

Satellite Image Analysis and Processing Tools to be used in Air-Pollution Forecast and Simulation Systems

Alfons Salden

▶ **To cite this version:**

Alfons Salden. Satellite Image Analysis and Processing Tools to be used in Air-Pollution Forecast and Simulation Systems. Systems Analysis Modelling Simulation, 2000, 39 (1). inria-00532726

HAL Id: inria-00532726

https://inria.hal.science/inria-00532726

Submitted on 18 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Satellite Image Analysis and Processing Tools to be used in Air-Pollution Forecast and Simulation Systems *

Alfons H. Salden

INRIA Rocquencourt (AIR), Domaine de Voluceau BP 105
F-78153 Le Chesnay Cedex, France

e-mail: Alfons.Salden@inria.fr,alfons@first.gmd.de

March 11, 1999

Abstract

We present image analysis and processing tools that can be useful in the extraction of stable and reproducible input data from satellite images for meso-scale meteorological systems. The tools used consist of modern geometric machines capable in quantifying Volterra processes to which the modelled and observed physical fields are subjected, and filter operations with respect to and coupled to those Volterra processes. In particular we apply these tools to satellite images in the visual, infrared and water vapour spectrum in order to retain segmentations and quantifications of correlated physical fields which in turn can be used in air-pollution forecast and simulation systems. These physical fields concern the heat currents due to reflected solar irradiance in the short- and long-wave band. In this context the analysed and processed radiances serve the estimation of the short- and long-wave band transmission function.

1 Introduction

The output of air-pollution forecast and simulation systems heavily depends on the physical principles incorporated into the related mathematical and numerical models. These principles are normally captured by various conservation laws for the following physical fields: air mass density, linear momentum density, potential temperature, mass fractions or concentrations of chemical constituents of air, fluxes of phase transformations of those constituents, and fluxes of chemical reactions between them. These conservation laws are in turn expressed in terms of a nonlinear dynamical system with respect to those physical fields. Such a dynamical system then consists of a system of partial differential equations and/or integral equations with a set of initial-boundary value conditions. An important problem related to dynamical systems concerns the initialisation over space and time of the considered physical fields and their dynamics. An

*This report has been submitted to "Systems Analysis Modelling Simulation" containing proceedings of "4-th ERCIM Environmental Modelling Group Workshop", February 1999

initialisation of the system, namely, requires numerically stable and physically reproducible input data representations. Here physical reproducibility of the input data refers to the more or less independence of those representations under active or passive transformations of the physical fields. Passive transformations concern transformations on the level of the mathematical model, whereas active ones concern physical transformations between fields and measuring device (e.g. rotations of satellite around its optical center or nonlinear scaling of filter characteristics). Given an ensemble of measuring devices for the physical fields and their dynamics, each with their own resolution and sensitivity characteristics yielding different primal input data representations for possibly different meso-scale meteorological systems, the question arises how to retain (partially) equivalent input and output data representations for these systems. In particular the initialisation of the input data concerns local as well as boundary information about the physical fields and dynamics before and during model run.

In order to set stable and reproducible input data for the physical fields processed in an air-pollution forecast and simulation system various data assimilation procedures exist among which those based on objective analysis, dynamic initialisation and normal mode initialisation [1]. In order to improve initialisation of inhomogeneous physical fields over steep topography a so-called adjoint method has recently been introduced [2]. All these initialisation procedures aim at generating these data on the basis of ground, airborn and satellite observations with respect to the physical fields.

In respect to the physical principles underlying air-pollution forecast and simulation models and the initialisation of their related systems an analysis and processing of the observed physical fields is indispensable. Image analysis and processing can not only play a crucial role in this initialisation issue but also in the derivation of other physical conservation principles by catching more complex processes. The main research objectives, therefore, in this area of environmental modeling are the development, the integration and the validation of advanced image analysis and processing tools for air-pollution forecast and simulation systems.

The imaging tool requirements are subjected to accuracy and computational complexity requirements put forward by a meso-scale meteorological system. E.g., if the input data representations of the physical fields used by a system should be of a different spatio-temporal dynamic resolution and complexity than the primal input data representations delivered by the measuring devices reading out those fields, then there should be tools at our disposal that bring the resolution and complexity of these representations in line. These tools should then also be consistent with and coupled to the spatio-temporal dynamics of air-pollution measured on the ground, airborn or by satellites. In particular we head for developing tools in line with meso-scale meteorological models, such as **GESIMA** [3] and **REWIMET**, capable of generating accurate input data of various physical fields (for the related system) retrievable from satellite images. The satellite images in mind are e.g. **Meteosat** images in the visual, infrared and water vapour absorption spectrum, and **POLDER** images providing additionally polarisation parameters with respect to aerosols.

In order to meet the above requirements we apply a so-called dynamic scale-space paradigm [4, 5] that combines modern geometry, dynamical system theory and finite element methods. In image analysis and processing there were developed various scale-space techniques to quantify the formation of an image [4, 5, 6, 7]. Characteristic for all these techniques is to adopt a certain topology, geometry or dynamics on the observed image possibly consistent with that of the induced external electromagnetic field activity. Another feature of these techniques is to filter the coherent image such that one retains a restoration, enhancement or dynamic ordering of coherent image

structures such as edges. Such filter operations normally result in a data reduction and compression. The opposite operation is also possible but requires knowledge of the dynamics of the physical fields on a subpixel level which is in general absent, but gives rise to a so-called ill-posed problem. How to derive from coarse scale information about the physical field dynamics a physical sensible splitting process on a lower than pixel scale will remain one of the outstanding and unresolved questions for the future. Whether or not the latter will appear to be achievable, the construction of filters or structuring elements consistent with the induced electromagnetic field activity will still remain of major importance.

Our paradigm for the construction of filters is data and concept driven. Let us briefly elaborate on the fundamental principles involved in the paradigm. The past decade several methods in image analysis have been proposed for the extraction of physical fields, in particular (apparent) motion fields [8], [9], [10], [11], [12]. All these approaches are mainly based on the assumption of (point) correspondences [13] possibly related to integral invariants [11]. The latter requirement of correspondence or identifiability of spatial configurations is bounded to be violated in general, for a variation of spatio-temporal resolution properties of a vision system with respect to the physical fields and non-diffeomorphic transformations of them cause dramatic topological chances in the description and interpretation of the spatial sections of the images of the physical fields. In order to overcome such field analysis problems filtering schemes of the spatio-temporal images of of the physical fields have been proposed satisfying field (un)committed scale-space principles [14], [7] and [4]. These schemes do not intend to guarantee point correspondence, because the inner spatio-temporal scales remain always limiting factors and more importantly the scene dynamics certainly cannot merely be captured in terms of smooth time-dependent spatial deformation fields of one spatial reference configuration or spatial input image into another. In terms of modern geometry the dynamics of the physical fields consists not only of a total integrable deformational part, but also of non-exact (non-integrable) parts due to the breaking of symmetries such as rotational or translational symmetry caused by so-called Volterra processes (insertion and removal processes). Such processes one encounters frequently in defect theory and gauge field theories [15], [16]. Studying these theories one realises that the field equations derived from a variational principle may not model processes, such as defect dynamics that are dissipative in the macroscopic thermodynamic sense, and occur for open systems [15]. From a mathematical point of view Kadi's remark is also true, for in general the evolution equation describing defect dynamics does not have a Lagrangian formalism [17]. However, considering a bounded region of the space-time history of a system a variational approach and a (global) conservation principle are still justifiable. The only problem to be tackled in the context of retrieving suitable input and output data representations then is to derive a consistent similarity group action ensuring the complexity and resolutions of these data requested by the system and the user. Updating the spatio-temporal history of the system the evolution equations of defect dynamics are distorted. This distortion can subsequently be considered as a manifestation of the updated system dynamics.

The dynamic scale-space paradigm, however, is just developed for obtaining a multi-scale representation of physical fields on a bounded region of space-time, corresponding to the induced electromagnetic field activity, that is partially equivalent above some scale for a whole ensemble of slightly distorted images of those fields. These distortions can be due to fluctuations in the physical fields themselves or due to variations in which a vision system has acquired the input data (differences in resolution and sensitivity properties, acquisition protocols, etcetera). Our dynamic scale-space paradigm

is based on two insights acquired in biology, computer vision, physics and mathematics [4, 5, 6]. Firstly, directed and oriented circuits or (path) integrals over a set of physical observations can reveal the set of rules involved in image formation and so in the induced external electromagnetic field activity. One may conceive that set of rules then as the primal input and/or output data representations for (meso-scale) meteorological systems. Secondly, dynamic scale-spaces of an ensemble of such sets of rules for slightly perturbed images contain stable and reproducible (partially) equivalent representations of this set above some scale. Here scale refers to a particular volume element spanned by a stencil of pixels, voxels or groups of them operationalising some current, and used in the creation of the dynamic scale-spaces by sampling, analysing and processing the input image with it. Above the mentioned scale the representations are slightly affected by noise effects, preserve the most salient coherent image structures such as watersheds, crestlines, edges, plateaus, divides and ridges, channels and ruts, and defect lines [4, 5, 6, 18].

One of our main goals in this paper is to present our dynamic scale-space paradigm delivering suitable machines to generate stable and reproducible meso-scale input data for air-pollution forecast and simulation systems (see (Section 2)). We make explicit the physical principles underlying the paradigm; we introduce the modern geometric and statistical physical concepts used in this paradigm for quantifying and qualifying the induced external electromagnetic field activity. Next we point out how our paradigm can be applied to analyse and process the system output, to compare different model setups and to bring observations, models and systems in line (see Section 3.1). The other main goal is to apply the image analysis and processing tools to generate stable and reproducible input data of the radiance field in the visual, infrared and water vapour absorption spectrum using **Meteosat** images (see Section 3.2). We show how to obtain a multi-scale segmentation of those physical fields controlled by the dynamics of those fields themselves. The latter analysis and processing serve various modules computing the dynamics of different physical fields in meso-scale meteorological systems such as **GESIMA** and **REWIMET**. Especially, the parametrisation of the heat currents at the model domain bottom and top, i.e., the solar irradiation and the blackbody's radiation of the atmosphere's boundary layer, are addressed for they directly influence all other modules in the system, e.g., the momentum transport module. The analysed and processed radiances observed by the satellite are intended to be used for the computation of of the transmission functions in those wave-bands to be fed into the system.

2 Dynamic Scale-Space Paradigm

We treat our dynamic scale-space paradigm that is (un)committed to the induced image formation, i.e., a paradigm that listens only to the geometry of the vision system or in addition also to the induced dynamics on the vision system inflicted by the external electromagnetic field. A proper description of this geometry and dynamics is indispensable before a well substantiated image analysis and processing can be carried out that retrieves stable and reproducible equivalence relations of of the input image, i.e., properties of the image invariant under a specific transformation group.

2.1 Initialisation of Equivalences

Let us first mathematically model an image I as a mapping of a vector-valued energy-density field (current) of the external electromagnetic field activity M onto a vector-valued density field for the activity N of the vision system:

Definition 1 *An image I is defined by a mapping:*

$$I : M \rightarrow N,$$

where M is a possible state of the external electromagnetic field and N is a possible state of the vision system.

In the above definition M is the present external electromagnetic field activity fallen onto the vision system, whereas N can be conceived as the induced vision system activity given the transducer-mapping I. The above image can be a superposition of several other images of physical observations. The camera system could, for example, analyse the fine structures of the energies as function of the frequency of the light fallen onto the set of detector arrays [19], as function of the chirality (left- or right-handedness) [20] and/or as function of their polarisation states both energetically and angular-temporally [21]. It could perform such measurements for a temporal sequence of stereo-images I_L and I_R. In this context the bi-reflective distribution function [21] does not need to satisfy any symmetry relations. Furthermore, the perceived geometric and topological equivalences of these fine structures of an image depend on the illumination field, reflection and absorption properties of surfaces and media between sources and surfaces and between surfaces and camera system.

Now our dynamic scale-space paradigm is based first of all, as mentioned in the introduction, on the derivation of a set of equivalences of the image (Definition 1) that are invariant under a gauge group, a group of transformations freely acting on the underlying physical objects being the vision system and the external electromagnetic field activity. E.g., the sensitivity and resolution properties together with the view of an ensemble of vision systems may be different asking for adaptive analysis and processing of the primal data such that weakly equivalent input data representations can be realised.

Definition 2 *A gauge group on image (Definition 1) is a group of transformations of the external electromagnetic density field M, the mapping I and the system density field N.*

A discretisation of image (Definition 1), i.e., a sampling of the vector-valued energy-density field N, invariant under such a gauge group can be obtained by aggregates of detectors satisfying related discrete gauge symmetry group requirements. Among the gauge groups considered in computer vision and mathematical morphology are [4, 5, 6, 7, 18]:

- Groups of Euclidean movements: the semi-direct product of the translation group and the rotation group on Euclidean space,
- Groups of (unimodular) affine movements: the semi-direct product of the translation group and the general linear group on (unimodular) affine space,
- Groups of projective movements: central perspective transformations of planar objects onto planar imaging domain can be covered by the projective group on the plane,

- Groups of Galilean movements: the semi-direct product of the group of temporal shifts and the group of Euclidean movements,
- Groups of Lorentzian movements: the group of spatio-temporal transformations preserving the Minkowski metric $ds^2 = dx^2 - dt^2$,
- Groups of anamorphoses or homotopies: the product of the group of monotonic spatio-temporal homogeneous grey-value transformations, and one of the classical groups above,
- Groups of diffeomorphisms: groups of image deformations in a spatio-temporal as well as a dynamical sense,
- Groups of similarity transformations or scale transformations: the groups are defined by a spatio-temporal exchange principles for image equivalences of M, N and I, that are captured in terms of a system of partial differential and/or integral equations (the similarity solutions form solutions of the related Cauchy problems) (see Section 2.2).

The action of gauge group G on M, N and I then can either yield, as discussed in the introduction, a passive or active transformation of the induced state of the vision system N and of the derived observations. Passive transformations concern then mainly deformations of the mathematical representation of N, M and I, whereas active transformations include non-diffeomorphic transformations of them. Except for the last group of transformations above all the other groups generate passive transformations. However, most of these groups are also active but can readily transformed into passive ones. The similarity group, however, consists of active transformations (morphisms) of the image that cannot be viewed as deformations of an initial image, i.e., M, N and I. Besides that active transformations are caused by a particular image analysis and processing paradigm carried out by the vision system, such as a similarity group action, they can also come about by an (discontinuous) inhomogeneous Lie group action on M, N and I due to morphisms of the resolution and sensitivity characteristics and position in space-time of the vision system, and due to perturbations of the external electromagnetic energy density fields. Groups of similarity transformations, however, allow us to forget partially about morphisms caused by such perturbative active transformations, and to cling to the relevant induced external electromagnetic field activity. For our purposes we consider in the sequel the Galilean transformation group and the groups of similarity transformations consistent with the former group. A reason for our choice is that in meso-scale meteorological modelling the conservation of total energy or heat (conservation of heat currents) within a closed region of space-time is one of the basic principles used to formulate a heat module in a related system.

The above mentioned equivalences of the (induced) external electromagnetic field activity come about after setting up a so-called frame field, metric and connection consistent with a specific gauge group (Definition 2). In the following we present these geometric attributes to arrive at gauge invariant physical objects represented by the induced external electromagnetic energy density fields N to be used in the dynamic exchange principles proposed in (Section 2.2).

The product space of M and N in image (Definition 1) can be associated a frame field [4, 5, 6]:

Definition 3 *A frame field* $(v_p) = (x_i, e_j, l_k)$ *is a realisation of a section of the tangent bundle* $T(M \times N)$ *of the image (Definition 1).*

By exponentiating the frame vector fields x_i, e_j and l_k one obtains a parametrisation of space-time occupied by the vision system and a parametrisation of the dynamical aspects perceivable by the vision system. The first two sets of frame vector fields form a section of the Galilean group on space-time. Besides a frame field there's also realised a differential frame field to operationalise the observations through the frame fields.

Definition 4 *A differential frame field $(dv^p) = (dx^i, de^j, dl^k)$ of frame field (Definition 3) is a section of the cotangent bundle $T^*(M \times N)$ on the image (definition 1).*

A frame field and its differential counterpart then satisfy not necessarily a duality constraint:

Definition 5 *A frame field (Definition 3) and its differential counterpart (Definition 4) are their duals, if and only if:*

$$dv^p(v_q) = \delta^p_q,$$

where δ is the Kronecker delta-function.

In order to relate and compare local states of the vision system besides a frame field (Definition 3) and its differential counterpart (Definition 4) also a metric g and a connection Γ are quite useful [4, 5, 6]:

Definition 6 *A metric tensor g is a (non-degenerate) bilinear form on $T(M \times N)$ such that*

$$g(v_p, v_q) = \delta_{pq},$$

where δ is the Kronecker delta function.

Note that in a particular reference frame, e.g., a global Cartesian coordinate frame the components of the metric may still be functions of coordinates (external observer), whereas on the level of the frame field (Definition 3) these space dependencies are not felt (internal observer).

Definition 7 *A connection Γ on image (Definition 1) is defined by one-forms o^q_p on the tangent bundle $T(M \times N)$:*

$$\nabla^\Gamma v_p \quad = \quad o^q_p \otimes v_q; \;\; o^q_p(v_r) \in \mathrm{K},$$

where $\otimes$ denotes the tensor product, and ∇^Γ is the covariant derivative on image (Definition 1), and K *a field of scalar numbers representing physical observations such as energies and rotations.*

Here the covariant derivative operator ∇^Γ is equivalent to taking an adapted differential operator d without specifying yet the frame vector field in the direction of which this derivative is taken. The covariant derivative of a tensor t of type (k, l) yields a new tensor of type $(k, l+1)$. E.g., the frame vector fields in (Definition 3) are lifted to $(1, 1)$ tensor fields that yield a representative of the Lie group action only after insertion of a the frame vector field into o^q_p. Now the component functions $t^{u_1 \cdots u_l}_{v_1 \ldots v_k ; r}$ of the covariant

derivative operator ∇ with respect to frame field vector v_r of a tensor t of type (k,l) are given by:

$$t^{u_1\ldots u_l}_{v_1\ldots v_k;r} = v_r t^{u_1\ldots u_l}_{v_1\ldots v_k} + \sum_{s=1}^{l} t^{u_1\ldots u_{s-1}pu_{s+1}\ldots u_l}_{v_1\ldots v_k}\Gamma^{u_s}_{rp} - \sum_{r=1}^{k} t^{u_1\ldots u_l}_{v_1\ldots v_{r-1}pv_{r+1}\ldots v_k}\Gamma^{p}_{rv_r}.$$

where Γ^a_{bc} are so-called connection coefficients to be made explicit shortly.

In general the metric g and the connection Γ are assumed to be compatible with each other [4, 5, 6].

