic filter operations called.
This value expresses the minimal distance between two edges for them to be represented by two distinct primitives. Below this distance, one single line primitive will be extracted.

In Fig. 5 shows a narrow triangle for which two edges get closer until the vertex. Vertical sections of the local amplitude close to the vertex features only one maximum, whereas it splits into two distinct maxima further on, where the triangle is broader.
The *line-edge bifurcation distance* for a given scale is the minimal distance between two edges for them to produces two distinct maxima.

In order to represent accurately the colour structure of the edge, the colour information held by a primitive is composite. Also, we have seen that, depending on the phase, the primitive may express a step-edge or a line-like structure.
Consequently, the colour information is defined relatively to the phase.

If $\frac{\pi}{4} \le \phi < \frac{3\pi}{4}$, indicating an edge between two surfaces, then the colour information is sampled on the left and right sides of the central line (see Fig. 5.c).

Otherwise, the phase indicates a line, and the colour is sampled on the left and right sides, but also additionally the middle, to represent the colour of the line itself (see Fig. 5).

References

[OPA98] Opara, R., Wörgötter, F. (1998) A fast and robust cluster update algorithm. Neural Computation, 10 (6), 1547–1566.

[SAB04] Sabatini, S.P., Solari, F., Secchi, L.: A continuum-field model of visual cortex stimulus-driven behaviour: emergent oscillations and coherence fields. Neurocomputing 57: 411-433 (2004)

[SAB02] Sabatini, S., Solari, F., Andreani, G., Bartolozzi, C., and Bisio, G.M. A hierarchical model of complex cells in visual cortex for the binocular perception of motion-in-depth . In Proceedings of Advances in Neural Information Processing Systems 14 2002 Conference, pages 1271-1278, Vancouver, Canada.

Appendix F

Motion-in-depth. Geometrical considerations. Technical report.

Binocular Perception of Motion-in-depth from a Geometric Point of View

Fabio Solari, Giulia Gastaldi and Silvio Sabatini

February 14, 2007

Abstract

The perception of motion-in-depth relates to 2nd-order measures, which can be gained either by interocular velocity differences or temporal variations of binocular disparity. We can analyze, exploiting the projective geometry, the stereo vision system to obtain a range of possible values (pixel/frame) of the computed motion-in-depth (MID) for different scenarios. Thus, we can infer the required accuracy of the extracted velocity and disparity maps to obtain reliable MID maps.

1 Introduction to dynamic stereopsis

In many real-world visual application domains it is important to extract dynamic 3-D visual information from 2-D images impinging the cameras. One of this kind of problems concerns the perception of MID, i.e. the capability of discriminating between forward and backward movements of objects from an observer, having important implications for autonomous robot navigation and surveillance in dynamic environments. In general, the solutions to these problems rely upon a global analysis of the optic flow or on token matching techniques which combine stereo correspondence and visual tracking. Alternatively, in the light of behaviour-based perception systems, a more direct estimation of MID can be gained through the local analysis of the spatiotemporal properties of stereo image signals.

To better introduce the topic, let us briefly consider the correspondence problem in the stereo image pairs acquired by a binocular vision system. In a first approximation, the positions of corresponding points are related by a 1-D horizontal shift, the binocular disparity $\delta(x)$. Formally, the left and right observed intensities from the two eyes, respectively $I^L(x)$ and $I^R(x)$, result related as $I^L(x) = I^R[x + \delta(x)]$. Several researchers [Sanger, 1988] [Jenkin and Jenkin, 1988] proposed phase-based techniques in which disparity is estimated in terms of phase differences in the spectral components of the stereo image pair. Spatially-localized phase measures can be obtained by filtering operations with complex-valued quadrature pair bandpass kernels (e.g. Gabor filters [Gabor, 1946] [Daugman, 1985]), approximating a local Fourier analysis on the stereo images.