Definition 8 *The connection (Definition 7) and the metric (Definition 6) are compatible if and only if:*

$$\nabla^\Gamma g = 0.$$

This means that, e.g., the angles between and lengths of vectors measured by the metric tensor g under the parallel transport associated with the affine connection Γ are preserved. The advantage of having no compatibility is that one can detect possible mass-creation in terms of dilation, shearing and rotational curvature currents [4, 5, 6]. These energy creation currents can be derived by setting up a field dependent classical metric-connection on space-time and consider the evolution of the curvatures of a field dependent connection not necessarily compatible with the former metric and compute the energy creation currents by means of equivalences, to be introduced shortly, with time running.

If the metric g is, however, compatible with connection Γ, then the connection coefficients Γ^r_{pq} are normally assumed to be set by the metric (see also (Example 3) for an affine-metric connection that is not torsion-free):

$$\Gamma^r_{pq} = \frac{1}{2}g^{rs}\left(v_p g_{qs} + v_q g_{ps} - v_s g_{pq}\right).$$

At discontinuity, singularity and bifurcation sets of an image (Definition 1) physical observations are path-dependent. Let us clarify this path-dependency by first selecting a two-dimensional surface S, parametrised by $u^k = u^k(d,\delta)$, on the set of energy-states of the detector array, i.e., the jet $j^N(I)$ of the image, and by taking $\tilde{g}$ and $\tilde{h}$ as vector fields generating an infinitesimally small circuit $C = (p_0p_1p_2p_3p_0)$ around p in S. Next let us study the frame field $V = (e_i)$ consisting of a set of independent physical observations e_i such as the image itself or the covariant derivative ∇I of the image. Now let us quantify the variation of frame field V at point p_0 along path $C = (p_0p_1p_2p_3p_0)$ with respect to the local frame V_{p_0} (see figure 1).

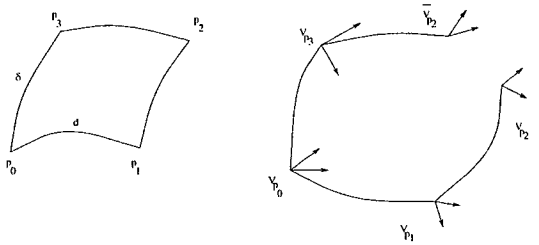


Figure 1: A closed circuit $C = (p_0p_1p_2p_3p_0)$ on S and the frame field V along the upper part and lower part of circuit C.

In general the local frame $\bar{V}_{p_2}$ is not equal to local frame V_{p_2}. The change of the frame field along the upper part $C^+ = (p_0 p_3 p_2)$ of the circuit C generated by $(d\tilde{g})(\delta\tilde{h})$ and its lower part $C^- = (p_0 p_1 p_2)$ generated by $(\delta\tilde{h})(d\tilde{g})$ defines curvatures Φ_i of the physical observations e_i at the point p on S [22, 16, 5]. In order to quantify the formation of the vision system itself and the induced external electromagnetic field activity, in our dynamic scale-space paradigm, curvatures V_i of frame field (Definition 1) are read out.

Definition 9 *The curvature V_i of a frame vector field in (Definition 3) at point p on a two-dimensional surface S parametrised by frame field (Definition 3) is defined by:*

$$V_i(p, S) = \oint_C \nabla^\Gamma v_i,$$

where the sense of traversing circuit C is chosen such that the interior of the circuit C on surface S is to its left.

Note that on S at point p one can still distinguish directions but that Cartan's affine transport is actually directed [4, 5, 6]. The latter directional aspect is in many text-books on differential and integral geometry neglected or obscure [22]; forgetting about it leads naturally to an unwanted averaging of morphisms. Furthermore, that the vision system is in general gauged for his own internal topological, geometric and dynamical intricacies; whether the system is initially in any sense curved or twisted only partly influences the analysis, processing and interpretation of the induced external electromagnetic field activity. Clinging through the induced dynamics suffices to forget or to correct those intrinsicacies.

Using Stokes' theorem curvature (Definition 9) can be expressed as:

$$V_i(p, S) = \int_{C^o \subset S} \nabla^\Gamma \wedge \nabla^\Gamma v_i = \int_{C^o \subset S} O_i^j v_j,$$

where C^o is the interior of the circuit C in S, $\nabla^\Gamma \wedge$ the covariant exterior derivative in which $\wedge$ is the wedge product consistent with metric g and/or connection Γ, and O_i^j represent the curvature two-forms. These curvature two-forms also pop up in the (Cartan) structure equations in case of a differential geometric treatment of the curvature of a connection [6].

In defect theory [15, 16] the above curvatures are known as so-called Burgers and Frank vector density fields for the inhomogeneity of the translation group action and that of the rotation group action, respectively (see Fig. 2).

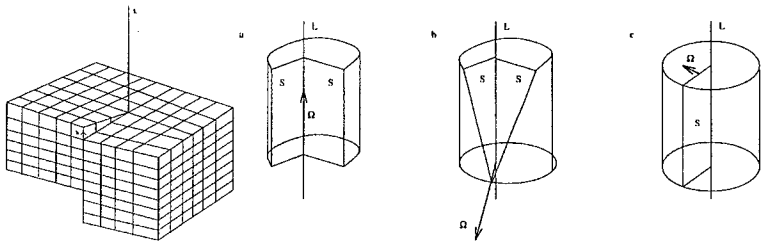


Figure 2: To the left: Burgers vector density field b due to a screw dislocation caused by a displacement field u parallel to defect line L. The underlying Volterra process breaks the cubic lattice symmetry by creating a cut line. Traversing a circuit C around L one returns to the initial cubic lattice to the starting point, whereas in the defected lattice one ends up being Burgers vector b from the starting point. To the right: Frank vector density fields Ω due to a wedge (a), splay (b) and twist (c) disclination. The underlying Volterra process breaks the cubic lattice symmetry by cutting from or opening the initial cubic lattice segments along Volterra surfaces S and applying a specific rotation Ω with respect to defect line L in order to connect the surfaces S or to twist them to each other.

From these curvatures we can in turn derive higher order curvatures $V_{i;j_1\ldots j_k}$ by taking successively covariant derivatives $\nabla_{v_{j_l}}$ with respect to frame vector fields v_{j_l}. Together they form locally and directionally equivalence relations that quantify the formation of the vision system, and that of the induced external electromagnetic field activity.

Equivalence 1 *The local and directional equivalences of the formation of the vision system and those of the induced external electromagnetic field activity are given by:*

$$V_{i;j_1\ldots j_k} = \nabla^{\Gamma}_{v_{j_k}} \cdot \ldots \cdot \nabla^{\Gamma}_{v_{j_1}} V_i,$$

where ; indicates taking a covariant derivative (see also (Definition 8)).

If there are some symmetries, such as those of rigid Euclidean movements, underlying the formation of the vision system and the induced external electromagnetic field activity, then it can be valuable to try to find the irreducible equivalences [22]. (Equivalence 1) allows the quantification of the homogeneity of the formation of the vision system and the above induced activity [4, 5, 6]. They can be used to locate coherent structures, as we'll demonstrate in Section 3, in either the system or its induced activity.

If one considers a set of circuits, $\{C\}$, on a set of related surfaces, $\{S\}$, through point p, then (Equivalence 1) at p satisfies obviously a local conservation law (superposition principle) such that the directional information is obsolescent.

Equivalence 2 *The local equivalences of the formation of the vision system and those of the induced external electromagnetic field activity are given by:*

$$\bar{V}_{i;j_1\ldots j_k}(p,\{S\}) = \sum_{\{S\}} V_{i;j_1\ldots j_k}(p,S),$$

being total curvatures of the vector fields v_i in frame field (Definition 3) over the set of surfaces, $\{S\}$, each of which contains one corresponding circuit C, through point p.

The integral geometric conservation laws (Equivalence 1) and (Equivalence 2) appearing in differential geometry as Bianchi identities (curvature and torsion two-forms form together with the metric, connection and frame field a closed system sufficient and necessary to capture any local equivalence) applies in particular for dislocations and disclinations, i.e., Volterra processes inserting or removing coherent image structures (see Fig. 2), in which one averages Cartan's affine transport over all possible patches S to quantise the formation of the vision system or the induced external electromagnetic field activity at interfaces between regions with different formation aspects. In the case of dislocations and disclinations a superposition principle holds that coincides with the well-known law of Kirchoff for electric currents in a circuit [5].

(Equivalence 2) can be complemented by a global conservation law for a region U on the vision system.

Equivalence 3 *The global equivalences of the formation of the vision system and those of the induced external electromagnetic field activity are given by:*

$$\tilde{V}_{i;j_1\ldots j_k}(\{S\},U) = \int_U \bar{V}_{i;j_1\ldots j_k}(p,\{S\})dU,$$

in which U is a region on N not necessarily of constant dimension nor simply connected to point p.

An important property of a physical object F related to a gauge group G and the induced external electromagnetic field activity N is its invariance under this group.

Definition 10 *A physical object F is invariant under the gauge group G if and only if:*

$$GF = F.$$

All above mentioned geometric objects are by definition unaffected by the gauge group G.

Theorem 1 *The frame field (Definition 3), its differential counterpart (Definition 4), metric (Definition 6), connection (Definition 7), and equivalences (Equivalence 1), (Equivalence 2) and (Equivalence 3) are invariant under gauge group G.*

Example 1: Polarisation It is well-known that the observation of an energy by a detector is not affected by the observation of energy by another detector, whereas the brightness is. The brightness of a white object within a black background and that of the same white object within a grey background are different. Furthermore, the brightness of a white object within a black background and that of the black background surrounding the same white object are as well different.

Several energy to contrast models exist among which a logarithmic one that also appears in Weber's law. This law states that, if the energy I_o of an object is just noticeably different from the energy I_s of its surrounding, then the following ratio ρ should be constant:

$$\rho = \triangle \log I = \frac{|I_s - I_o|}{I_o},\ TI^e \leq \sum I_o \leq TI^E,$$

where T is the total number of detectors in the set of arrays. Let us assume that this Weber law is applicable to our image (Definition 1). The image then consists of a finite set of objects with total energies I_k. Let us consider the objects with energies I_k and I_{k+1}

that touch each other and are simply connected regions on the set of detector arrays (the restriction of adjacency and simply connectedness is not compulsory though).

Now let us define the brightness-function for two objects labelled by k and $k+1$ as follows:

$$B_{kk+1} \quad = \quad [\log I_k + \triangle_{kk+1} \log I],$$

in which

$$\triangle_{kk+1} \log I \quad = \quad \frac{I_k - I_{k+1}}{I_k}$$

is a generalisation of a contrast-function (well-known in almost any field of exact science as a so-called Weber-fraction). One could also conceive each sequence of the contrast function as coupling constants between objects (see also Section 2.2).

Both the brightness- and contrast-function we conceive as two-point functions depending on both the objects' energies and their order $(kk+1)$. The contrast function determines the energy change between the objects relative to one of the energies. Thus one knows exactly how to construct both objects energetically from each other. The energy-change $|I_1 - I_2|$ has only meaning at the interface between the objects and relative to the energies I_1 and I_2 going from the first object to the second and vice versa, respectively. Let us now take for each order $(kk+1)$ the contrast function as frame field v.

Computing a curvature (Definition 9) related to frame field v one experiences at the interface of two objects with energies I_1 and I_2 a jump going from the first to the second object given by (see also Fig. 3):

$$V = \oint_C \nabla_I v = (I_1 - I_2) \left(\frac{1}{I_1} + \frac{1}{I_2} \right) dx \wedge dI \otimes \frac{\partial}{\partial I}.$$

At discontinuity sets such as edges between objects this oriented physically dimensionless curvature, i.e., a polarisation of the induced electromagnetic field activity, is considerably higher than at the interface between objects that differ only say I^e in energy.

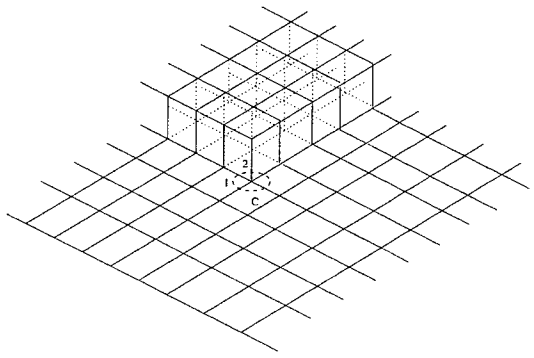


Figure 3: Circuit integral around C reading out polarisation V.

Given $V \neq 0$ at discontinuity sets the question arises what will happen with its value at junctions in a temporal slice of a $(2,1)$-dimensional image. On the basis of

superposition principle (Equivalence 2) this curvature will be just the sum of curvatures over the pairs of objects having a common interface ending at the junction. The latter result is from a geometrical and topological point of view quite disappointing as one rather would like to point out differences between different junctions that have by coincidence equal curvature. The reason for this is, as mentioned above, that the circuit integrals are located by point p within the interior of C of S [22]. Applying Stokes' theorem seemingly sums the curvature information and apparently projects a sequence of geometric and topological quantum numbers onto one number. Therefore, we introduce directed circuit integrals with the initial point p_0 and the endpoint p_2 located, and determining the direction of a vector $p_2 - p_0$. In general we obtain a set of circuits parametrised by a unit vector on a hypersphere Σ that enables to read out the geometric curvatures V in any direction (see also half-space method in [5]). This hypersphere may live on a subset of the tangent spaces to the image. Doing so one can, e.g., determine the valency (a topological curvature) of a junction, i.e., the degree of a vertex on a graph, in a two-dimensional grey-valued image on the basis of the number of jumps occurring in curvature V around the junction. The valency changes in a natural manner over the discontinuity set. At end-points the valency is one, between end-point and junction it is two, at the junction it is just the number of components of the discontinuity set arriving at the junction and elsewhere it is zero.

Example 2: Phases Performing again a directed circuit integral but now with respect to the already found curvatures allows us to measure the coherence, homogeneity or inhomogeneity in the image formation. The curvature or just the covariant derivative (be aware of the direction!) of curvatures allows us to quantify coherency measures for image structures among which textures [5]. For example, a change in polarity between the curvature V at the interface of the objects labelled by k and $k+1$, and the curvature V at that of those labelled by $k+1$ and $k+2$ indicate in relation to the sense of the circuit the presence of non-isolated singularity sets, crest lines of the image gradient field, watersheds, divides and ridges, channels and ruts, and defect lines. Thus the number of changes of polarity (a topological curvature) of the curvature V refines the dynamic quantification of the image formation. Note that the superposition principles (Equivalence 2) and (Equivalence 3) also hold for these refined topological and geometrical quantifications. Furthermore, that at a bifurcation in a spatio-temporal image these quantifications can be achieved again by performing directed circuit integrals of the frame field V^0 but now carried out on an infinitesimally small sphere around the bifurcation. Conceiving the valency of a point as a kind of local topological dimension of the discontinuity set, it is clear that the discontinuity set can be of non-constant dimension. Last but not least, instead of the energies of the objects also other properties such as elliptic moments, orientability and Euler-Poincaré characteristics [5] can be used to quantify optic flow and non-integrable deformations of the image caused by splitting and merging processes [5]. Extending this all to a stereo pair of images one can define a topological-geometric current to match the pair of images [5] (see also Section 2.2).

Let us to conclude consider a two-dimensional grey-valued image I:

$$I(x, y; x_0, y_0) = \Re[\sqrt{z - z_0}\sqrt{z + z_0}], \ z = x + iy, z_0 \in \mathbb{C}.$$

Choosing as frame field v:

$$v = \frac{\nabla I}{|\nabla I|}$$

the curvature V becomes:

$$V = \begin{cases} 2v & \text{if} \quad (x,y) \in Q^o \\ v & \text{if} \quad (x,y) \in \partial Q \\ 0 & \text{if} \quad (x,y) \notin Q \end{cases}$$

where Q is the so-called branch cut of the image gradient v (see figure 4).

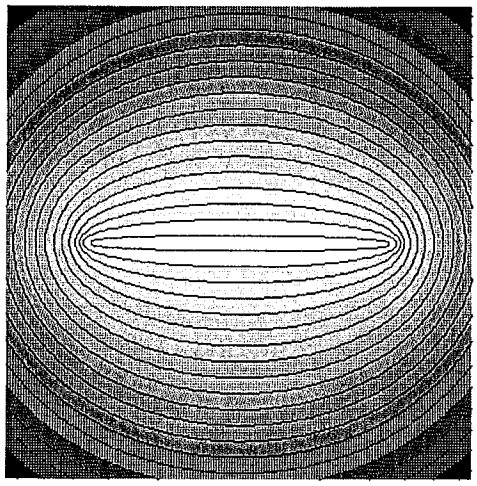


Figure 4: Image I with branch-cut discontinuities on Q for V.

In order to retain global equivalences it is crucial to extract coherent structure before applying the superposition principles mentioned in Section 2.2. Fortunately, the circuit integrals are directed and the circuits in a discrete setting do not need to be taken infinitesimally small. One can assess at a vertex, junction or bifurcation whether a set of topological and/or geometric equivalences along a certain component of their discontinuity or singularity set is satisfied until one reaches a next vertex, junction or bifurcation. Repeating the same procedure at those new physical objects along other components than already examined for a certain property, one can start connecting equivalent components. For example, take a two-dimensional topographic image subjected to a gauge field group (Definition 2). Computing the image gradient field it is clear that the essential physical objects of the transformed image that can also be traced in the original topographic image can be obtained by measuring whether the polarity of the image gradient is consistently changing along paths in the imaging plane. Whether a component of a discontinuity set or singularity set in a region is a ridge or rut can subsequently easily be verified by measuring the lengths and the increment or decrement in the grey-values between begin and end point on each component, and ordering them. Note that this assumes a well-defined ordering of the components of the singularity set [23, 5], and that this ordering is not influenced by the gauge field transformation which in general destroys these ordering relations. The latter destruction of the grey-value ordering caused by a general gauge field transformation shows that physical objects such as crest lines, ridges, ruts and plateaus strongly depend on a certain dynamic ordering (see also Section 3.2.2. The latter dependency does not imply that grey-values are the only physical objects that can allow a topological classification of grey-valued images in terms of crest lines, ridges, ruts, plateaus and alike. Even if true gauge field transformations are allowed, then still it is possible on the basis of the regions partitioned by

the net of crest lines, etcetera, to achieve a topological classification for these regions. For example, the topological or geometric properties of the net or those regions, respectively, can be used to define grey-value like entities not affected by the gauge field transformation of the grey-valued image. Subsequently, again crest lines, ridges, ruts, plateaus and alike can be conceived but now for these more complex physical objects.

Example 3: Torsion and Curvatures Tensors In the sequel we compute the torsion and curvature tensors involved in the formation of a $(2,1)$-dimensional spatio-temporal image I of a density field M. We assume that the gauge group G to consist of the group of Galilean transformations in space-time and the group of spatio-temporally homogeneous and linear density field transformations (where the Galilean group does not affect the density field). The fundamental physical object N that is invariant under G is the set of dynamically ordered isophotes together with the corresponding flowlines. Considering space-time as a regular square grid and temporally equidistant lattice, one can conceive image I as a singular transformation of Galilean space induced by the density field. Furthermore, the group of singular transformations Ξ generated by G can also be viewed as a representation of the fundamental physical object itself.