When the stereopsis problem is extended to include time-varying images, one has to deal with the problem of tracking the monocular point descriptions or the 3-D descriptions which they represent through time. Therefore, in general, dynamic stereopsis is the integration of two problems: static stereopsis and temporal correspondence [Jenkin and Tsotsos, 1986]. Considering jointly the binocular spatiotemporal constraints posed by moving objects in the 3-D space, the resulting dynamic disparity is defined as $\delta(x,t) = \delta[x(t),t]$, where $x(t)$ is the trajectory of a point in the image plane. The disparity assigned to a point as a function of time is related to the trajectories $x^R(t)$ and $x^L(t)$ in the right and left monocular images of the corresponding point in the 3-D scene. Perspective projections of a motion in depth leads to different motion fields on the two cameras, that is a temporal variation of the disparity of a point moving with the flow observed by the left and right views. The rate of change of such disparity provides information about the direction of MID and an estimate of its velocity. The following approximated expressions can be derived [Sabatini *et al.*, 2003]:

$$\frac{d\delta}{dt} \simeq \frac{\partial\delta}{\partial t} = \frac{\phi_t^L - \phi_t^R}{k_0} \simeq v^R - v^L \tag{1}$$

It is worthy to note that the approximations depend on the robustness of phase information, and the error made is the same as the one which affects the measurement of phase components around singularities [Fleet *et al.*, 1991] [Fleet and Jepson, 1990]. Hence, on a local basis, valuable predictions about MID can be made, without tracking, through phase-based operators which need not to know the direction of motion on the image plane $x(t)$.

2 Projective Geometry for Motion-in-depth

We briefly summarize the general relation between 3D world coordinates (X, Y, Z) and image coordinates (x^L, y^L) and (x^R, y^R). The model of the optical setup of the stereo system is shown in Fig. 1. Using a perspective projection model, a point $\mathbf{X}$ in the world coordinates is mapped onto image plane points $\mathbf{x^L}$ and $\mathbf{x^R}$ on the left and right cameras, respectively. For identical left and right focal lengths f_0, the image coordinates are:

$$\begin{aligned}
x^L &= f_0 \frac{(X+b/2)cos(\alpha^L) + Z\sin(\alpha^L)}{(X+b/2)sin(\alpha^L)cos(\beta^L) - Y\sin(\beta^L) - Z\cos(\alpha^L)cos(\beta^L)} \\
y^L &= f_0 \frac{(X+b/2)sin(\alpha^L)sin(\beta^L) + Y\cos(\beta^L) - Zcos(\alpha^L)sin(\beta^L)}{(X+b/2)sin(\alpha^L)cos(\beta^L) - Y\sin(\beta^L) - Z\cos(\alpha^L)cos(\beta^L)}
\end{aligned} \tag{2}$$

$$\begin{aligned}
x^R &= f_0 \frac{(X-b/2)cos(\alpha^R) + Z\sin(\alpha^R)}{(X-b/2)sin(\alpha^R)cos(\beta^R) - Y\sin(\beta^R) - Z\cos(\alpha^R)cos(\beta^R)} \\
y^R &= f_0 \frac{(X-b/2)sin(\alpha^R)sin(\beta^R) + Y\cos(\beta^R) - Zcos(\alpha^R)sin(\beta^R)}{(X-b/2)sin(\alpha^R)cos(\beta^R) - Y\sin(\beta^R) - Z\cos(\alpha^R)cos(\beta^R)}
\end{aligned} \tag{3}$$

We can define the horizontal disparity $d_x = x^R - x^L$ and the vertical disparity $d_y = y^R - y^L$, that establish the relations between a world point $\mathbf{X}$ and its associated disparity vector $\mathbf{d}$.