With respect to a vision system, modelled as a manifold parametrised by coordinates x and endowed with spatio-temporal frame field e, a Galilean metric g (having components $g_{pq} = g(e_p, e_q)$), a flat and non-twisted connection ($\Gamma = 0$) and a linear sensitivity profile for the density field, the fundamental physical object N can be concisely modelled as a manifold parametrised by singular coordinate transformation $\xi = \xi(x)$ of Galilean space and endowed with an affine frame field ϵ, and a metric-affine connection (γ, Ξ). All the above three fundamental physical objects, to be made explicit shortly, are induced by I.

The induced affine frame field ϵ in terms of that of the vision system and the differential structure of I reads:

$$\epsilon_\alpha = e_p \epsilon^p{}_\alpha(x), \epsilon^p{}_\alpha = \frac{\partial x^p}{\partial \xi^\alpha}.$$

Subsequently, the induced metric tensor γ is assumed to have components $\gamma_{\alpha\beta}$ determined by the induced frame field ϵ and the metric tensor g of the vision system:

$$\gamma_{\alpha\beta} = g(\epsilon_\alpha, \epsilon_\beta) = \epsilon^p{}_\alpha \epsilon_{p\beta}, \epsilon_{p\beta} = g_{pq} \epsilon^q{}_\beta.$$

At the non-degenerate points the following orthonormality relations hold:

$$\epsilon_p{}^\alpha \epsilon^p{}_\beta = \delta^\alpha{}_\beta, \epsilon^p{}_\alpha \epsilon_q{}^\alpha = \delta^p{}_q.$$

In order to determine the connection Ξ we write the components of ordinary gradient of a vector field $v = (v_p(x))$ in the Galilean frame field e in terms of the singular frame field ϵ:

$$\frac{\partial v_p}{\partial x^q} = \epsilon_p{}^\alpha \epsilon_q{}^\beta \nabla^\Xi_\beta v_\alpha,$$

with

$$\nabla^\Xi_\beta v_\alpha = \frac{\partial v_\alpha}{\partial \xi^\beta} - \left(\epsilon_r{}^\gamma \frac{\partial \epsilon_\alpha{}^r}{\partial \xi^\beta} \right) v_\gamma = \frac{\partial v_\alpha}{\partial \xi^\beta} - \Gamma_{\beta\alpha}{}^\gamma v_\gamma,$$

where

$$\Gamma_{\beta\alpha}{}^\gamma = \epsilon_p{}^\gamma \frac{\partial \epsilon^p{}_\alpha}{\partial \xi^\beta},$$

are the affine connection coefficients of Ξ.

Now it is readily shown that the covariant derivatives of ϵ_α, ϵ^α and γ vanish locally (even when the reference system is curved and twisted):

$$\nabla_\alpha \epsilon^\beta = 0, \;\; \nabla_\alpha \epsilon_\beta = 0, \;\; \nabla_\alpha \gamma = 0,$$

guaranteeing the compatibility of the connection Ξ with the metric γ.

Studying the symmetry relations for the metric-affine connection (γ, Ξ) by setting up the Cartan structure equations (see (Definition 9)) it appears that the affine connection Γ can be decomposed into a symmetric part $\mathcal{S}$ and a skew-symmetric part $\mathcal{T}$:

$$\Gamma = \mathcal{S} + \mathcal{T},$$

with components

$$\begin{aligned} S_{\alpha\beta}{}^{\gamma} &= \frac{1}{2}\gamma^{\gamma\delta}\left(\frac{\partial \gamma_{\delta\beta}}{\partial \xi^\alpha} + \frac{\partial \gamma_{\alpha\delta}}{\partial \xi^\beta} - \frac{\partial \gamma_{\alpha\beta}}{\partial \xi^\delta}\right), \\ T_{\alpha\beta}{}^{\gamma} &= \frac{1}{2}(\Gamma_{\alpha\beta}{}^{\gamma} - \Gamma_{\beta\alpha}{}^{\gamma}). \end{aligned}$$

Here $\mathcal{T}$ defines the torsion associated with the metric-affine connection (γ, Ξ). If the torsion tensor vanishes identically then the metric-affine manifold N is called a symmetric or Riemannian manifold, else an Einstein-Cartan manifold. Besides establishing the torsion tensor $\mathcal{T}$ the Cartan structure equations also determine the components of the curvature tensor $\mathcal{R}$ associated to the metric-affine connection (γ, Ξ):

$$R_{\alpha\beta\gamma}{}^{\delta} = \epsilon_\beta \Gamma_{\alpha\gamma}{}^{\delta} - \epsilon_\alpha \Gamma_{\beta\gamma}{}^{\delta} + \Gamma_{\alpha\gamma}{}^{\epsilon}\Gamma_{\epsilon\beta}{}^{\delta} - \Gamma_{\beta\gamma}{}^{\epsilon}\Gamma_{\epsilon\alpha}{}^{\delta}.$$

with symmetry relations:

$$R_{\alpha\beta\gamma}{}^{\delta} = -R_{\beta\alpha\gamma}{}^{\delta} = -R_{\alpha\beta\delta}{}^{\gamma}.$$

Note that these symmetry relations are different from a manifold with a general connection Ξ not compatible with a metric γ. Furthermore, that the components $T_{jk}{}^{i}$ are related to the so-called torsion and curvature two-forms, Ω^α and Ω^β_α (see Stokes' version of (Definition 9)):

$$\begin{aligned} \Omega^\alpha &= T_{\beta\gamma}{}^{\alpha}\omega^\beta \wedge \omega^\gamma, \\ \Omega^\beta_\alpha &= R_{\alpha\gamma\delta}{}^{\beta}\omega^\gamma \wedge \omega^\delta. \end{aligned}$$

where ω^α are the connection one-forms associated to Ξ.

Now we are ready to set up the reference frame field e, connection one-forms o^i and o^i_j and those for the physical object N induced by the density field represented by image I.

For the vision system we choose simply as frame field e, Galilean metric g and Galilean connection Γ given by:

$$\begin{aligned} (e_i) &= \left(\frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \frac{\partial}{\partial x^3}\right), \\ g &= dx^i \otimes dx^i, \\ (o^i) &= (dx^1, dx^3, dx^3), \\ (o^i_j) &= 0, \end{aligned}$$

where x^3 is the time coordinate.

For the induced physical object N such a choice is far from trivial, because identifying moving non-rigid spatial objects in Galilean space can only be achieved by inflicting some kind of projection field [5]. Nevertheless, assuming sheer deformations before topological or morphological transformations, approximate velocity or momentum fields can be derived (see also Section 3.2.2). Here, however, we advocate the following induced frame field ϵ given by:

$$(\epsilon_\alpha) = \left(\frac{\partial \log I}{\partial s^2} \frac{\partial}{\partial s^1}, \frac{\partial \log I}{\partial s^2} \frac{\partial}{\partial s^2}, \frac{\partial \log I}{\partial t} \frac{\partial}{\partial t} \right),$$

where s^1 and s^2 are the Euclidean arclength along the isophotes and flowlines in the spatial segments of the image I, respectively. Here the vector fields $\partial/\partial s^\alpha$ with $\alpha, \beta = 1, 2$ are oriented, orthonormal and represented by:

$$\begin{aligned} \frac{\partial}{\partial s^1} &= \frac{\nabla^\Gamma_{\epsilon_\beta} \wedge I}{||\nabla^\Gamma_s I||} \nabla^\Gamma_{\epsilon_\beta}, \\ \frac{\partial}{\partial s^2} &= \frac{\nabla^\Gamma_{\epsilon_\beta} I}{||\nabla^\Gamma_s I||} \nabla^\Gamma_{\epsilon_\beta}, \end{aligned}$$

with

$$||\nabla^\Gamma_s I|| = \sqrt{g(\nabla^\Gamma_{e_\beta} I, \nabla^\Gamma_{e_\beta} I)}.$$

The reason for this choice of induced frame field is that it forms a physical object invariant under the imposed gauge group G (scaling effects of density field are inconsequential for the field analysis and processing).

The connection one-forms ω^α are retained on the basis of the duality constraint, $\omega^\alpha(\epsilon_\beta) = \delta^\alpha_\beta$. The remaining connection one-forms ω^α_β come about by the definition of the affine connection, $\nabla^\Xi \epsilon_\alpha = \omega^\beta_\alpha \epsilon_\beta$. However, the connection coefficients $\Xi^\alpha_{\beta\gamma}$ already suffice to compute the induced torsion and curvature. What is really needed for the derivation of those coefficients are the components of the following matrix E:

$$E = (\epsilon^p{}_\alpha) = \begin{pmatrix} \frac{\partial \log I}{\partial x^2} & -\frac{\partial \log I}{\partial x^1} & 0 \\ \frac{\partial \log I}{\partial x^1} & \frac{\partial \log I}{\partial x^2} & 0 \\ 0 & 0 & \frac{\partial \log I}{\partial x^3} \end{pmatrix}.$$

The connection coefficients $\Xi^\alpha_{\beta\gamma}$ can then readily represented by:

$$\begin{aligned} (\Xi_\beta)^\alpha{}_\gamma &= (\epsilon_\beta \log E)^\alpha{}_\gamma \\ &= (E^{-1} \epsilon_\beta E)^\alpha{}_\gamma. \end{aligned}$$

We observe that, because of the fact that the Euclidean curvatures and the higher order differential structures of the density field represented by I on flowlines deviates from those on isophotes, the induced metric-affine connection is associated a non-zero torsion and curvature with components in terms of the induced reference frame given above. If the the induced spatio-temporal frame field ϵ_3 would also have some kind of momentum field part (retrieved by following coherent structures over time [5]), then there would also appear true spatio-temporal torsions and curvatures.

We also reckon that by the requirement of the duality constraint essential singularities are introduced at those locations where the frame vector fields ϵ_α are branching

[6]. Such singularities are in fact artificial and would not occur if the duality constraint would be replaced. By imposing the connection one-forms ω^{α} to have the same component functions as the frame field ϵ_{α}, the connection and its associated torsion and curvature will never become essentially singular at physical objects. If the gauge group G would include also anamorphoses, then another gauge invariant frame field would be required and the corresponding connection would again become essentially singular. Although this singular behaviour would then not be removable for continuous images it would be unnoticeable for discrete ones. In case we locally assume space-time to be modelled as a Minkovski-space, the above statements still hold.

In order to obtain stable and reproducible physical field representations of the solar irradiance field in Section 3.2 we partition Galilean space-time by considering phases of the induced frame field (see also Example 2). The above found curvatures integrated over cells then become physically sensible representatives above some dynamic scale.

2.2 Exchange Principle for Equivalences

In order to extract from image (Definition 1) a stable and reproducible set of equivalences despite perturbations in this image or defects in the vision system's layout or dynamics there exist a possibility to derive a dynamic scale-space paradigm committed to the connection and metric living on the image of the induced external electromagnetic field activity [4, 5, 6]. This can be achieved by coupling the exchange principles intrinsically to the vision system's geometry and to the induced activity. Essential in these principles are the assessment of the topological interactions activated on the vision system. These interactions can be operationalised by the vision system as a topological current.

For all irreducible equivalences a committed ordering of the the activated vision system can be succinctly formulated through the use of a statistical partition function Z related to free energy F for (Equivalence 1), (Equivalence 2) and (Equivalence 3).

Equivalence 4 *The statistical partition function Z related to free energy F for irreducible (Equivalence 1), (Equivalence 2) and (Equivalence 3) of the formation of the vision system and those of the induced external electromagnetic field activity is defined by:*

$$Z = \sum_{V} \prod_{x,i} \exp\left[-F\left[V_i(x)\right]\right],$$

with

$$F\left[V_i(x)\right] = -\log Z = \sum_{i,k,p} dv^p \left(\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}(x, \tau_{i;\pi_k(g_1 \ldots g_k)})\right),$$

where x labels any state of a detector of the set of arrays giving field V, π_k a permutation of a sequence of $k \geq 0$ integers $(g_1 \ldots g_k)$ with $k = 0$ for labeling frame vector fields v_{g_k} and $\tau_{\alpha;\pi_k(g_1 \ldots g_k)}$ (inner) scale-parameters consistent with the gauge group G and the equivalences $\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}$.

Note that all equivalences are incorporated in the partition function Z by taking all products of equivalences $\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}$ that reside in (Equivalence 1), (Equivalence 2) and (Equivalence 3). Thus one ensures that curvature aspects of Volterra processes are operationalised through an additional weight in free energy F via $\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}$. One

could in this context conceive the partition function and its constituting factors as measures of complexity of the induced system activity. Furthermore, that each component of each density vector field (a finite sequence of topological and geometric numbers) in the local partition function $Z(x)$ forms a factor in the total statistical partition function Z. Last but not least, realise that for each polarisation direction and path one should compute the associated factor in the partition function. Each component of the tensor $\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}$ determines a factor in this function. One could equally well apply a probability argument for yielding configuration V. E.g., in the case that curvature V_i is parallel to v_i has a lower probability than that it is antiparallel (recall that Z and $Z(x)$ can be used to define probability measures for field configurations V).

Because the redistribution of states/currents of a vision system should be independent of the topological, geometric and dynamic intricacies of the vision system at ground state the standard approach is just to operationalise a gauge invariant frame field, metric and connection supported by the vision system but induced by the external electromagnetic field activity. Note that in these parametrisations it becomes for certain gauge groups still indispensable to apply the related Lie group to the frame fields. Doing so and using related integral measures, allows us to retrieve the necessary equivalences. This situation occurs in wavelet theory as well as mathematical morphology [7]. It's clear that this approach has a computational drawback residing in the fact that a overcomplete set of kernels is needed to analyse and process images. Only in a globally Euclidean, Galilean or Lorentzian setting those scale-space paradigms can be of equal computational costs as the induced dynamic ones [4, 5, 6, 7]. E.g., in the full globally affine case a suitable measure for identifying equivalent grey-valued image details is obtained by considering affine invariant structures in the two-jet. An analysis and processing of this primal image data is then carried out most effectively and in a concise manner by applying an affine geometric flow with respect to this data [4, 5, 6].

Besides the induced gauge invariant canonical parametrisation of space-time and dynamics also a topological interaction is needed to ensure an evolution towards a hierarchy of partially equivalent states of the vision system for an ensemble of induced external electromagnetic field activities that are slight perturbations of each other. This topological current is in our dynamic scale-space paradigm [4, 5, 6] brought about by the statistical partition function (Equivalence 4). Studying two local factors $Z(p_1)$ and $Z(p_2)$ in the statistical partition function going from state p_i to state p_j involves a factor $k(i,j)$ to generate $Z(p_j)$ from $Z(p_i)$, whereas going from p_j to p_i requires a factor $K(i,j)$ (assume $k \leq K$) to generate $Z(p_i)$ from $Z(p_j)$ (such that $kK = 1$, i.e., the notable Artin-Whaples formula in disguise). Realising that the interaction can only be defined through the interactions between detector states, it is more than reasonable to let a topological current between states p_i and p_j to be controlled by the partition function Z^r for two-state interactions capturing all the possible couplings between the states of all pairs of detectors:

$$Z^r = \prod_{i \neq j} Z^r_{ij} = \prod_{i \neq j} \left(\frac{K(i,j) + K^{-1}(i,j)}{2} \right) = \prod_{i \neq j} \cosh \left(F(p_i) - F(p_j) \right).$$

Note that these interactions between two states do not exclude long range forcings as $Z(p_i)$ already incorporates (also instantaneously) such field properties.

With this two-state coupling partition function, Z^r, there's associated also a two-state coupling free energy F^r:

$$F^r = -\log Z^r.$$

Assuming the vision system to be a closed system for a particular region of space-time realised on the vision system, then the change in the state of that part of the vision system in order to dynamically scale the acquired image should be governed by a change in the two-state coupling free energy F^r. Keeping in mind that the free energy, see (Equivalence 4), should be preserved, i.e., $dF(p_i) = -dF(p_j)$, this change in the two-state coupling free energy, $\mathrm{d}F^r$, is given by:

$$dF^r = -\sum_{i \neq j} \tanh\left(F(p_i) - F(p_j)\right) dF(p_i).$$

Thus the geometric or topological charges have become the generators of the induced external electromagnetic field activity on the vision system. Now let us consider again the interaction mechanisms between pairs of states p_i and p_j and define the topological current to be the spatio-temporal curl of the induced connection on the two-state coupling free energy:

Definition 11 *The topological current for the free energy on activated vision system is defined by:*

$$j^F = \nabla^\Gamma \wedge dF^r = -\frac{\nabla^\Gamma_{v_s} F}{\cosh^2\left(\sqrt{g(\nabla^\Gamma_{v_s} F, \nabla^\Gamma_{v_s} F)}\right)} dv^s \wedge dF,$$

where v_s is connecting free energy states $F(p_1)$ and $F(p_2)$ of the vision system.

Note that the topological current is steered by (Equivalence 1), (Equivalence 2) and (Equivalence 3) [4, 5, 6]. E.g., the fact that a pixel belongs to a long spatial edge-segment can be used as some kind of stopping criterion or local reflective boundary condition during the dynamic exchange principle stated below.

As the free energy (Equivalence 4) to the vision system should be preserved the dynamic exchange principle for free energy is in our dynamic scale-space paradigm made manifest through a physical law involving topological current (Definition 11):

Law 1 *The dynamic exchange principle for free energy says that the change per unit scale τ in the free energy (Equivalence 4) in a region Ω of the vision system is equal to the exchange of free energy F between this region and its surrounding across their common boundary $S = \partial\Omega$ quantised by topological current (Definition 11):*

$$\delta_\tau F = -j^F,$$

with suitable initial and boundary conditions

For (Equivalence 1), (Equivalence 2) and (Equivalence 3) one can subsequently derive similar laws keeping in mind their tensorial character and effects of the gauge symmetries in taking covariant derivatives of them [4, 5, 6].

3 Applications

We briefly indicate in Section 3.1 how by means of the modern geometric and statistical techniques presented in Section 2 new meso-scale meteorological models can be derived and compared, and the output of existing meso-scale meteorological models can be analysed and processed. In Section 3.2 we apply our dynamic scale-space paradigm to visual, infrared and water vapour Meteosat images in order to find measures for solar irradiation of the earth's surface and cloud cover needed in **GESIMA** [3].

3.1 Models and Model Outputs

In the following paragraphs we recall briefly the physical principles, the subgrid scale and meso-scale decompositions of physical fields and their dynamical equations, the meso-scale assumptions, and parametrisations used in deriving meso-scale meteorological models.

3.1.1 Conservation Laws

Meteorological models are based on various conservation principles concisely represented in terms of a system of integral equations:

$$\int_{\Omega} \frac{\partial \psi}{\partial t} dV dt + \int_{\partial\Omega} \gamma(j_{\psi}, dS) dt = \int_{\Omega} J_{\psi} dV dt,$$

with initial and boundary value conditions:

$$\begin{aligned} \pi(x,0) &= \pi_{obs};\ \pi = \psi, J_{\psi}, A_{\psi}, \\ \int_{\partial\bar{\Omega}} g(j_{\psi}, dS) dt &= -\int_{\partial\bar{\Omega}} g(j_{\psi}^{out}, dS) dt + \int_{\partial\bar{\Omega}} g(A_{\psi}, dS) dt, \end{aligned}$$

where

$$\begin{aligned} \psi &= (\rho, \rho v, \rho\Theta, \rho q_i, \rho\chi_j), \\ j_{\psi} &= \psi v, \\ J_{\psi} &= (J_{\rho}, J_{\rho v}, J_{\rho\Theta}, J_{\rho q_i}, J_{\rho\chi_j}). \end{aligned}$$

Here dV is a differential volume element in Euclidean space, E^3, dt is a differential time element in Euclidean space E^1, Ω is a fixed finite region in space-time, $\partial\Omega$ is its boundary, γ is the ordinary Euclidean metric on space, dS is the unit normal vector field to the spatial part of $\partial\Omega$, $\partial\bar{\Omega}$ is the model domain boundary, and ψ represents the set of conserved physical fields to be predicted and simulated. Among those fields are the mass-density, ρ, of a parcel of air, the linear momentum per unit volume, ρv, the potential temperature times ρ, $\rho\Theta$, water phase mass-fraction times ρ, $\rho q_i = \rho M_{q_i}/M_{air}$, and aerosol phase mass-fraction, $\rho\chi_j = \rho M_{\chi_j}/M_{air}$.

Imposing the following meso-scale assumptions [1]:

- Local thermodynamic equilibrium, i.e., locally radiation is isotropic, and temperature is independent frequency and direction of electromagnetic radiation but dependent on the molecular collisions,
- Avogrado's principle, i.e., gasses at the same pressure and temperature contain the same number of molecules,
- Ideal gas law (Boyle's and Chasles law):

  $$P = \rho RT,\ R = \frac{R^*}{\mu_{atm}},$$

  with

  $$R^* = \frac{P_0 V_0}{T_0} = 8.314310^3 JK^{-1},\ (P_0, T_0, V_0) = (1014mb, 273K, 22.4kl)$$

  where $\alpha = \rho^{-1}$ is the specific volume, (P_0, T_0, V_0) corresponds to 1 $kmol$ of air of μ_{atm} kg mass, and R^* is the universal gas constant,

- Dalton's law of partial pressures:

$$P = \sum P_i$$

and

$$P\alpha = \frac{R^*T}{\mu_{atm}} = \sum q_k P\alpha = \sum \frac{q_k}{\mu_k} R^*T$$

with apparent molecular weight μ_{atm} of air:

$$\mu_{atm} = \frac{\sum q_k}{\sum \frac{q_k}{\mu_k}}.$$

yields the following expression for the potential temperature:

$$\Theta = T_G \left(\frac{P_G}{P}\right)^{\frac{R}{C_p}}$$

in case of isentropy:

$$ds = (C_\alpha + R)\frac{dT}{T} - R\frac{dp}{p} = C_p\frac{dT}{T} - R\frac{dp}{p} = 0,$$

where T_G and P_G are a ground temperature and pressure, C_p is the specific heat at constant pressure and C_α is the specific heat at constant volume (for earth's atmosphere $C_p : C_\alpha \approx 7 : 5$).

The entity j_ψ is the flow of ψ with v the velocity vector field belonging to a parcel of air, and J_ψ is the current of sources and sinks of ψ per unit volume and per unit time.

The momentum p of a parcel of air with respect to the earth's center and a rotating coordinate frame can be quantised as:

$$p = \rho\frac{dR}{d\tau} = j + \rho\omega \times R;\ \ j = \rho\frac{dR}{dt} = \rho v,$$

where R is the position vector with respect to the earth's center. Using the conservation of mass law, the change of momentum with respect to a rotating coordinate frame can be expressed as:

$$\begin{aligned}\frac{dp}{d\tau} &= \frac{dp}{dt} + \omega \times p \\ &= \frac{d\rho}{dt}(v + \omega \times R) + \rho\frac{dv}{dt} + \rho\omega \times v + \rho(\omega \times v + \omega \times \omega \times R) \\ &= \frac{dj}{dt} + \rho(2\omega \times v + \omega \times \omega \times R)\end{aligned}$$

consisting of the force exerted by the rotating earth on the parcel of air, a Coriolis force and centripetal force, respectively. This temporal change in momentum is balanced by external forces such as pressure gradients and gravity, and internal forces caused by fluid self-interactions such as molecular frictional momentum dissipation (neglected). Consequently, the current for the momentum field $J_{\rho v}$ is given by:

$$J_{\rho v} = -(\nabla P + \rho g + 2j \times \omega),$$

with gravity force:

$$g = \omega \times \omega \times R - G, \;\; G = -M_{earth}\frac{R}{|R|^3}.$$

The potential heat current J_Θ is caused by (freezing - melting), (condensation - evaporation), (deposition - sublimation), (exothermic - endothermic) chemical reactions, net radiative flux (convergence - divergence) and dissipation of kinetic energy by molecular motions.

The water phase current J_{q_i} can be decomposed in terms of a water ice phase current J_{q_1} consisting of the sum of (freezing - melting), (deposition - sublimation) and (fallout from above - fallout from below); the rain and cloud water phase current J_{q_2} consisting of the sum of (melting - freezing), (condensation - evaporation) and (fallout from above - fallout from below); and the vapour water phase current J_{q_3} consisting of the sum of (evaporation - condensation) and (sublimation - deposition).

J_{χ_j} consist of the aerosols phase, chemical transformation and reaction, precipitation (fallout due to phase change or chemical reaction) and sedimentation (fallout not due to phase change or chemical reaction) currents.

Note that $\rho_{q_i} = \rho q_i$ and $\rho_{\chi_j} = \rho\chi_j$ are just the mass concentration of the various phases of water and those of other chemical species. Furthermore, that the potential temperature Θ and the mass fractions q_i and χ_j are considered as internal variables of the parcel of air of mass-density ρ. Furthermore, that the momentum field is coupled to the mass density ρ of an air parcel and not to the individual chemical mass concentrations ρ_{q_i} and ρ_{χ_j}.

In the initial-boundary value conditions ψ_{obs} are the observed physical fields, j_ψ^{out} is the outward vector density field not necessarily perpendicular to $\partial\bar{\Omega}$ nor balancing j_ψ, and A_ψ is the total accumulation or dissipation vector density field at $\partial\bar{\Omega}$.

There exist various initialisation procedures of a meso-scale model based on objective analysis, dynamic initialisation and normal mode initialisation [1]. Objective analysis extrapolates the initial dependent physical fields ψ to grid-points using variational analysis to minimise the difference between the observations and analysed fields. With dynamic initialisation the model equations are integrated over time so that the observed fields that are not representative at a meso-scale resolution are minimised. This type of initialisation can be realised by initialisation integration or nudging. In practice one adds to the conserved total physical field ψ a global constraint of the form:

$$\int_\Omega G_\psi(\psi_{obs} - \psi)dV,$$

where G_ψ is the so-called nudging coefficient for field ψ. In the normal mode initialisation high frequency components, assumed to be of no meteorological significance, are filtered by means of horizontal and vertical structure functions being product decompositions of the dependent variables.

The spatial boundary conditions can be divided into [1]:

- Lateral boundary conditions,
- Top boundary conditions,
- Bottom boundary conditions.

The lateral boundary conditions can on the basis whether they are open or closed be subdivided into:

- constant inflow, gradient outflow conditions: ψ remain unchanged at inflow boundaries such that $\nabla\psi = \psi(\Omega^{\odot}) - \psi(\partial\Omega)$
- radiative boundary conditions: minimising the reflection of outward propagating perturbations to the flow, back into the model by means of $\partial\psi/\partial t = -cg(\nabla\psi, dS)$
- sponge boundary conditions: damping advective and wave disturbances in ψ as they move towards $\partial\Omega$ by $\partial\psi/\partial t = -v \cdot \nabla\psi - r(\psi - \psi_0)$, where r a relaxation coefficient (maximal at $\partial\Omega$) and $\psi_0 = \psi(\partial\Omega)$
- periodic boundary conditions

The top of the atmosphere is placed deep within the stratosphere, at the troposphere or within the stable layer of the troposphere. Its shape is assumed to be rigidly or dynamically set by the the variables ψ. Furthermore, the top may be represented by an absorbing layer or a local upper boundary condition.

The bottom boundary conditions generate terrain-induced and synoptically-induced meso-scale systems. In this context the effects of the interactions between ocean and atmosphere and between land and ocean have to be quantised in terms of the observables ψ.

3.1.2 Decomposition of Fields and Dynamics

In order to arrive at a meso-scale meteorological model first of all a decomposition of the physical fields ψ is carried out into a geometric mean $\bar{\psi}$ over a spatio-temporal region Ω of fixed volume and a subgrid scale perturbation ψ':

$$\psi = \bar{\psi} + \psi',$$

with

$$\bar{\psi} = \frac{\int_{\delta\Omega} \psi dV dt}{\int_{\delta\Omega} dV dt}.$$

Subsequently, a similar decomposition with respect to the conservation laws (see Section 3.1.1) is performed on regions $\delta\Omega' = \delta\Omega$.

Next a layer-domain averaging procedure decomposes the geometric mean $\bar{\psi}$ into a synoptic scale mean ψ_0 and a meso-scale deviation ψ'':

$$\bar{\psi} = \psi_0 + \psi'',$$

with

$$\psi_0 = \int_S \bar{\psi} dS.$$

where S is a layer-domain on which the second decomposition is carried out.

3.1.3 Meso-scale Assumptions

After performing scale analysis the following meso-scale assumptions are imposed [1]:

- Reynolds assumption

$$\overline{\bar{\psi}} = \bar{\psi}, \ \overline{\psi'} = 0.$$

- Soundproof assumption; on a synoptic scale the mass conservation law can be rewritten as:

$$\nabla \cdot (\rho_0 v) = 0.$$

- Hydrostatic assumption; on a synoptic scale the vertical equation of motion can be written as:

$$\frac{\partial p_0}{\partial z} = -g\rho_0.$$

- Geostrophic assumption; on a synoptic scale geostrophic wind velocity field can be written as:

$$v^{//} = -\frac{1}{\rho_0 f}\nabla \times p_0,$$

where

$$f = 2|\omega \times e_x|$$

is the Coriolis parameter.

- Boussinesq assumption; on a synoptic scale thermodynamic fields $\tilde{p}$ and $\tilde{\Theta}$ can be split into isentropic reference values and deviations from that:

$$\begin{aligned} \tilde{p} &= p_0 + p'', \\ \tilde{\Theta} &= \Theta_0 + \Theta'', \end{aligned}$$

such that, using the ideal gas law and the expression for the potential temperature, the mass density variation can be approximated by:

$$\frac{\rho''}{\rho_0} \approx \left(1 - \frac{\Theta''}{\Theta_0}\right).$$

3.1.4 Parametrisation of Currents, Turbulent Fluxes and PBL

Incorporating these assumptions in the averaged system of integral equations with initial-boundary value equations for the decomposed physical fields yields a new system in which the currents J_ψ of the physical fields ψ and the turbulent momentum flux tensor, turbulent heat flux vector, turbulent water phase flux vector and the turbulent aerosol phase flux vector still need to be parametrised [1, 3]. In this context it should be remarked that, e.g., in the parametrisation of the turbulent momentum fluxes on the basis of the so-called flux or gradient Richardson number the parametrisations and observations of the heat currents are of eminent importance.

Besides these parametrisations of the currents and turbulent fluxes a planetary boundary layer parametrisation is required. The planetary boundary layer can subdivided in a viscous sublayer, a surface layer and a transition layer. Furthermore, there exist the possibility of internal boundary layers separating layers with different turbulence characteristics. The viscous sublayer lies between z_G and z_0 and is characterised by $\bar{v} = 0$. The surface layer lies between $z_0 = 10\ m$ and $h_s \approx 100\ m$, whereas the transition layer lies between $h_s \approx 100m$ and $z_i > 3\ km$. At height z_i the ground surface no longer influences the dependent variables through the turbulent transfer of mass. The depth of the planetary boundary layer can usually be associated with some kind of inversion:

- Inversions caused by cooling:
  - Radiational cooling at night night, or above stratiform clouds and smog layers
  - Evaporative cooling over moist ground
- Inversions caused by warming:
  - Synoptic subsidence
  - Cumulus induced subsidence
- Inversions caused by advection
  - Frontal inversions
  - Warm air over cold land, water, or snow
  - Vertical differences in the horizontal advection of temperature

Other parametrisations of the boundary layer can be based on different sets of balances of forces [1]. Furthermore, the layers are characterised each by their typical turbulent flux representations which depend on the chosen closure forms [1].

3.1.5 Output, Sensitivity and Setup of Systems

Every meso-scale meteorological system's output in terms of the physical fields ψ determines a partitioning of space-time into regions characterised by specific dynamical states of the atmospheres. As we show also in Section 3.2 the fields, i.e., the currents, themselves induce a particular canonical parametrisation of space-time with which are related a specific frame field, connection and/or metric and consequently a set of equivalence relations as presented in Section 2.1. For example, the mass-density ρ of the air parcels over space-time determines a net of isophotes, i.e., surfaces of iso-densities, and flowlines, i.e., the integral curves of the mass-density gradient field, with a characteristic segmentation in spatio-temporal fore- and background dynamics.

An ensemble of (similar) meso-scale meteorological systems will show various outputs because of different model assumptions, closure forms, current and turbulent flux parametrisations and errors in the initialisation phase of the observed physical fields among which the parametrised currents and initial physical fields on the model domain [1]. Again our dynamic scale-space paradigm yields a mean to unravel the dynamical structures of and differences in the evolutions of the observed physical fields under various systems. Performing a multi-scale analysis and processing of the outcomes of these systems analogous Section 2.2 we can derive a set of stable and reproducible equivalence relations despite various types of noise, e.g., related to observation systems, for each system, and compare these sets with each other. Besides on the level of the outputs of the systems such a comparison can equally well be carried out on the level of the models by subjecting the symmetries involved in the various models, having different complexities, parametrisations and assumptions, to a modern geometric or Lie theoretic examination. A major impact of such an analysis will be that a sensitivity analysis yields then not only local measures of influences of e.g. changing parametrisations but also non-local and topological-geometrical measures. Whether there will apppear more than one heat island due to changing human activities in an area, for instance, will then be retrievable from the model and the initialisation of the system. A refined and immediate scenario analysis thus becomes feasible and into sight through

a modern geometric analysis and processing. Last but not least, (not time-causal and recurrence) effects of adjoint methods [2], during the initialisation phase of the system, on the evolution of the physical fields can qualitatively and quantitatively determined by our presented methods.

More importantly, novel models can be found by adapting the models to the observed physical field dynamics. The adaptations concern the various decompositions and averaging procedures of physical fields and dynamics, meso-scale assumptions, parametrisations of currents, turbulent fluxes and planetary boundary layer. Let us point out more precisely which strategy to follow to couple the model, system and observed dynamics to each other.

On the basis of the observed physical field dynamics a decomposition of physical fields and dynamics is possible in which the subgrid scale and synoptic scale averaging procedures couple to the dynamics of those fields (see Section 3.1.2). This implies that the regions $\delta\Omega$ become spatio-temporally and field dependent. The latter can be brought about by a Galilean frame field induced by the physical fields or derived fields such as the Burgers and Frank vector density fields mentioned in Section 2.1. A subgrid and synoptic scale subsequently have a pure field or dynamics based substantiation. These field dependent scales for averaging and decomposition then allow a validation of meso-scale assumptions over space-time (see section 3.1.3). For example, we might be bound to abandon Reynolds assumption in certain regions of space-time where there are sharp transitions in the fields and the dynamics. The partitioning of space-time by the physical fields or currents (see also Section 3.2) then also permits a better parametrisation of currents, turbulent fluxes and PBL (see Section 3.1.4). E.g., turbulent flux exchange coefficients dictated by the observed physical field dynamics will turn out to be spatio-temporally inhomogeneous, a-symmetric and anisotropic. Furthermore, the regimes (regions in space-time and in field dynamics) that are normally determined on the basis of so-called Richardson numbers can be given more solid definitions in terms of the equivalences and scaling algorithm presented in Section 2.1 and Section 2.2, respectively.

3.2 Solar Irradiation Input Data

The heat current $\bar{J}_\Theta$ is dependent on the temporal variation of the grid-volume averaged absolute temperature or the grid-volume averaged absorbed irradiance due to all wavelengths of solar electromagnetic radiation, $\bar{J}_{\Theta,a}$, on the heating and cooling, $\bar{J}_{\Theta,(q,\chi)}$, due to phase changes in water and aerosols, and on the dissipation of kinetic energy by molecular friction, $\bar{J}_{\Theta,dis}$:

$$\bar{J}_\Theta = \bar{J}_{\Theta,a} + \bar{J}_{\Theta,(q,\chi)} + \bar{J}_{\Theta,dis},$$

with

$$\bar{J}_{\Theta,(q,\chi)} = \bar{J}_{\Theta,q} + \bar{J}_{\Theta,\chi},$$

where

$$\bar{J}_{\Theta,q} = \rho Q_{\Theta,q}, \quad \bar{J}_{\Theta,\chi} = \rho Q_{\Theta,\chi}.$$

As the **Meteosat** satellite images mainly deliver information related to $\bar{J}_{\Theta,a}$ we concentrate on the fundamentals in the parametrisation of this current (see Section 3.2.1) corresponding to the **GESIMA** model [3]. In Sections 3.2.2 and 3.2.3 we analyse and process the solar irradiation currents in the short-wave and long-wave band.

3.2.1 Solar Irradiance Currents

The absorbed radiant energy from the sun is equal to the opposite of the change in total heat of an atmospheric layer. In the sequel we treat the contribution of the long-wave and short-wave band fluxes relevant to the **GESIMA** system [3]. For a treatment of the parametrisations of those fluxes in case of clear, clouded and polluted atmosphere we refer to [1].

The monochromatic intensity (radiance) flux vector of electromagnetic radiation through dS per time per wavelength is given by:

$$I_\lambda(\zeta,\xi) = \frac{de_\lambda}{d\Omega} e_S, \;\; d\Omega = dS d\lambda dt$$

with de_λ the differential amount of radiant energy passing through differential surface area dS in time interval dt and per increment $d\lambda$ of wavelength. The differential area on the surface of the earth can be written as:

$$dS = dA \cdot d\Psi, \;\; d\Psi = \sin\zeta d\zeta d\xi e_\Psi,$$

with dA at the center of the earth a unit surface normal vector to the plane through the equator and $d\Psi$ a differential solid angle in direction direction e_Ψ.