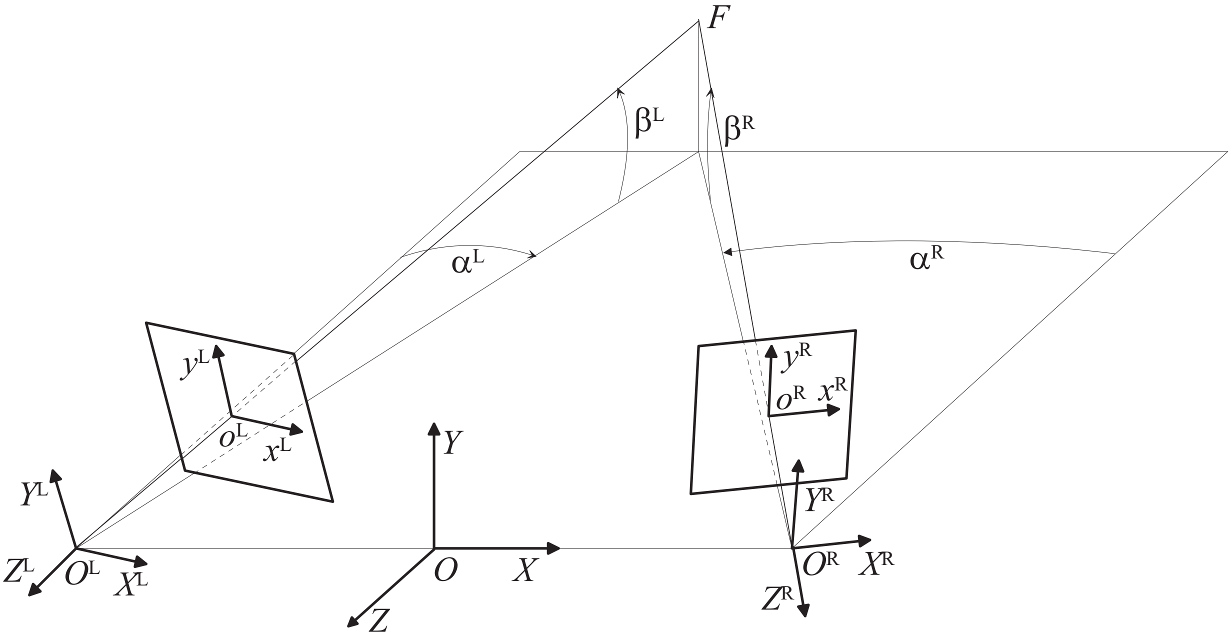


Figure 1: Optical setup of a stereo system. Left and right cameras: (X^L, Y^L, Z^L) and (X^R, Y^R, Z^R). Left and right image planes: (x^L, y^L) and (x^R, y^R). Left and right focal lengths: $O^L o^L = O^R o^R$, named f_0 in the text. Optical axes $O^L F$ and $O^R F$ are adjusted to fixation point F. The baseline b is denoted by $O^L O^R$, the slant angles by α^L and α^R and the tilt angles by β^L and β^R. The camera optics are modelled as a perspective projection.

3 Discussion

We have analyzed two different scenarios to obtain a range of possible values (pixel/frame) of the computed MID and the geometrical constraints for its reliability. In particular, we considered an outdoor scenario, related to a car moving on a road, and an indoor scenario, related to a laboratory setup. The parameters of the binocular camera setup are: (i) for the outdoor scenario: baseline=0.55 m; focal length=25 mm; pixel pitch=6.5 μm; and (ii) for the indoor scenario: baseline=0.08 m; focal length=8 mm; pixel pitch=11 μum.

For the sake of simplicity, in the following we restrict the analysis to the plane (X, Z), that is $Y = 0$, with the tilt angles $\beta^L = \beta^R = 0$ and the slant angles $\alpha^L = \alpha^R$.

Exploiting the equations 2 and 3, we can infer the areas in the horizontal plane (X, Z), where the MID is reliable, by imposing constraints on the values of the stereo and motion visual features. In particular, we impose the following constraints: MID values > 1 pixel/frame; optic flow values < 30 pixels/frame; disparity values < 50 pixels. These constraints mean that we should be able to match image point pairs 50 pixel apart and to perform a reliable optic flow difference in a range of [-30,+30] pixels/frame with an uncertainty of 0.25 pixel/frame.

In the outdoor scenario Fig. 2 and Fig. 3, there is a large area of reliable MID in front of the car only for high speed and for vergent cameras.

Similar conclusions can be gathered for the indoor scenario, see Fig. 4 and Fig. 5.

An example of outdoor scenario is shown in Fig. 6a: in this case motion-in-depth is not detected because the binocular sequence is acquired by cameras with parallel optical axes (cf. the first row of panels of the Fig. 2). An example of reliable MID detection is shown in Fig. 6b: in this case, the indoor scene is observed by vergent cameras (cf. the third row of panels of the Fig. 4).

References

[Daugman, 1985] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Amer.*, A/2:1160–1169, 1985.

[Fleet and Jepson, 1990] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 1:77–104, 1990.