Integrating over the hemisphere yields the monochromatic irradiance flux vector on dA:

$$R_\lambda = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} (I_\lambda(\zeta,\xi) \cdot d\Psi) e_A$$

which for isotropic radiation boils down to:

$$R_\lambda = \pi |I_\lambda| e_A.$$

The total isotropic irradiance is obtained by integrating over all wavelengths λ:

$$R = \pi \int_0^\infty |I_\lambda| e_A d\lambda.$$

A blackbody absorbs all the radiation fallen upon it; in this context the blackbody irradiance that directly relates to the Stefan-Boltzman law is of fundamental importance. Let N_{ϵ_0} be the number of oscillators in the ground energy state ϵ_0 and the number $N_{e_0+n\epsilon}$ of oscillators at an energy state $n\epsilon$ higher than the ground state governed by Boltzman statistics:

$$N_{e_0+n\epsilon} = N_{\epsilon_0} \exp\left(\frac{-n\epsilon}{k_B T}\right), \;\; \epsilon = h\nu.$$

Now the total number N_{tot} of oscillators in energy state ϵ is given by:

$$N_{tot} = \sum_{n=0}^{\infty} N_{e_0+n\epsilon} = \frac{N_0}{1 - \exp\left(\frac{-\epsilon}{k_B T}\right)},$$

and the total energy E_{tot} over all energy states by:

$$E_{tot} = \sum_{n=0}^{\infty} n\epsilon N_{e_0+n\epsilon} = \frac{N_0 \epsilon}{\left(1 - \exp\left(\frac{-\epsilon}{k_B T}\right)\right)^2},$$

such that the mean energy $< E >$ becomes:

$$< E >= \frac{E_{tot}}{N_{tot}}.$$

Planck's second postulate assumes that the phase changes of the oscillators are quantised. The monochromatic energy density u_ν, i.e. the energy per unit volume per unit frequency interval in a cavity with temperature T is given by:

$$u_\nu = A < E >,$$

where on the basis of the equi-partition principle, that is if $T \to \infty$ and $\nu \downarrow 0$:

$$u_\nu = \frac{8\pi\nu^2}{c^3} k_B T$$

follows from $u_\nu / k_B T \to 0$ that:

$$A = \frac{8\pi\nu^2}{c^3}.$$

The blackbody irradiance was assumed isotropic such that:

$$B_\nu(T) = \frac{u_\nu c}{4\pi}$$

which reads in wavelengths λ:

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5 \left(\exp\left(\frac{hc}{k_b \lambda T} \right) - 1 \right)}$$

and leads to a total emitted isotropic blackbody irradiance:

$$R^* = \int_0^\infty B_\lambda(T) d\lambda = \sigma T^4, \;\; \sigma = \frac{2\pi^4 k_B^4}{15 c^2 h^3}.$$

The blackbody irradiance spectrum can be effectively divided into a short-wave radiation from the sun and long-wave radiation from the earth. Furthermore, the monochromatic blackbody irradiance can be used to define so-called monochromatic absorptivity, a_λ, reflectivity, r_λ, and transmission function, t_λ:

$$g_\lambda = \frac{I_\lambda^g}{B_\lambda}, \;\; g = a, r, t,$$

with

$$\sum_g g_\lambda = 1.$$

Here the reflectivity can be further decomposed into multiple reflections (scattering) in direction of source field r_λ^s and those in directions other than that of the source field r_λ^o. The sum of $a_\lambda + r_\lambda^o$ is called the extinction or attenuation. In local thermodynamic equilibrium the monochromatic absorptivity is equal to the monochromatic emissivity ϵ_λ (Kirchoff's law):

$$\epsilon_\lambda = a_\lambda.$$

Short-Wave Radiation The solar flux at the surface is retained from the position of the sun and from either amount of cloud cover or from liquid water path. The solar zenith angle ζ_{zen} between a horizontal surface element and the solar energy flux vector is defined by:

$$\zeta_{zen} = \sin\Psi \sin\delta + \cos\Psi \cos\delta \cos t_H,$$

with Ψ is the geographical latitude, and δ the declination of the sun which is a function of the day N_d in the year:

$$\sin\delta = \sin\epsilon \sin\left(\frac{2\pi N_D}{365}\right),$$

where is the ecliptic earth angle and N_D the days after the start of spring. The local hour angle t_H is defined $0^{\odot}$ at noon and $180^{\odot}$ at nidnight and is related to local time t_L by:

$$t_H = \frac{(t_L - 12)\pi}{12}.$$

The solar radiation flux S_0 reaching the surface can be written as [3]:

$$S_0 = (\bar{S}_0\mu - \bar{S}_1)T_s, \;\; \mu = \cos\zeta_{zen},$$

with short-wave transmission function T_s in case there's no explicit cloud parametrisation, and where $\bar{S}_0$ and $\bar{S}_1$ are numerical constants.

Long-Wave Radiation Flux In the infrared spectrum the measured radiance relates to the temperature of the emitting atmospheric column. In the water vapour spectrum it's possible to relate the long-wave transmission function T_l the long-wave emissivity ϵ_l as follows:

$$\epsilon_l = 1 - T_l.$$

This long-wave emissivity then supplies information about the precipitable water $\delta P = \int \rho q_3 dz$ between the satellite and either the earth's surface or the cloud tops [1].

3.2.2 Analysis of Solar Irradiation Currents

We partition a time-sequence of two-dimensional infrared images by extracting spatial and temporal edges, and spatial ridges, ruts and inflection lines. We choose, thereto, the following frame field ϵ and one-forms of connection Ξ:

$$\begin{aligned}
(\epsilon_i, \epsilon_t) &= (\nabla_s^{\Xi} \wedge R, \nabla_s^{\Xi} R, \nabla_t^{\Xi} R), \\
(\omega^{\alpha}) &= (ds^i, dt), \\
\gamma &= \omega^{\alpha} \otimes \omega^{\alpha}, \\
(\omega_j^i, \omega_t^t) &= \left(\begin{pmatrix} 0 & -\kappa_i \\ \kappa_i & 0 \end{pmatrix} ds^i, 0dt \right), \\
\omega_j^t \omega_t^i &= 0
\end{aligned}$$

where I image (Definition 1), i, j denote the isophotes and flowlines, ds and κ refers to the Euclidean differential arclength and curvature field on those curves, and t denotes the time variable. Note that space-time is locally assumed, for simplicity, to be

Galilean. Instead we could also adapt the local geometry to the perspective transformation, induced by the vision system, of the atmosphere's boundary layer surface geometry. Doing so one can also associate to each visual ray and perceived pixel radiance a surface area of the boundary layer (see [24] and Appendix A).

Studying the changes of the frame vector field ϵ_2 in the direction of this field itself, one can nicely locate the spatial edges by means of the signature of $\nabla^{\Xi}_{\epsilon_2}\epsilon_2$ projected onto ϵ_2; a similar signature can be given for the temporal edges (see also Fig. 5):

$$\begin{aligned} \sigma^{s}_{edge} &= sign(\gamma(\nabla^{\Xi}_{\epsilon_2}\epsilon_2, \epsilon_2)), \\ \sigma^{t}_{edge} &= sign(\nabla^{\Xi}_{\epsilon_t}\epsilon_t, \epsilon_t)), \end{aligned}$$

where $\cdot$ denotes the standard inner product.

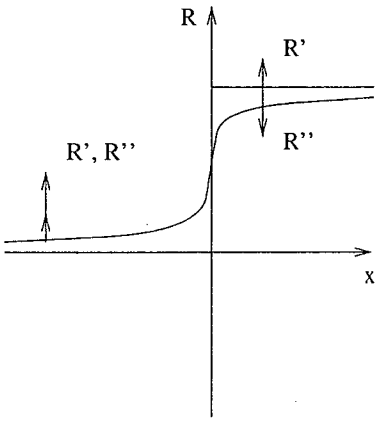


Figure 5: Inflections of solar irradiance R; in the direction of the x-axis R' and R'' are parallel and subsequently anti-parallel.

For the spatial slices of the image sequence in the neighbourhood of ridges, ruts and inflection lines the flowline curvature vector field $\vec{\kappa}_2$ reverses orientation. Following the ridge, rut or inflection line and continuing on the tangent inscribing circle, one traverses this circle clock- or counter clockwise (right- or left-handedness). On the basis of the sign of the isophote curvature one can subsequently distinguish between ridges and ruts: $\kappa_1 > 0$ for a ridge and $\kappa_1 < 0$ for a rut. In order to retrieve all these special curves at once we have voted instead for the signature of $\nabla^{\Xi}_{\epsilon_2}\epsilon_2$ projected onto ϵ_1:

$$\sigma_{phase} = sign(\gamma(\nabla^{\Xi}_{\epsilon_2}\epsilon_2, \epsilon_1)).$$

Note that on either side of the ridges, ruts and inflection lines the frame vector field e_1 is the same but the flowline curvature vector reverses direction across them. In Fig. 7, Fig. 8 and Fig. 9 edges, and ridges, ruts and inflection lines in a time-sequence of images in the visual, infrared and water vapour spectrum, respectively, occur there where σ^{s}_{e}, σ^{t}_{e} and σ_{phase}, respectively, change sign. The sought curves are retrieved by simple binary operations; one obtains a dynamic cellularisation of the spatio-temporal infrared image in fore- and background dynamics. For the Ansi-C code we refer the reader to Appendix B.1 and Appendix B.2. The documentation of the included library `<inrimage>` can be recovered at:

$$http://www-syntim.inria.fr/chieze/publichtml/inrimage/.$$

The ridges and ruts distinguish themselves from the inflection lines that the polarity at either side of the curves changes phase, i.e., $V_1 \cdot V_2 < 0$, whereas at inflection lines this phase remains the same, i.e., $V_1 \cdot V_2 > 0$. Thus a simple algorithm to find the ridges and ruts consists of summing all gradient information within a cell and compare it with neighbouring cells. Computing the phases between (neighbouring) cells gives then a clue whether their interfaces are ridges and ruts, or inflection lines.

However, it should be reckoned that it may happen that ridges or ruts end. But then one could indeed object that in the case of u-type of junctions (see also Fig. 6) the branch-cut or ridge/rut cannot be detected in the above manner causing in addition problems in the assessment of the enveloping interfaces of such a region.

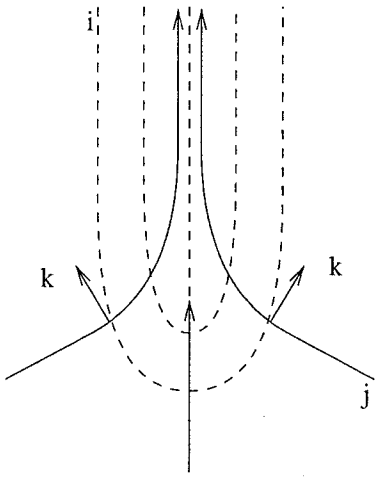


Figure 6: Ridge in case of u-junction where i are isophotes and j are flowlines (solid curves) and k are flowline curvature vector fields.

In that case the only machine available seems to be the one measuring the curvature of the frame vector field using a special frame field $\bar{\epsilon}_\alpha$ and connection $\bar{\Xi}$:

$$\begin{aligned}\Phi &= \oint_C \nabla^{\bar{\Xi}} \epsilon_2 \\ &= \int_S ds^1 \wedge \frac{\partial}{\partial s^1}\left(\frac{1}{|\kappa_2|}\frac{\partial \epsilon_2}{\partial s^2} ds^2\right),\end{aligned}$$

with

$$\begin{aligned}(\bar{\epsilon}_1, \bar{\epsilon}_2) &= (\epsilon_1, \left(\frac{1}{|\kappa_2|}\right)\epsilon_1), \\ (\bar{\omega}^1, \bar{\omega}^2) &= (ds^1, ds^2), \\ \bar{\omega}_i^j &= 0.\end{aligned}$$

At ridges and ruts this curvature is nonvanishing as the normalised flowline curvature vector field $\hat{k}$ changes polarity. At inflection lines this measure is normally vanishing as an isophote need not be coinciding with an inflection line. However, realising that near an inflection line σ^s_{edge} is changing sign, whereas at ridges and ruts it preserves sign we can better study this field to locate the inflection lines first. Subsequently, by simple set-theoretic considerations the ridges and ruts can be retained.

Note that the recovered inflection lines can be ordered going from a rut up to a ridge; there exist apparently tilted and slanted plateaus. Furthermore, walking along different ridges and ruts one can also induce a meaningful ordering on these dual physical objects.

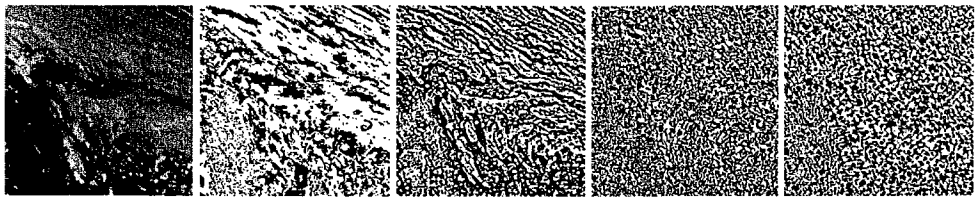


Figure 7: From left to right: spatial slice of an image in the visual spectrum at noon, $t = 12\ h$, its temporal and spatial edges coinciding with the interfaces and regions where σ^t_{edge} and σ^s_{edge} change or have a particular sign, the ridges, ruts and inflection lines coinciding with the interfaces where σ_{phase} changes sign, and the ridges occur where the signature of k_i is positive. The input image acquired between $t = 9\ h$ and $t = 18\ h$ consists of 18 temporal views of spatial slices composed of 200×200 voxels each with a dynamic resolution of 8 $bits$.

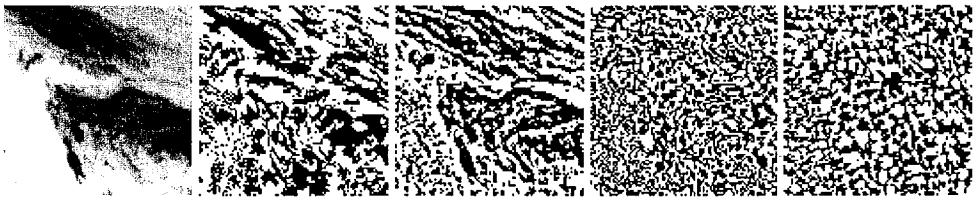


Figure 8: From left to right: spatial slice of an infrared image at noon, $t = 12\ h$, its temporal and spatial edges coinciding with the interfaces and regions where σ^t_{edge} and σ^s_{edge} change or have a particular sign, the ridges, ruts and inflection lines coinciding with the interfaces where σ_{phase} changes sign, and the ridges occur where the signature of k_i is positive. The input image acquired between $t = 9\ h$ and $t = 18\ h$ consists of 18 temporal views of spatial slices composed of 200×200 voxels each with a dynamic resolution of 8 $bits$.



Figure 9: From left to right: spatial slice of a water vapour image at noon, $t = 12\ h$, its temporal and spatial edges coinciding with the interfaces and regions where σ^t_{edge} and σ^s_{edge} change or have a particular sign, the ridges, ruts and inflection lines coinciding with the interfaces where σ_{phase} changes sign, and the ridges occur where the signature of k_i is positive. The input image acquired between $t = 9\ h$ and $t = 18\ h$ consists of 18 temporal views of spatial slices composed of 200×200 voxels each with a dynamic resolution of 8 $bits$.

3.2.3 Processing of Solar Irradiation Currents

We give an illustration of our dynamic scale-space paradigm for the images considered in (Example 1). The reasons for such a filtering scheme is to retain stable and reproducible input data concerning the short-wave and long-wave transmission function of the solar irradiation. The generating integral equations with reflective boundary conditions for our paradigm is given by:

$$\delta_\tau R = \sum_\chi \frac{\nabla^\Gamma_\chi R}{\cosh^2\left(\sqrt{g(\nabla^\Gamma_\chi R, \nabla^\Gamma_\chi R)}\right)} d\chi \wedge dR,$$

with χ a frame vector field in any direction. We have shown in Fig. 10, Fig. 11 and Fig. 12 not only the infrared image in Fig. 7, Fig. 8 and Fig. 9, respectively, at different scales, but as well the segmentation in fore- and background dynamics on the basis of the signatures, mentioned in (Example 1), at those scales. As to be expected the dynamic scale-space paradigm preserves larger scale edges and alike longer over scale. There's, of course, a boundary condition interfering with the analysis at the image domain boundary that is possibly not supported by the input data outside the image domain. It's, therefore, in case studies concerning the atmospheric dynamics better not to include these boundary analyses (unless one has earth covering input data). Nevertheless, the partitioning by edges and alike produce a natural and direct multi-scale segmentation in terms of micro-, meso- and synoptic scale physical regimes for the atmospheric dynamics [1]. For the Ansi-C code we refer the reader to Appendix B.3.

A segmentation of the dynamic scale-space of the various sequences of input images comes about by using simply the following signature σ_τ:

$$\sigma^\tau_{edge} = sign(d\tau(\nabla^\Gamma_{\epsilon_\tau}\epsilon_\tau)d\tau(\epsilon_\tau)),$$

with

$$\epsilon_\tau = \frac{\frac{\partial R}{\partial \tau}}{\sqrt{\frac{\partial R}{\partial \tau}\frac{\partial R}{\partial \tau}}} \frac{\partial}{\partial \tau}.$$

Note that similar remarks can be made as in (Example 1) concerning the influence of the geometry involved in the data acquisition. Furthermore, that it might be more

appropriate to retrieve first currents for the Volterra processes involved in the atmospheric radiation dynamics, and to use them to steer the dynamic scaling [4, 5, 6]. All the images can supply us with particular mean heat momentum fields by following the segmented regions over space-time. The processed image sequence in the visual spectrum yields us directly a measure for the short-wave transmission function. Using the geocoding used by **Meteosat** (see Appendix A) and that used in **GESIMA**, [3], we can supply during the day a related meso-scale meteorological system with that function. On the basis of the correlation between the three types of image sequences we hope to retain also logical/real values for the cloud coverage or short-wave transmission function during model run at night and early morning the next day. Furthermore, we can use the cloud cover measure during the night to represent the long-wave transmission function during the night. Last but not least, that in case stable and reproducible data are needed during model run it's more plausible to restrict the filtering on a cone in space-time pointing in the past and normalise the filter output in a unique manner (the latter normalisation procedure is not obligatory and can even be inconsistent with observed dynamics). In this manner the observed history of the radiance fields in the different spectra determines an initialisation of the transmission functions long- and short-wave band. If there would be a cloud module and supporting satellite observations available, then we could also start off forecasting and simulating the transmission functions system-wise.

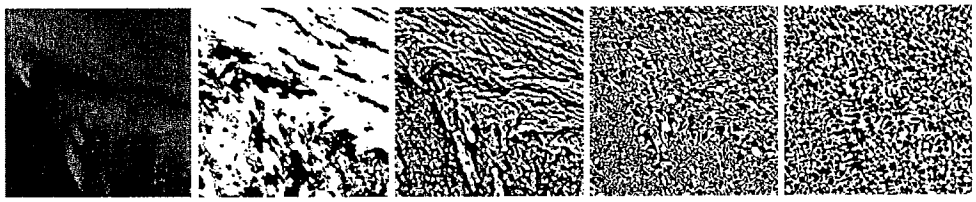


Figure 10: From left to right and top down: similar sequence of images as in Fig. 7 at scale $\tau = 1$.

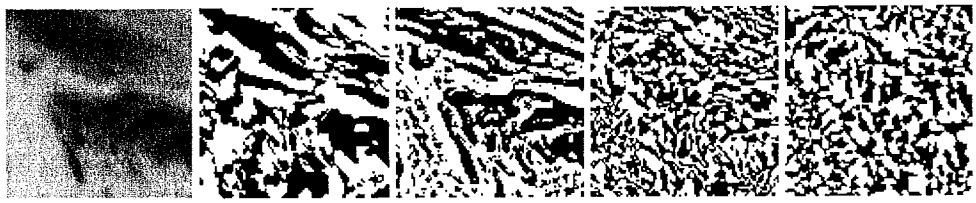


Figure 11: From left to right and top down: similar sequence of images as in Fig. 8 at scale $\tau = 1$.

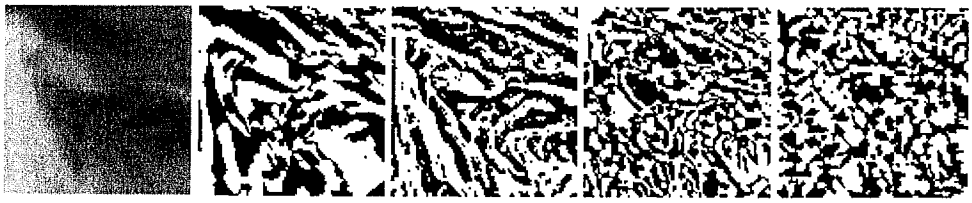


Figure 12: From left to right and top down: similar sequence of images as in Fig. 9 at scale $\tau = 1$.

4 Conclusion and Discussion

We have presented the concepts involved in our dynamic scale-space paradigm. After identifying the relevant gauge group in a vision task we are able to formulate a dynamic scale-space paradigm. The gauge group supports subsequently a frame field consistent with the vision system's topology, geometry and dynamics and with those of the induced external electromagnetic field activity. Furthermore, the gauge group allows a construction of a suitable metric and/or connection, such that equivalences for the field activity can be recovered. We showed how a simple geometric analysis of a two-dimensional grey-valued image can supply us with its edges, ridges, ruts and inflection lines; the interfaces between physically essentially different image formation processes invariant under (volume-preserving) diffeomorphisms of the image. The equivalences, mainly retrieved by performing directed circuit or path integrals over the frame field along physical objects, such as interfaces between different media, allow in turn the construction of a partition function that forms the basis for the derivation of (dynamic) exchange principles for the equivalences themselves. These principles can be succinctly quantified as a controlled distribution by a topological current of equivalences. The partition function can be conceived as a measure of the topological, geometric and dynamical complexity of the external electromagnetic field activity. The advantage of our measure(s) is, as we will become clear shortly, that it readily substantiates and extends information theoretic measures as proposed in [25]. Furthermore, the new measures of complexity are to be preferred for their conciseness, i.e., completeness and irreducibility. Moreover, the dynamic scale-space paradigm falls nicely within the realm of modern theory of dynamical systems [26].

Of course, our approach raises also a lot of hardly ever addressed research issues. The frame field, metric and connection and curvatures presented in this paper determine equivalences of the observed external electromagnetic field activity. They were acquired by performing just circuit integrals on a two-dimensional surface and integrating them subsequently over other physical objects. Characteristic for these equivalences and thus also for related paradigm is that they are all based on purely local interaction mechanisms, i.e., the Green's functions have all a simply connected support. The paradigm excludes the operationalisation of really nonlocal, collective (cooperative or antigonistic) interactions between subprocesses in the induced external electromagnetic field activity. It should be possible in the line of, e.g., Biot-Savart's law for the force between two conducting currents to formulate attractive and and repulsive currents of related equivalences. We already reckoned in Ref. [27] that among these equivalences are those of the self-linking numbers, (generalised) Vassiliev invariants for knots and higher dimensional manifolds, invariants related to links and braids, and Möbius energies [28, 29, 30, 31, 32, 33]. These invariants and energies are true generalisations

of the presented equivalences, because they can capture the (self)-interactions of physical processes in the electromagnetic field. The chirality of a medium, i.e., the left- or right-handedness of the microstructure of a medium manifesting itself as a nonlocal phenomenon via the dependence of the polarisation and magnetisation of the medium on the circulation of the electric and magnetic field, respectively, supports the idea that mechanisms grouping electromagnetic activities by observing linking aspects in the external fields might also belong to the standard operations of a vision system. Although the mentioned invariants and energies possess a very high computational complexity they may play a major role as constraints or granulometric characteristics in filtering schemes restoring, enhancing and/or simplifying coherent dynamical processes. E.g, they may be used as actions to derive equations of motion upon varying canonical coordinates, metric and connection. After solving these equations structures living on those topologically or geometrically invariant objects can be diffused using the Beltrami-Laplace operator consistent with the found metric and connection. These topological invariants and energies come into play in tracing the transitions in dynamical processes. Furthermore, the above new equivalences may form new factors in statistical partition functions, information or topological entropy measures as suggested in the beginning of our discussion, that engender a hierarchy of dynamical systems that can again be explored by means of our topological, geometric or algebraic techniques.

Besides having presented our dynamic scale-space paradigm for analysing and processing physical observations we also pointed out how to apply it to the output of meteorological systems, how to compare different systems by mean of it, and how to derive, given the observations, induced meso-scale meteorological models. For instance, sensitivity and scenario analysis can be enhanced and extended supplying decision makers with objective measures concerning the impacts of e.g. desertification on esosion. In order to demonstrate the applicability of our method to the analysis and processing of physical fields, we segmented **Meteosat** temporal image sequences in the visual, infrared and water vapour spectrum into fore- and background dynamics at various scales. This multi-resolution filtering and segmentation scheme has been considered in order to retain stable and reproducible input data concerning the short-wave and long-wave transmission function of the solar irradiation (which plays a very important role in several modules in meso-scale meteorological models such as **GESIMA**, [3]). Integration and evaluation of our method in such systems, by keeping in mind the geocoding used in the satllite images and the models, has still to be accomplished.

Acknowledgments

This work was supported by the European Research Consortium on Informatics and Mathematics financed by the European Commission. The author would like to express his sincere gratitude to Dr. R. C. Venkatesan, Systems Research Corporation, Pune, India, for the discussions on the relation between chirality of media and the modern geometric notion of curvature.

References

[1] R. A. Pielke. *Mesocale Meteorological Modeling*. Academic Press, 1984.

[2] H. Kapitza. Numerical experimenst with the adjoint of a non-hydrostatic mesoscale model. *Monthly Wea. Rev.*, 119:2993–3011, 1991.

[3] H. Kapitza and D. P. Eppel. The non-hydrostatic mesoscale model GESIMA. Part I: Dynamical equations and test. *Beitr. Phys. Atmosph.*, 65(2):129–146, 1992.

[4] A. H. Salden, B. M. ter Haar Romeny, and M. A. Viergever. Modern geometry and dynamic scale-space theories. In *Proc. Conf. on Differential Geometry and Computer Vision: From Pure over Applicable to Applied Differential Geometry*, Nordfjordeid, Norway, August 1–7 1995.

[5] A. H. Salden. *Dynamic Scale-Space Paradigms*. PhD thesis, Utrecht University, The Netherlands, November 1996.

[6] A. H. Salden, B. M. ter Haar Romeny, and M. A. Viergever. Differential and integral geometry of linear scale-spaces. *Journal of Mathematical Imaging and Vision*, 9(1):5–27, 1998.

[7] B. M. ter Haar Romeny, editor. *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1994.

[8] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge MA, 1986.

[9] J. Aggarwall and N. Nandhakumar. On the computation of motion from sequences of images. *IEEE Tr. PAMI*, 76(8):917–935, 1988. A review.

[10] O. Gaugeras and S. Maybank. Motion from point matches: Multiplicity of solutions. 4:225–246, 1990.

[11] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In G. Sandini, editor, *Proc. ECCV'92*, pages 187–202, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[12] L. M. J. Florack and M. Nielsen. The intrinsic structure of the optic flow field. Technical Report ERCIM-07/94-R033 or INRIA-RR-2350, ERCIM, July 1994. http://www-ercim.inria.fr/publication/technical_reports.

[13] J. Aggarwall and N. Nandhakumar. Correspondence processes in dynamic scene analysis. *IEEE Tr. PAMI*, 69(6):562–572, 1981.

[14] J. J. Koenderink. Scale-time. *Biol. Cybern.*, 58:159–162, 1988.

[15] A. Kadi and D.G.B. Edelen. A gauge theory of dislocations and disclinations. In H. Araki, Ehlers. J., K. Hepp, R. Kippenhahn, H. A. Weidenmüller, and J. Zittartz, editors, *Lecture Notes in Physics*, volume 174. Springer-Verlag, Berlin, 1983.

[16] H. Kleinert. *Gauge Fields in Condensed Matter*, volume 1–2. World Scientific Publishing C0. Pte. Ltd., Singapore, 1989.

[17] P. J. Olver. *Applications of Lie Groups to Differential Equations*, volume 107 of *Graduate Texts in Mathematics*. Springer-Verlag, 1986. Second Edition 1993.

[18] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, New York, Paris, San Diego, San Francisco, São Paulo, Sydney, Tokyo and Toronto, 1982.

[19] G. E. Healey and S. A. Shafer, editors. *Physics-based Vision: Principles and Practice; Color*. Jones and Bartlett Publishers, Boston London, 1992.

[20] A. Lakhtakia, V. K. Varadan, and V. V. Varadan. *Time-Harmonic Electromagnetic Fields in Chiral Media.* Number 335 in Lecture Notes in Physics. Springer-Verlag, 1989.

[21] L. B. Wolff, S. A. Shafer, and G. E. Healey, editors. *Physics-based Vision: Principles and Practice; Radiometry.* Jones and Bartlett Publishers, Boston London, 1992.

[22] E. Cartan. *Sur les variétés à connexion affine et la théorie de la relativité généralisée.* Gauthiers-Villars, 1955.

[23] F. Preteux. Watershed and skeleton by influence zones: A distance-based approach. *Journal of Mathematical Imaging and Vision*, 1(3):239–256, September 1992. Special Issue: Image Algebra and Morphological Image Processing.

[24] EUMETSAT. *User Handbook*, meteolorogical archive and retreival facilities edition, July 1998. Issue 2.

[25] J. Sporring and J. Weickert. Information measures in scale-spaces. *IEEE Trans. Information Theory*, 45:–, 1999.

[26] A. Katok and B. Hasselblatt. *Introduction to the modern theory of dynamical systems.* Cambridge University Press, 1995.

[27] A. H. Salden, B. M. ter Haar Romeny, and M. A. Viergever. Linearised Euclidean shortening flow of curve geometry. Revised version submitted to International Journal of Computer Vision, May 1996.

[28] G. Calugareanu. Sur les classes d'isotopie des noeuds tridimensionels et leaurs invariants. *Czechoslovak Math. J.*, 11:588–625, 1961.

[29] W. F. Pohl. The self-linking number of a closed space curve. *Journal of Mathematics and Mechanics*, 17(10):975–985, 1968.

[30] V. A. Vassiliev. Topology of complements to discriminants and loop spaces. *Advances in Soviet Mathematics*, 1:9–21, 1990.

[31] D. Bar-Natan. *Perturbative aspects of the topological quantum field theory.* PhD thesis, Princeton University, 1991.

[32] M. Freedman, Z.-X. He, and Z. Wang. Möbius energies of knots and unknots. *Ann. Math.*, 139:1–50, 1994.

[33] R. Bott and C. Taubes. On the self-linking of knots. *Journal of Mathematical Physics*, 35:5247–5287, 1994.

A Meteosat Images

A.1 Image Data and Acquisition

Meteosat [24] has three channels measuring radiance in $Wm^{-2}sr^{-1}$ in the visible spectrum $0.4 \leq \lambda \leq 1.1\mu\ m$), in the infrared spectrum $10.5 \leq \lambda \leq 12.5\mu\ m$) and in the absorption spectrum of water vapour $5.7 \leq \lambda \leq 7.1\mu\ m$). Highest possible spatial resolutions of the three channels on the ground are $2.5 \times 2.5\ km^2$, $5 \times 5\ km^2$

and $5 \times 5\ km^2$, respectively. The temporal resolution of each channel, i.e. the time of image acquisition by scanning, is 25 min with a 5 min delay before next acquisition. For all three channels the dynamic sensitivity is 8 $bits$ and the spectrum is covered from 24 h.

The radiance R measured together with the known solar irradiance S allows us to measure for the visible channel the albedo A:

$$A = \frac{R}{S}. \tag{1}$$

In case of cloudy air and sharp incidence angle of solar radiation or viewing angle with the earth's surface unit normal vector the isotropic reflection principle does not apply anymore. Consequently, the measured radiance includes also absorption and diffusion phenomena.

In case of the infrared and water vapour channel the radiance R given by:

$$R = \alpha(C - C_0) \tag{2}$$

with calibration parameters α and C_0, supplied by Meteosat [24], can be related to the blackbody irradiance as follows:

$$R = \int_{\lambda_1}^{\lambda_2} d\lambda F(\lambda, T') I(\lambda, T) = \int_{\lambda_1}^{\lambda_2} d\lambda F(\lambda, T') \epsilon(\lambda) B(\lambda, T), \tag{3}$$

where I is the spectral luminance of the observed object, ϵ is the emissivity of the object, F is the filter characteristics of the channel at operational temperature T' and B is the blackbody irradiance at absolute temperature T.

A.2 Image Data Geolocation and Geocoding

Meteosat is geostationary, i.e., it is located over approximately the Greenwich meridian and the earth equator at $R_S = 42.164,0\ km$ from the earth's center. Assume the world Cartesian coordinate system to be located at the earth's center with Meteosat on the x-axis, such that its coordinates are $X_S = (R_S, 0, 0)$, and the earth's equator in the xy-plane. Now a geocentric spherical coordinate system (r, θ_c, ψ) is related to the world Cartesian coordinate system through the following transformation rules:

$$(x, y, z) = (r \cos\theta_c \cos\psi, r \cos\theta_c \sin\psi, r \sin\theta_c) \tag{4}$$

and

$$(r^2, \tan\theta_c, \tan\psi) = \left(x^2 + y^2 + z^2, \frac{z}{\sqrt{x^2 + y^2}}, \frac{y}{x} \right). \tag{5}$$

As the earth's surface is not exactly spherical it's important for the computation of the solar irradiation of the earth's surface to change from geocentric to geodetic or geographic latitudes θ_d. The latter latitude is defined as the angle between the local normal at the earth's surface and the the equatorial plane. In order to find the relationship between the geodetic or geographic latitudes, let us first parametrise the earth's surface as an ellipsoid in terms of the world coordinates:

$$S(x, y, z) = \frac{x^2 + y^2}{R_e^2} + \frac{z^2}{R_p^2} = 1, \tag{6}$$

in which the equatorial radius $R_e = 6378.140\ km$ and the polar radius $R_p = 6356.755\ km$. Now the unit normal vector $\hat{n}$ to the earth's surface is simply defined as:

$$\hat{n} = \frac{\nabla S}{||\nabla S||} \tag{7}$$

such that the geodetic latitude is in world coordinates:

$$\tan\theta_d = \frac{\frac{z}{R_p^2}}{\frac{\sqrt{x^2+y^2}}{R_e^2}}. \tag{8}$$

Now it's readily shown that the geocentric and geodetic latitudes are related as follows:

$$\tan\theta_d = \frac{R_e^2}{R_p^2}\tan\theta_c = \frac{1}{1-\Omega^2}, \tag{9}$$

where the oblateness Ω is given by:

$$\Omega = \frac{R_e - R_p}{R_e}. \tag{10}$$

Using the earth's surface parametrisation in world coordinates one can now also give readily that parametrisation in spherical or geographical coordinates:

$$r^2 = r_{\theta_c}^2 = \frac{R_e^2 R_p^2}{R_p^2\cos^2\theta_c + R_e^2\sin^2\theta_c},\ \tan\theta_c = (1-\Omega^2)\tan\theta_d. \tag{11}$$

In the next sections we treat the conversions between geographic and digital data.

A.2.1 Geographic to Digital

The vertical and horizontal resolution $\delta_V = \delta_H = \delta$ are just the fraction of Meteosat's field of view (2π radials) and the number of lines N_l or sample points N_s depending on whether the image concerns an infrared and water vapour image, or an image in the visual spectrum.

The digital pixel coordinates in an image corresponding to a particular geographic coordinate $P = (x, y, z)$ can be obtained simply by expressing the geocentric latitude in terms of the geodetic latitude (see equation 9), by expressing the surface parametrisation in terms of equatorial and polar radius, oblateness and geodetic latitude (see equation 11), and by expressing the world coordinates through the spherical into geographical coordinates (see equation 4). With $X_S = (R_S, 0, 0)$ and taking $O' = (x, y, 0)$ the line number n_l and the sample number n_s on a line can be given by:

$$\arctan\angle PSO' = \arctan\frac{O'P}{O'S} = \frac{z}{\sqrt{y^2+(R_S-x)^2}} = n_l\delta_V, \tag{12}$$

$$\arctan\angle O'SO = \arctan\frac{OO'}{OS} = \arctan\frac{y}{(R_S-x)} = n_s\delta_H. \tag{13}$$

The visibility condition boils down to the requirement that the angle between viewing angle and earth's surface normal is less than 90 degrees:

$$(X_S - P, \hat{n}) \geq 0. \tag{14}$$

A.2.2 Digital to Geographic

It is more important, however, to express the digital data in geographical data because we need the reflection of solar irradiation at the corresponding geographical coordinates to be fed in to the air-pollution forecast and simulation systems. In particular the transmission function T of the solar radiation $\vec{I}$ projected onto and averaged over the related surface area, I_s, has to be computed and compared to the observed reflected solar radiation I_o. For the latter we need to integrate the inner product of the solar irradiation vector density field and the surface unit normal vector field to the earth's surface, over the earth's surface area spanned by a digital pixel and multiply by half the total solid angle, i.e., the field of view (assuming isotropic reflection):

$$T = 2\pi \frac{I_o}{I_s};\ I_s = (\hat{n}, \vec{I}) dS,\ dS = \hat{n}_x dy \wedge dz + \hat{n}_y dz \wedge dx + \hat{n}_z dx \wedge dy$$

In this context it is equally crucial to parametrise the transmission function T in terms of geographical coordinates. Thereto we have to solve a simple problem in finding a condition for the intersection of a viewing line and the earth's surface. Let the position of the satellite X_S and the point viewed on the earth's surface P determine these so-called viewing lines l:

$$l(k) = X_S + kv,\ v = (X_S - P) = (p, q, r),\ k \in \mathrm{R}. \tag{15}$$

The angles along vertical and horizontal directions, a_V and a_H, are available through equations (12) and (13), and are linked to the v-components as follows:

$$\begin{aligned} \tan a_V &= \frac{r}{\sqrt{p^2 + q^2}}, \\ \tan a_H &= \frac{q}{p}, \end{aligned} \tag{16}$$

in which we may choose $p = 1$ without loss of generality.