[Fleet *et al.*, 1991] D.J. Fleet, A.D. Jepson, and M.R.M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

[Gabor, 1946] D. Gabor. Theory of communication. *J. Inst. Elec. Eng.*, 93:429–459, 1946.

[Jenkin and Jenkin, 1988] A.D. Jenkin and M. Jenkin. The measurement of binocular disparity. In Z. Pylyshyn, editor, *Computational Processes in Human Vision*. Ablex Publ., New Jersey, 1988.

[Jenkin and Tsotsos, 1986] M. Jenkin and J.K. Tsotsos. Applying temporal constraints to the dynamic stereo problem. *CVGIP*, 33:16–32, 1986.

[Sabatini *et al.*, 2003] S.P. Sabatini, F. Solari, P. Cavalleri, and G.M. Bisio. Phase-based binocular perception of motion in depth: Cortical-like operators and analog VLSI architectures. *EURASIP J. on Applied Signal Proc.*, 7:690–702, 2003.

[Sanger, 1988] T.D. Sanger. Stereo disparity computation using Gabor filters. *Biol. Cybern.*, 59:405–418, 1988.

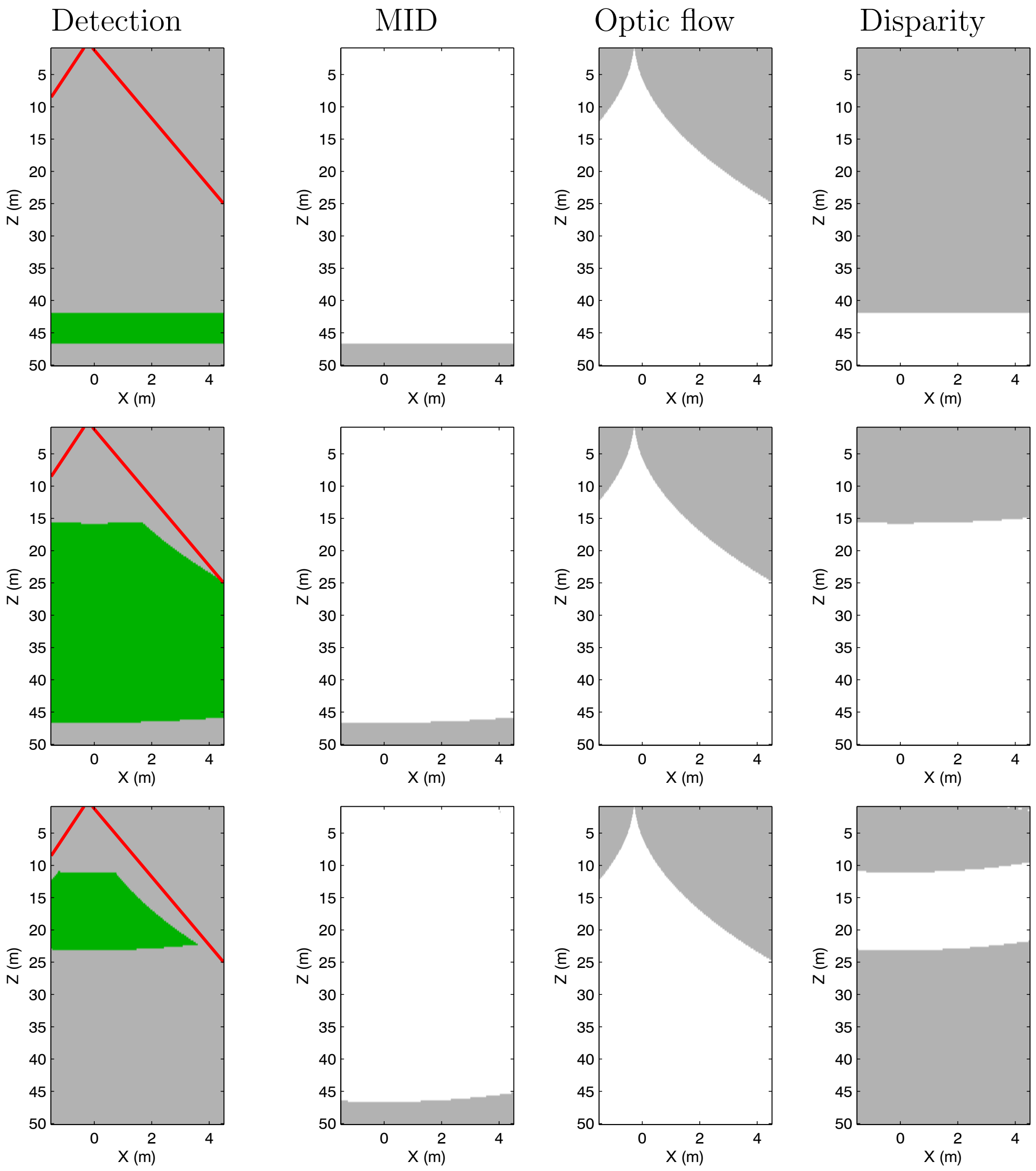


Figure 2: Each panel represents the top view of a road: the car position is at top-left, the car moves along the road (Z axis) and the red lines mark the boundary of the camera field of view. The cameras with parallel optical axes are considered in the first row of panels, in the second row the fixation point is at 25 m and in the third row at 15 m. The car speed is 90 km/h. The gray areas in each panel denote the zones where the feature constraints are not met. The green areas denote the resulting reliable detection regions of the MID.

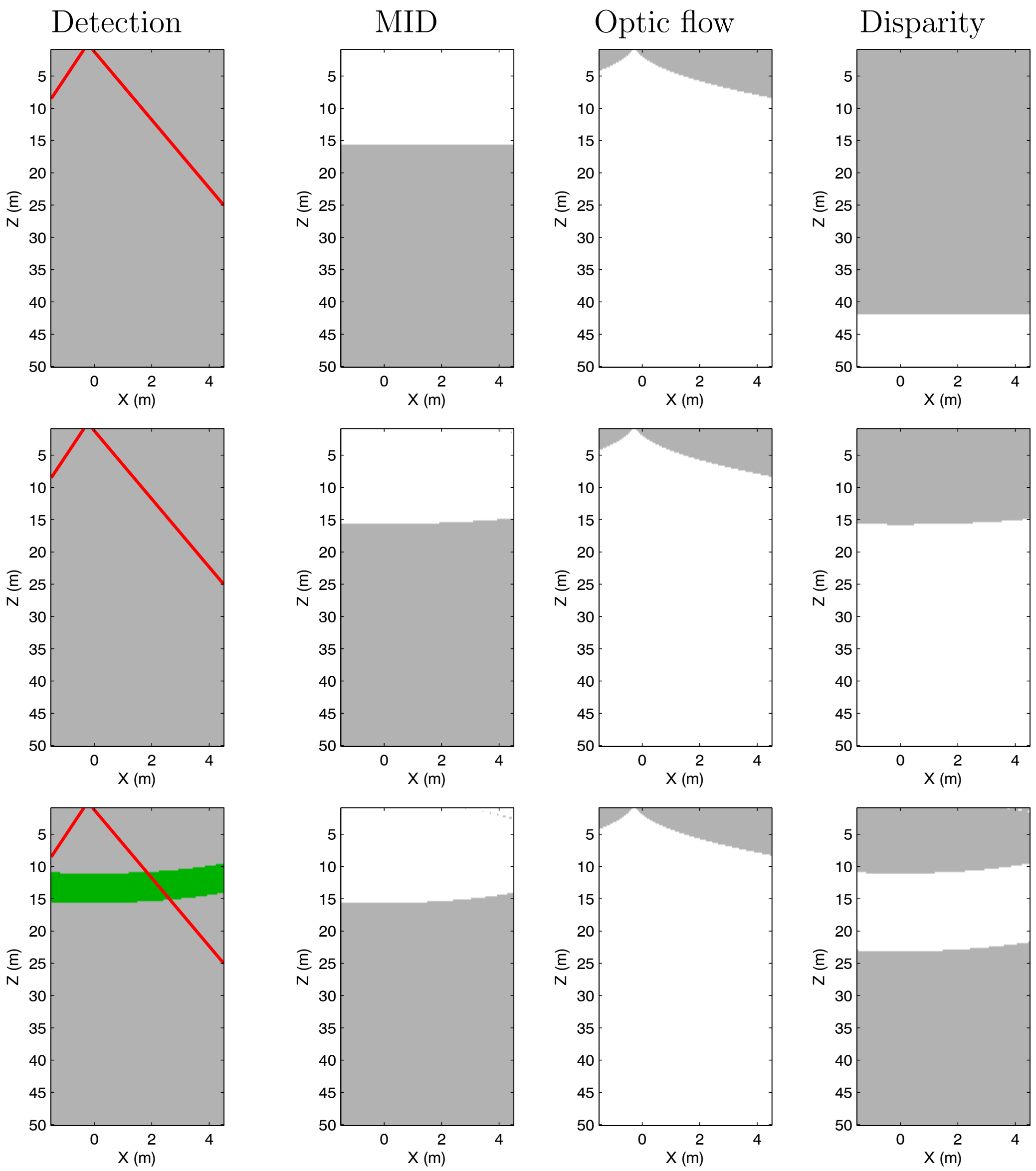


Figure 3: Same as Fig. 2, but with a car speed of 10 km/h.

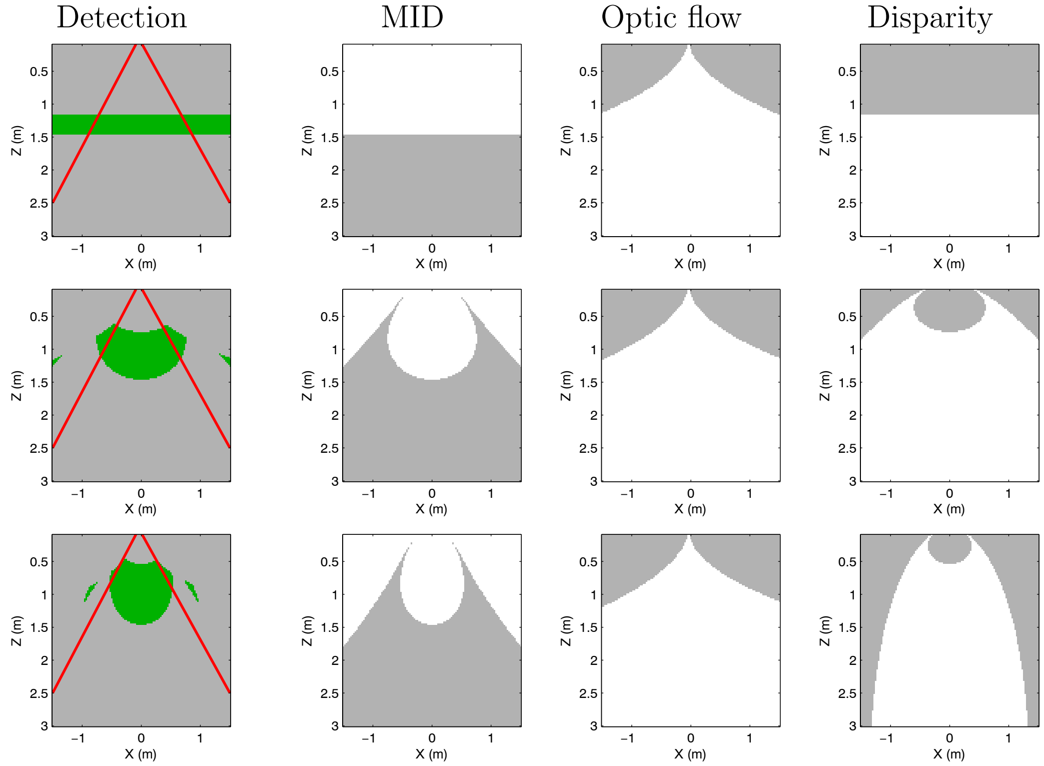


Figure 4: Each panel represents the top view of laboratory setup: the camera position is at center, the motion is along the Z axis and the red lines mark the boundary of the camera field of view. The cameras with parallel optical axes are considered in the first row of panels, in the second row the fixation point is at 2 m and in the third row at 1 m. The speed is 0.9 m/s. The gray areas in each panel denote the zones where the feature constraints are not met. The green areas denote the resulting reliable detection regions of the MID.