Now the intersection condition for viewing line l with a point P on the earth's surface boils down to substituting the viewing line representation $l(k)$ into the earth's (surface world coordinates) parametrisation (6), making above choice for p and substituting the relations between the remaining v-components, q and r, and the angles along vertical and horizontal directions, a_V and a_H. One obtains a quadratic equation for k with coefficients all expressed in observed entities. In case of complex conjugated roots there's no intersection for that particular digital point. Whereas in the cases of one or two real roots the minimum of the sets of solutions results in a really visible intersection point.

After substitution the found k-value into the viewing line $l(k)$ (and using $p = 1$) one has obtained the world coordinates of the observed earth's surface point P. Now from these world coordinates one retains readily the geocentric and geographic coordinates using the transformations rules presented in the previous section.

B Ansi-C Codes

In the following subsections one finds the Ansi-C codes for the computation of the discrete ordinary derivatives, for the segmentation algorithms presented in Section 3.2.2, and for the dynamic scale-space filtering algorithm presented in Section 3.2.3.

B.1 derivatives.c

```
#include <stdio.h>
#include <math.h>
#include <inrimage/image.h>

extern struct image *image_();
extern char *i_malloc();

extern int debug_;


void alloc_threedimmatrix

        (double ****threedimmatrix,
         long dimX,                          /* size in x direction */
         long dimY,                          /* size in y direction */
         long dimT)                          /* size in t direction */

 allocates storage for threedimmatrix of size dimX * dimY * dimT */


{
long i,j;

*threedimmatrix = (double ***) malloc (dimX * sizeof(double **));
if (*threedimmatrix == NULL)
    {
    printf("alloc_threedimmatrix:  not enough storage
available\n");
    exit(1);
    }
for (i=0; i<dimX; i++)
     {
     (*threedimmatrix)[i] = (double **) malloc (dimY * sizeof(double *));
     if ((*threedimmatrix)[i] == NULL)
         {
         printf("alloc_threedimmatrix:  not enough storage
available\n");
         exit(1);
         }
          for (j=0; j<dimY; j++)
          {
     (*threedimmatrix)[i][j] = (double *) malloc (dimT * sizeof(double));
     if ((*threedimmatrix)[i][j] == NULL)
         {
         printf("alloc_threedimmatrix:  not enough storage
available\n");
         exit(1);
         }
```

```
    }
}
return;
}


void disalloc_threedimmatrix

      (double ***threedimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
       long dimT)                                  /* size in t direction */

 disallocates storage for threedimmatrix of size dimX * dimY * dimT */

{
long i,j;


for (i=0; i<dimX; i++)
          {
          for (j=0; j<dimY; j++)
                    {
                         free(threedimmatrix[i][j]);
                    }
          }
free(threedimmatrix);
return;
}



call_derivative( char *in, char *out)
{
          struct image *nfi, *nfo;
          Fort_int lfmti[9], lfmto[9];
          long dimXV, dimYZ;
          long dimX, dimY, dimT,dimV;
          float *nfibuffer;
          float *nfobuffer;
          int size, type, imagetype;

          nfi = image_(in,"e"," ",lfmti);

          dimXV = lfmti[I_DIMX];
          dimYZ = lfmti[I_DIMY];
          size = lfmti[I_BSIZE];
          type = lfmti[I_TYPE];
          dimX = lfmti[I_NDIMX];
          dimY = lfmti[I_NDIMY];
```

```
        dimT = lfmti[I_NDIMZ];
        dimV = lfmti[I_NDIMV];
        imagetype = lfmti[I_EXP];

        nfibuffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));

        c_lptset(nfi,1);
        c_lecflt(nfi,dimYZ,nfibuffer);
        fermnf_(&nfi);

        lfmto[I_DIMX] = dimXV;
        lfmto[I_DIMY] = dimYZ;
        lfmto[I_BSIZE] = sizeof(float) ;
        lfmto[I_TYPE] = REELLE;
        lfmto[I_NDIMX] = dimX;
        lfmto[I_NDIMY] = dimY;
        lfmto[I_NDIMZ] = dimT;
        lfmto[I_NDIMV] = 1;
        lfmto[I_EXP] = imagetype;


        nfo = image_(out,"c"," ",lfmto);
        nfobuffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));

        do_derivative(dimX,dimY,dimT,&nfibuffer,nfobuffer);

        c_ecr(nfo,dimYZ,nfobuffer);
        fermnf_(&nfo);
        i_Free(&nfobuffer);

}


do_derivative(long dimX,
              long dimY,
              long dimT,
              float **nfibuffer,
              float *nfobuffer)
{
        long i, j, k, l;
        double ***threedimmatrix;
        double ***xcomthreedimmatrix;
        double ***ycomthreedimmatrix;
        float * tmpbuf ;

        alloc_threedimmatrix (&threedimmatrix, dimX+2, dimY+2, dimT+2);
        tmpbuf = *nfibuffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
```

```
        for (i=0; i<dimX; i++)
        threedimmatrix[i+1][j+1][k+1] = (double) *(tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfibuffer) ;



        derivative(dimX,dimY,dimT,threedimmatrix,xcomthreedimmatrix,ycomthreedimmatrix);

        for (i=0; i<dimX; i++)
          for (j=0; j<dimY; j++)
            for (k=0; k<dimT; k++)
              {
                  nfobuffer[i + j*dimX + k*dimX*dimY] = (float)
threedimmatrix[i+1][j+1][k+1];
              }



        disalloc_threedimmatrix (threedimmatrix, dimX+2, dimY+2,dimT+2);



}

derivative( long dimX,
                long dimY,
                long dimT,
                double ***f)
{
long i, j, k;                                   /* loop variables */
double ***g;                                    /* work copy of f */


alloc_threedimmatrix (&g, dimX+2, dimY+2, dimT+2);


 --- copy f into g --- */

for (i=1; i<=dimX; i++)
 for (j=1; j<=dimY; j++)
   for (k=1; k<=dimT; k++)
     g[i][j][k] = f[i][j][k];


 --- create ghost boundaries for g --- */

for (i=1; i<=dimX; i++)
  for (j=1; j<=dimY; j++)
```

```
      {
       g[i][j][0] = g[i][j][1];
       g[i][j][dimT+1] = g[i][j][dimT];
      }

for (j=1; j≤dimY; j++)
   for (k=0; k≤dimT+1; k++)
      {
       g[dimX+1][j][k] = g[dimX][j][k];
      }

for (i=1; i≤dimX+1; i++)
   for (k=0; k≤dimT+1; k++)
      {
       g[i][dimY+1][k] = g[i][dimY][k];
      }

for (j=1; j≤dimY+1; j++)
   for (k=0; k≤dimT+1; k++)
      {
       g[0][j][k] = g[1][j][k];
      }

for (i=0; i≤dimX+1; i++)
   for (k=0; k≤dimT+1; k++)
      {
       g[i][0][k] = g[i][1][k];
      }



 —- computation of e.g. discrete x-derivative —- */

for (i=1; i≤dimX; i++)
 for (j=1; j≤dimY; j++)
    for (k=1; k≤dimT; k++)
       {

              f[i][j][k] =
                 ((g[i+1][j][k] - g[i][j][k]) +
                  (g[i][j][k] - g[i-1][j][k]) +
                  (g[i+1][j+1][k] - g[i][j][k]) / sqrt(2.0) +
                  (g[i][j][k] - g[i-1][j-1][k])/sqrt(2.0)) / 4.0;

       }



disalloc_threedimmatrix (g, dimX+2, dimY+2, dimT+2);
```

```
return;

}


static char Ucmd[]=
"[-D] [-k nb_Ko] [input | -] [output]";

static char Udetail[]=
"\tDERIVATIVE (2,1) SPATIO-TEMPORAL IMAGE\n\
\t\n\
\tAlfons Salden, 09/98\n\
\t\n\
\tinput :  input image of type float( stdin if '-')\n\
\toutput :  output image of type float( stdout if
absent)";

main(argc,argv)
int argc;
char **argv;
{

        char in[256], out[256];

        inr_init(argc,argv," ",Ucmd,Udetail);

        igetopt( "", "%s", in, "", 0, "", 0);
        igetopt( "", "%s", out, "", 0, "", 0);

        if(tstopts())
                iusage_(Ucmd,Udetail);

        call_derivative(in,out);

        return 0;
}
```

B.2 signature.c

```
#include <stdio.h>
#include <math.h>
#include <inrimage/image.h>

extern struct image *image_();
extern char *i_malloc();

extern int debug_;
```

```
void alloc_threeddimmatrix

      (double ****threeddimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
       long dimT)                                  /* size in t direction */

 allocates storage for threeddimmatrix of size dimX * dimY * dimT */


{
long i,j;

*threeddimmatrix = (double ***) malloc (dimX * sizeof(double **));
if (*threeddimmatrix == NULL)
   {
   printf("alloc_threeddimmatrix:  not enough storage
available\n");
   exit(1);
   }
for (i=0; i<dimX; i++)
    {
    (*threeddimmatrix)[i] = (double **) malloc (dimY * sizeof(double *));
    if ((*threeddimmatrix)[i] == NULL)
       {
       printf("alloc_threeddimmatrix:  not enough storage
available\n");
       exit(1);
       }
        for (j=0; j<dimY; j++)
        {
    (*threeddimmatrix)[i][j] = (double *) malloc (dimT * sizeof(double));
    if ((*threeddimmatrix)[i][j] == NULL)
       {
       printf("alloc_threeddimmatrix:  not enough storage
available\n");
       exit(1);
       }
    }
}
return;
}


void disalloc_threeddimmatrix

      (double ***threeddimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
```

```
        long dimT)                                  /* size in t direction */

 disallocates storage for threeddimmatrix of size dimX * dimY * dimT*/

{
long i,j;


for (i=0; i<dimX; i++)
        {
        for (j=0; j<dimY; j++)
                {
                    free(threeddimmatrix[i][j]);
                }
        }
free(threeddimmatrix);
return;
}


void alloc_threeucdimmatrix

      (unsigned char ****threeucdimmatrix,
       long dimX,                                   /* size in x direction */
       long dimY,                                   /* size in y direction */
       long dimT)                                   /* size in t direction */

 allocates storage for threeucdimmatrix of size dimX * dimY * dimT */


{
long i,j;

*threeucdimmatrix = (unsigned char ***) malloc (dimX * sizeof(unsigned char
**));
if (*threeucdimmatrix == NULL)
   {
   printf("alloc_threeucdimmatrix:  not enough storage
available\n");
   exit(1);
   }
for (i=0; i<dimX; i++)
    {
    (*threeucdimmatrix)[i] = (unsigned char **) malloc (dimY * sizeof(unsigned
char *));
    if ((*threeucdimmatrix)[i] == NULL)
        {
        printf("alloc_threeucdimmatrix:  not enough storage
available\n");
        exit(1);
```

```
        }
         for (j=0; j<dimY; j++)
         {
    (*threeucdimmatrix)[i][j] = (unsigned char *) malloc (dimT * sizeof(unsigned
char));
    if ((*threeucdimmatrix)[i][j] == NULL)
        {
        printf("alloc_threeucdimmatrix:  not enough storage
available\n");
        exit(1);
        }
    }
}
return;
}


void disalloc_threeucdimmatrix

      (unsigned char ***threeucdimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
       long dimT)                                  /* size in t direction */

 disallocates storage for threeucdimmatrix of size dimX * dimY * dimT */

{
long i,j;


for (i=0; i<dimX; i++)
        {
        for (j=0; j<dimY; j++)
                {
                    free(threeucdimmatrix[i][j]);
                }
        }
free(threeucdimmatrix);
return;
}


call_signature(char *x1der, char *y1der, char * x2der, char *y2der, char *x1y1der,
char *out)
{
        struct image *nfx1, *nfy1, *nfx2, *nfy2, *nfx1y1, *nfout;
        Fort_int lfmtx1[9], lfmty1[9], lfmtx2[9], lfmty2[9], lfmtx1y1[9], lfmtout[9];
        long dimXV, dimYZ;
        long dimX, dimY, dimT,dimV;
        float *nfx1buffer, *nfy1buffer, *nfx2buffer, *nfy2buffer, *nfx1y1buffer;
```

```
unsigned char *nfoutbuffer;
int size, type, imagetype;

nfx1 = image_(x1der,"e"," ",lfmtx1);
nfy1 = image_(y1der,"e"," ",lfmty1);
nfx2 = image_(x2der,"e"," ",lfmtx2);
nfy2 = image_(y2der,"e"," ",lfmty2);
nfx1y1 = image_(x1y1der,"e"," ",lfmtx1y1);

dimXV = lfmtx1[I_DIMX];
dimYZ = lfmtx1[I_DIMY];
size = lfmtx1[I_BSIZE];
type = lfmtx1[I_TYPE];
dimX = lfmtx1[I_NDIMX];
dimY = lfmtx1[I_NDIMY];
dimT = lfmtx1[I_NDIMZ];
dimV = lfmtx1[I_NDIMV];
imagetype = lfmtx1[I_EXP];

nfx1buffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));
nfy1buffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));
nfx2buffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));
nfy2buffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));
nfx1y1buffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));

c_lptset(nfx1,1);
c_lecflt(nfx1,dimYZ,nfx1buffer);
fermnf_(&nfx1);
c_lptset(nfy1,1);
c_lecflt(nfy1,dimYZ,nfy1buffer);
fermnf_(&nfy1);
c_lptset(nfx2,1);
c_lecflt(nfx2,dimYZ,nfx2buffer);
fermnf_(&nfx2);
c_lptset(nfy2,1);
c_lecflt(nfy2,dimYZ,nfy2buffer);
fermnf_(&nfy2);
c_lptset(nfx1y1,1);
c_lecflt(nfx1y1,dimYZ,nfx1y1buffer);
fermnf_(&nfx1y1);

lfmtout[I_DIMX] = dimXV;
lfmtout[I_DIMY] = dimYZ;
lfmtout[I_BSIZE] = sizeof(unsigned char);
lfmtout[I_TYPE] = FIXE;
lfmtout[I_NDIMX] = dimX;
lfmtout[I_NDIMY] = dimY;
lfmtout[I_NDIMZ] = dimT;
lfmtout[I_NDIMV] = dimV;
lfmtout[I_EXP] = 0;
```

```
        nfout = image_(out,"c"," ",lfmtout);
        nfoutbuffer = (unsigned char *)i_malloc(dimXV * dimYZ *
sizeof(unsigned char));

        do_signature(dimX,
                dimY,
                dimT,
                &nfx1buffer,
                &nfy1buffer,
                &nfx2buffer,
                &nfy2buffer,
                &nfx1y1buffer,
                nfoutbuffer);

        c_ecr(nfout,dimYZ,nfoutbuffer);
        fermnf_(&nfout);
        i_Free(&nfoutbuffer);


}


do_signature( long dimX,
                long dimY,
                long dimT,
                float **nfx1buffer,
                float **nfy1buffer,
                float **nfx2buffer,
                float **nfy2buffer,
                float **nfx1y1buffer,
                    unsigned char *nfoutbuffer)
{
        long i, j, k, l;
        double ***x1threeddimmatrix, ***y1threeddimmatrix,
***x2threeddimmatrix, ***y2threeddimmatrix, ***x1y1threeddimmatrix;
        unsigned char ***outthreeucdimmatrix;
        float * x1tmpbuf, * y1tmpbuf, * x2tmpbuf, * y2tmpbuf, * x1y1tmpbuf;

        alloc_threeddimmatrix (&x1threeddimmatrix, dimX, dimY, dimT);
        x1tmpbuf = *nfx1buffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        x1threeddimmatrix[i][j][k] = (double) *(x1tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfx1buffer) ;
        alloc_threeddimmatrix (&y1threeddimmatrix, dimX, dimY, dimT);
        y1tmpbuf = *nfy1buffer ;
```

```
        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        y1threeddimmatrix[i][j][k] = (double) *(y1tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfy1buffer) ;
        alloc_threeddimmatrix (&x2threeddimmatrix, dimX, dimY, dimT);
        x2tmpbuf = *nfx2buffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        x2threeddimmatrix[i][j][k] = (double) *(x2tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfx2buffer) ;
        alloc_threeddimmatrix (&y2threeddimmatrix, dimX, dimY, dimT);
        y2tmpbuf = *nfy2buffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        y2threeddimmatrix[i][j][k] = (double) *(y2tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfy2buffer) ;
        alloc_threeddimmatrix (&x1y1threeddimmatrix, dimX, dimY, dimT);
        x1y1tmpbuf = *nfx1y1buffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        x1y1threeddimmatrix[i][j][k] = (double) *(x1y1tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfx1y1buffer) ;
        alloc_threeucdimmatrix (&outthreeucdimmatrix, dimX, dimY, dimT);
        signature(dimX,
                dimY,
                dimT,
                x1threeddimmatrix,
                y1threeddimmatrix,
                x2threeddimmatrix,
                y2threeddimmatrix,
                x1y1threeddimmatrix,
                outthreeucdimmatrix);

        disalloc_threeddimmatrix (x1threeddimmatrix,dimX,dimY,dimT);
        disalloc_threeddimmatrix (y1threeddimmatrix,dimX,dimY,dimT);
        disalloc_threeddimmatrix (x2threeddimmatrix,dimX,dimY,dimT);
        disalloc_threeddimmatrix (y2threeddimmatrix,dimX,dimY,dimT);
        disalloc_threeddimmatrix (x1y1threeddimmatrix,dimX,dimY,dimT);
```

```
            for (i=0; i<dimX; i++)
              for (j=0; j<dimY; j++)
                for (k=0; k<dimT; k++)
                  {
                    nfoutbuffer[i + j*dimX + k*dimX*dimY] =
outthreeucdimmatrix[i][j][k];
                  }

          disalloc_threeucdimmatrix (outthreeucdimmatrix,dimX,dimY,dimT);



}

signature(long dimX,
          long dimY,
          long dimT,
          double ***x1f,
          double ***y1f,
          double ***x2f,
          double ***y2f,
          double ***x1y1f,
          unsigned char ***outf)
{
long i, j, k;                                        /* loop variables */
double temp;

for (k=0; k<dimT; k++)
  for (i=0; i<dimX; i++)
    for (j=0; j<dimY; j++)
      {
        temp =                                       /* see bottom latex code */
        if(temp > 0)
          {
           outf[i][j][k] = 0;
          }
        else
          {
            outf[i][j][k] = 1;
          }
      }
return;

}


static char Ucmd[]=
"[-D] [-k nb_Ko] [-x1der x1der] [-y1der y1der] [-x1y1der
```

```
x1y1der] [-x2der x2der] [-y2der y2der] [-out out]";

static char Udetail[]=
"\tSEGMENTATION ON BASIS OF SIGNATURE\n\
\t\n\
\tAlfons Salden, 09/98\n\
\t\n\
\t-x1der :  filename x derivative of input image\n\
\t-y1der :  filename y derivative of input image\n\
\t-x1y1der :  filename x1y1 derivative of input image\n\
\t-x2der :  filename x2 derivative of input image\n\
\t-y2der :  filename y2 derivative of input image\n\
\t-out :  filename of output image\n\
\tinput :  derivative images of type float( stdin if
'-')\n\
\toutput :  output image of type unsigned char( stdout if
absent)";

main(argc,argv)
int argc;
char **argv;
{

   char x1der[256], y1der[256], x2der[256], y2der[256], x1y1der[256];
   char out[256];

   inr_init(argc,argv,"",Ucmd,Udetail);

   igetopt1( "-x1der", "%s", x1der);
   igetopt1( "-y1der", "%s", y1der);
   igetopt1( "-x2der", "%s", x2der);
   igetopt1( "-y2der", "%s", y2der);
   igetopt1( "-x1y1der", "%s",x1y1der);
   igetopt1( "-out", "%s", out);

   if(tstopts())
      iusage_(Ucmd,Udetail);


   call_signature(x1der,y1der,x2der,y2der,x1y1der,out);

   return 0;
}

 signature for segmentation in fore and background dynamics
   of temporal image sequence */

temp = t1f[i][j][k] * t2f[i][j][k];

 signature for segmentation in fore and background dynamics
```

```
    of spatial slice of temporal image sequence */

temp = (x1f[i][j][k] * x2f[i][j][k] + y1f[i][j][k] * x1y1f[i][j][k]) * x1f[i][j][k] +
          (x1f[i][j][k] * x1y1f[i][j][k] + y1f[i][j][k] * y2f[i][j][k]) * y1f[i][j][k];

 signature for segmentation on the basis of the phase of the
    normalised flowline curvature vector field spatial slice of
    temporal image sequence */

temp = x1f[i][j][k] * ((x1y1f[i][j][k] * x1f[i][j][k] + y2f[i][j][k] * y1f[i][j][k]) *
                                (x1f[i][j][k] * x1f[i][j][k] + y1f[i][j][k] * y1f[i][j][k]) -
                                 y1f[i][j][k] * (x1f[i][j][k] * x2f[i][j][k] * x1f[i][j][k] +
                                 2.0 * x1f[i][j][k] * x1y1f[i][j][k] * y1f[i][j][k] +
                                 y1f[i][j][k] * y2f[i][j][k] * y1f[i][j][k])) -
          y1f[i][j][k] * ((x2f[i][j][k] * x1f[i][j][k] + x1y1f[i][j][k] * y1f[i][j][k]) *
                                (x1f[i][j][k] * x1f[i][j][k] + y1f[i][j][k] * y1f[i][j][k]) -
                                 x1f[i][j][k] * (x1f[i][j][k] * x2f[i][j][k] * x1f[i][j][k] +
                                 2.0 * x1f[i][j][k] * x1y1f[i][j][k] * y1f[i][j][k] +
                                 y1f[i][j][k] * y2f[i][j][k] * y1f[i][j][k]));
 signature for segmentation on the basis of the isophote
    curvature */

temp = - (x1f[i][j][k] * x1f[i][j][k] * y2f[i][j][k] +
             y1f[i][j][k] * y1f[i][j][k] * x2f[i][j][k] -
             2 * x1f[i][j][k] * x1y1f[i][j][k] * y1f[i][j][k]);
```