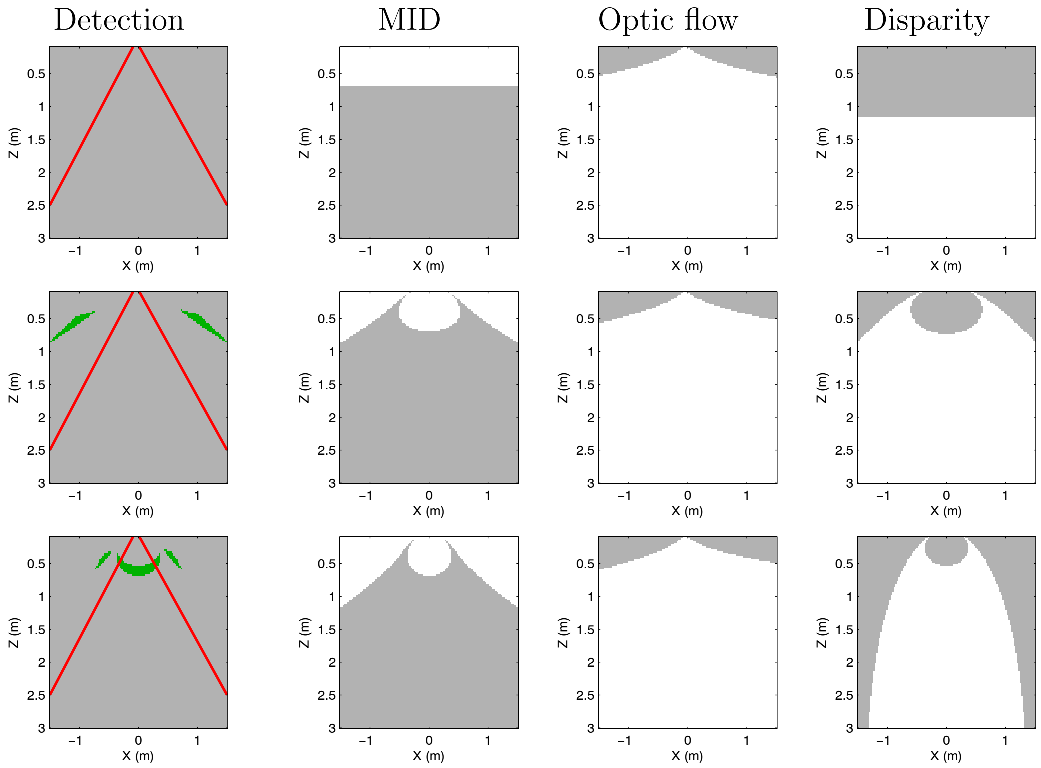
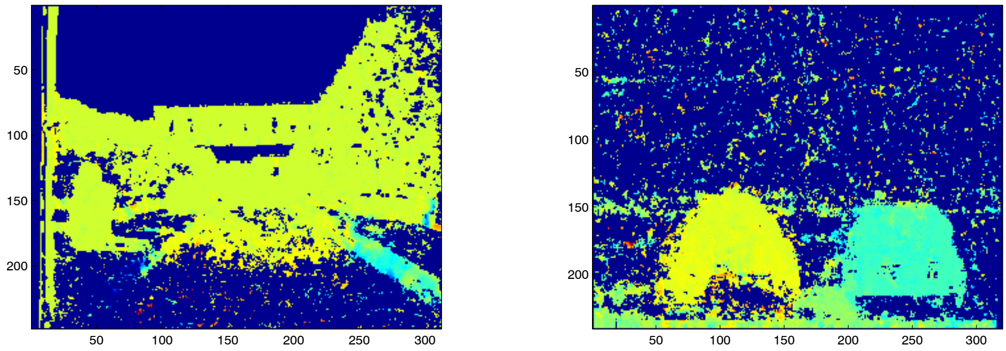


Figure 5: Same as Fig. 4, but with a speed of 0.2 m/s.



(a) (b)

Figure 6: Two examples of motion in depth maps obtained from an outdoor scenario acquired with parallel axes (a) and from an indoor scenario acquired with vergent cameras. The hue codes the motion-in-depth information: warm colors represent motion towards the observer, whereas cold colors code motion away from observer. The motion of the overtaking motorcycle in (a) cannot be distinguished by the egomotion. In (b) two toy cars are moving in opposite directions respect to the observer. The background dark blue represents points discarded according to the confidence measure. The few still present error points do not impair the interpretation of the MID map.