B.3 dynamic.c

```
#include <stdio.h>
#include <math.h>
#include <inrimage/image.h>

extern struct image *image_();
extern char *i_malloc();

extern int debug_;


void alloc_threedimmatrix

      (double ****threedimmatrix,
       long dimX,                          /* size in x direction */
       long dimY,                          /* size in y direction */
       long dimT)                          /* size in t direction */



{
long i,j;
```

```
*threedimmatrix = (double ***) malloc (dimX * sizeof(double **));
if (*threedimmatrix == NULL)
    {
    printf("alloc_threedimmatrix:  not enough storage
available\n");
    exit(1);
    }
for (i=0; i<dimX; i++)
     {
     (*threedimmatrix)[i] = (double **) malloc (dimY * sizeof(double *));
     if ((*threedimmatrix)[i] == NULL)
         {
         printf("alloc_threedimmatrix:  not enough storage
available\n");
         exit(1);
         }
          for (j=0; j<dimY; j++)
          {
     (*threedimmatrix)[i][j] = (double *) malloc (dimT * sizeof(double));
     if ((*threedimmatrix)[i][j] == NULL)
         {
         printf("alloc_threedimmatrix:  not enough storage
available\n");
         exit(1);
         }
     }
}
return;
}


void disalloc_threedimmatrix

      (double ***threedimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
       long dimT)                                  /* size in t direction */


{
long i,j;


for (i=0; i<dimX; i++)
          {
          for (j=0; j<dimY; j++)
                    {
                         free(threedimmatrix[i][j]);
                    }
          }
```

```
free(threedimmatrix);
return;
}


void alloc_fourdimmatrix

        (double *****fourdimmatrix,
         long dimX,                          /* size in x direction */
         long dimY,                          /* size in y direction */
         long dimT,                          /* size in t direction */
         long dimS)                          /* size in s direction */


{
  long i,j,k;
  int tempo ;


  *fourdimmatrix = (double ****) malloc (dimX * sizeof(double ***));
  if (*fourdimmatrix == NULL)
    {
      printf("alloc_fourdimmatrix:  not enough storage
available\n");
      exit(1);
    }
  for (i=0; i<dimX; i++)
    {
      (*fourdimmatrix)[i] = (double ***) malloc (dimY * sizeof(double **));
      if ((*fourdimmatrix)[i] == NULL)
        {
          printf("alloc_fourdimmatrix:  not enough storage
available\n");
          exit(1);
        }
      for (j=0; j<dimY; j++)
        {
          (*fourdimmatrix)[i][j] = (double **) malloc (dimT * sizeof(double *));
          if ((*fourdimmatrix)[i][j] == NULL)
            {
              printf("alloc_fourdimmatrix:  not enough storage
available\n");
              exit(1);
            }
          for (k=0; k<dimT; k++)
            {
              (*fourdimmatrix)[i][j][k] = (double *) malloc (dimS *
sizeof(double));
```

```
                if ((*fourdimmatrix)[i][j][k] == NULL)
                  {
                    printf("alloc_fourdimmatrix:  not enough
storage available\n");
                    exit(1);
                  }
              }
          }
      }
  return;
}


void disalloc_fourdimmatrix

      (double ****fourdimmatrix,
       long dimX,                                  /* size in x direction */
       long dimY,                                  /* size in y direction */
       long dimT,                                  /* size in t direction */
       long dimS)                                  /* size in s direction */


{
long i, j, k;

total_mem_size -= sizeof(double ***)*dimX + sizeof(double **)*dimX*dimY +
  sizeof(double *)*dimX*dimY*dimT + sizeof(double)*dimX*dimY*dimT*dimS ;

for (i=0; i<dimX; i++)
  {
    for (j=0; j<dimY; j++)
          {
            for (k=0; k<dimT; k++)
                {
                    free(fourdimmatrix[i][j][k]);
                }
          }
  }
free(fourdimmatrix);
return;
}



call_dynamic(long dimS,
             char *in,
             char *out)
{
        struct image *nfi, *nfo;
        Fort_int lfmti[9], lfmto[9];
```

```
long dimXV, dimYZ;
long dimX, dimY, dimT,dimV;
float *nfibuffer;
float *nfobuffer;
int size, type, imagetype;

nfi = image_(in,"e"," ",lfmti);

dimXV = lfmti[I_DIMX];
dimYZ = lfmti[I_DIMY];
size = lfmti[I_BSIZE];
type = lfmti[I_TYPE];
dimX = lfmti[I_NDIMX];
dimY = lfmti[I_NDIMY];
dimT = lfmti[I_NDIMZ];
dimV = lfmti[I_NDIMV];
imagetype = lfmti[I_EXP];

nfibuffer = (float *)i_malloc(dimXV * dimYZ * sizeof(float));

c_lptset(nfi,1);
c_lecflt(nfi,dimYZ,nfibuffer);
fermnf_(&nfi);

lfmto[I_DIMX] = dimXV * dimS;
lfmto[I_DIMY] = dimYZ;
lfmto[I_BSIZE] = sizeof(float) ;
lfmto[I_TYPE] = REELLE;
lfmto[I_NDIMX] = dimX;
lfmto[I_NDIMY] = dimY;
lfmto[I_NDIMZ] = dimT;
lfmto[I_NDIMV] = dimS;
lfmto[I_EXP] = imagetype;


nfo = image_(out,"c"," ",lfmto);
nfobuffer = (float *)i_malloc(dimS * dimXV * dimYZ * sizeof(float));


do_dynamic(dimX,dimY,dimT,dimS,&nfibuffer,nfobuffer);

c_ecr(nfo,dimYZ,nfobuffer);

fermnf_(&nfo);

i_Free(&nfobuffer);

}
```

```
do_dynamic(long dimX,
           long dimY,
           long dimT,
           long dimS,
           float **nfibuffer,
           float *nfobuffer)
{
        long i, j, k, l;
        double ***threedimmatrix;
        float * tmpbuf ;

        alloc_threedimmatrix (&threedimmatrix, dimX+2, dimY+2, dimT+2);
        tmpbuf = *nfibuffer ;

        for (k=0; k<dimT; k++)
        for (j=0; j<dimY; j++)
        for (i=0; i<dimX; i++)
        threedimmatrix[i+1][j+1][k+1] = (double) *(tmpbuf + i + dimX*j +
dimX*dimY*k);
        free(*nfibuffer) ;


        for (l=0; l<dimS; l++)
                {
                  fprintf(stderr,"scale %d\n",l);
                  dynamic(dimX,dimY,dimT,threedimmatrix);

                  for (i=0; i<dimX; i++)
                    for (j=0; j<dimY; j++)
                      for (k=0; k<dimT; k++)
                        {
                          nfobuffer[l + dimS*(i + j*dimX + k*dimX*dimY)] =
(float) threedimmatrix[i+1][j+1][k+1];
                        }
                }


                disalloc_threedimmatrix (threedimmatrix, dimX+2,
dimY+2,dimT+2);


}

dynamic(long dimX,
        long dimY,
        long dimT,
        double ***f)
{
long i, j, k;                                   /* loop variables */
double ***g;                                    /* work copy of f */
```

```
alloc_threedimmatrix (&g, dimX+2, dimY+2, dimT+2);


 —- copy f into g —- */

for (i=1; i≤dimX; i++)
 for (j=1; j≤dimY; j++)
   for (k=1; k≤dimT; k++)
     g[i][j][k] = f[i][j][k];


 —- create ghost boundaries for g —- */

for (i=1; i≤dimX; i++)
  for (j=1; j≤dimY; j++)
    {
     g[i][j][0] = g[i][j][1];
     g[i][j][dimT+1] = g[i][j][dimT];
    }

for (j=1; j≤dimY; j++)
  for (k=0; k≤dimT+1; k++)
    {
     g[dimX+1][j][k] = g[dimX][j][k];
    }

for (i=1; i≤dimX+1; i++)
  for (k=0; k≤dimT+1; k++)
    {
     g[i][dimY+1][k] = g[i][dimY][k];
    }

for (j=1; j≤dimY+1; j++)
  for (k=0; k≤dimT+1; k++)
    {
     g[0][j][k] = g[1][j][k];
    }

for (i=0; i≤dimX+1; i++)
  for (k=0; k≤dimT+1; k++)
    {
     g[i][0][k] = g[i][1][k];
    }


 —- dynamic filtering —- */
```

```
for (i=1; i≤dimX; i++)
  for (j=1; j≤dimY; j++)
    for (k=1; k≤dimT; k++)
      {
        f[i][j][k] = g[i][j][k] +
         ( (g[i+1][j][k] - g[i][j][k]) /
(cosh(g[i+1][j][k] - g[i][j][k]) * cosh(g[i+1][j][k] - g[i][j][k]) * 26.0)
                  + (g[i-1][j][k] - g[i][j][k]) /
(cosh(g[i-1][j][k] - g[i][j][k]) * cosh(g[i-1][j][k] - g[i][j][k]) * 26.0)
                  + (g[i][j+1][k] - g[i][j][k]) /
(cosh(g[i][j+1][k] - g[i][j][k]) * cosh(g[i][j+1][k] - g[i][j][k]) * 26.0)
                  + (g[i][j-1][k] - g[i][j][k]) /
(cosh(g[i][j-1][k] - g[i][j][k]) * cosh(g[i][j-1][k] - g[i][j][k]) * 26.0)
                  + (g[i][j][k+1] - g[i][j][k]) /
(cosh(g[i][j][k+1] - g[i][j][k]) * cosh(g[i][j][k+1] - g[i][j][k]) * 26.0)
                  + (g[i][j][k-1] - g[i][j][k]) /
(cosh(g[i][j][k-1] - g[i][j][k]) * cosh(g[i][j][k-1] - g[i][j][k]) * 26.0)
                  + (g[i+1][j+1][k] - g[i][j][k]) /
(cosh(g[i+1][j+1][k] - g[i][j][k]) * cosh(g[i+1][j+1][k] - g[i][j][k]) * 26.0)
                  + (g[i+1][j-1][k] - g[i][j][k]) /
(cosh(g[i+1][j-1][k] - g[i][j][k]) * cosh(g[i+1][j-1][k] - g[i][j][k]) * 26.0)
                  + (g[i-1][j+1][k] - g[i][j][k]) /
(cosh(g[i-1][j+1][k] - g[i][j][k]) * cosh(g[i-1][j+1][k] - g[i][j][k]) * 26.0)
                  + (g[i-1][j-1][k] - g[i][j][k]) /
(cosh(g[i-1][j-1][k] - g[i][j][k]) * cosh(g[i-1][j-1][k] - g[i][j][k]) * 26.0)
                  + (g[i+1][j+1][k+1] - g[i][j][k]) /
(cosh(g[i+1][j+1][k+1] - g[i][j][k]) * cosh(g[i+1][j+1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i+1][j+1][k-1] - g[i][j][k]) /
(cosh(g[i+1][j+1][k-1] - g[i][j][k]) * cosh(g[i+1][j+1][k-1] - g[i][j][k]) * 26.0)
                  + (g[i+1][j-1][k+1] - g[i][j][k]) /
(cosh(g[i+1][j-1][k+1] - g[i][j][k]) * cosh(g[i+1][j-1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i+1][j-1][k-1] - g[i][j][k]) /
(cosh(g[i+1][j-1][k-1] - g[i][j][k]) * cosh(g[i+1][j-1][k-1] - g[i][j][k]) * 26.0)
                  + (g[i-1][j+1][k+1] - g[i][j][k]) /
(cosh(g[i-1][j+1][k+1] - g[i][j][k]) * cosh(g[i-1][j+1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i-1][j+1][k-1] - g[i][j][k]) /
(cosh(g[i-1][j+1][k-1] - g[i][j][k]) * cosh(g[i-1][j+1][k-1] - g[i][j][k]) * 26.0)
                  + (g[i-1][j-1][k+1] - g[i][j][k]) /
(cosh(g[i-1][j-1][k+1] - g[i][j][k]) * cosh(g[i-1][j-1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i-1][j-1][k-1] - g[i][j][k]) /
(cosh(g[i-1][j-1][k-1] - g[i][j][k]) * cosh(g[i-1][j-1][k-1] - g[i][j][k]) * 26.0)
                  + (g[i][j+1][k+1] - g[i][j][k]) /
(cosh(g[i][j+1][k+1] - g[i][j][k]) * cosh(g[i][j+1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i][j-1][k+1] - g[i][j][k]) /
(cosh(g[i][j-1][k+1] - g[i][j][k]) * cosh(g[i][j-1][k+1] - g[i][j][k]) * 26.0)
                  + (g[i][j+1][k-1] - g[i][j][k]) /
(cosh(g[i][j+1][k-1] - g[i][j][k]) * cosh(g[i][j+1][k-1] - g[i][j][k]) * 26.0)
                  + (g[i][j-1][k-1] - g[i][j][k]) /
(cosh(g[i][j-1][k-1] - g[i][j][k]) * cosh(g[i][j-1][k-1] - g[i][j][k]) * 26.0)
```

```
                + (g[i+1][j][k+1] - g[i][j][k]) /
(cosh(g[i+1][j][k+1] - g[i][j][k]) * cosh(g[i+1][j][k+1] - g[i][j][k]) * 26.0)
                + (g[i-1][j][k+1] - g[i][j][k]) /
(cosh(g[i-1][j][k+1] - g[i][j][k]) * cosh(g[i-1][j][k+1] - g[i][j][k]) * 26.0)
                + (g[i+1][j][k-1] - g[i][j][k]) /
(cosh(g[i+1][j][k-1] - g[i][j][k]) * cosh(g[i+1][j][k-1] - g[i][j][k]) * 26.0)
                + (g[i-1][j][k-1] - g[i][j][k]) /
(cosh(g[i-1][j][k-1] - g[i][j][k]) * cosh(g[i-1][j][k-1] - g[i][j][k]) * 26.0)
                  );
        }


disalloc_threedimmatrix (g, dimX+2, dimY+2, dimT+2);


return;

}


static char Ucmd[]=
"[-D] [-k nb_Ko] [-lda a] [-idimS dimS] [input | -]
[output]";

static char Udetail[]=
"\tDYNAMIC SCALE-SPACE FILTERING\n\
\t\n\
\tAlfons Salden, 09/98\n\
\t\n\
\tI_t = - j^I \n\
\t26-connectedness\n\
\t-idimS : number of scales\n\
\tinput :  no-vectorial input image of type float( stdin
if '-')\n\
\toutput :  vectorial output image of type float( stdout
if absent)";

main(argc,argv)
int argc;
char **argv;
{
        long dimS;
        char in[256], out[256];

        inr_init(argc,argv,"",Ucmd,Udetail);

        igetopt1("-idimS","%ld",&dimS);

        igetopt( "", "%s", in, "", 0, "", 0);
        igetopt( "", "%s", out, "", 0, "", 0);
```

```
        if(tstopts())
                iusage_(Ucmd,Udetail);

        call_dynamic(dimS,in,out);

        return 0;
}
```