B323 Differential methods for Mobile Robots

Javier Finat and Zhicheng Hou

Índice

Previous remarks: These notes corresponds to an introduction to the Chapter 3 of the module B_{32} (Automatic Navigation) of the matter B_3 (Robotics). From the mathematical viewpoint, it is necessary to have some basic knowledge of Computational Geometry, Differential Geometry, Differential Equations, Graph Theory, and Statistics.

From the computational viewpoint, it is convenient to be familiar with Object Oriented Programming (OOP) framework to ease the information exchange with other modules. Python provides a common framework to integrate OOP and functional frameworks. Thus, it is advisable to be familiar with Python, specially for AI related issues, with TensorFlow as paradigm under Pytorch. Some basic notions of Motion Analysis B_{23} in Computer Vision is advisable, also.

Subsections or paragraphs marked with an asterisk ($*$) have a higher difficulty and can be skipped in a first lecture.

0.1. Introduction to the chapter B323

Visual and/or effective Navigation involves the modelling of flows and the corresponding strategies for mobile devices. It is estimated in terms of processing and analysis of the information captured by mobile sensors. Most processing and analysis tools use time series analysis based on finite differences of "quantities" (position or orientation in the simplest cases). The information affects all PeCWA spaces appearing in the Basic Analytical Pipeline $\mathcal{P} \to \mathcal{C} \to \mathcal{W} \to \mathcal{A}$ appearing in precedent chapters.

The presence of noise, errors or uncertainty in discrete data, and their propagation along the successive fibrations of the BAP, suggest the development of "regularization strategies".as an output, one must obtain an ideal Piecewise Smooth (PS) model to be learned. The comparison between PS-models is initially carried out in terms of smooth maps $f : M \to N$ between PS-manifolds involving the space-time evolution of the "most meaningful" packaged data.

The *main goal* of differential models for Navigation is the modelling of multiple trajectories under multiple evolving constraints in the ambient space. By taking a "snapshot" of an evolving scene, if we would have a "complete" information about the scene, the localization for each mobile segement and its space-time evolution could be described in terms of evolving vectors (for ideal trajectories of multiple agents) and by evolving covectors (representing eventually changing constraints).

Incomplete information and uncertainty about data make things a little bit more complex. Structural models are the key to "stabilize" to obtain "more regular" models. The first paragraphs are devoted to sketch how Differential Geometry of Manifolds provides the simplest structural models for modelling objects and behaviours, and how can be "relaxed" to include "events". More realistic models for objects and behaviours are obtained by means of (deterministic vs stochastic) "perturbations" of structural models.

Structural models for motion analysis have been developed in the module B_{14} (Computational Kinematics). Their estimation form visual data has been developed in the module B_{23}, where we introduce structural models for motion based on the incorporation of different constraints for the Optical Flow. The self-adjustment to a expected trajectory is developed in this chapter by using basic notions of time series. A more structured approach in terms of different extensions of Kalman filters will be developed in the chapter 6 of the module B_{33} (Robot Kinematics).

Classical Analytical Mechanics provides *structural models* for motion equations (Hamilton-Jacobi), volumetric flow information (Liouville) and variational principles (Euler-Lagrange). All of them use ideal models arising from Differential Geometry, which can be adapted to Optimization and Control issues in the Phase space P corresponding to the total space TM of the tangent bundle τ_M (for the simultaneous analysis of multiple trajectories) or the total space T^*M of the cotangent bundle τ^*M.

The first "pseudo-deterministic" approaches use interpolation models $(1-\lambda)f_0 + \lambda f_1$ for functions $f_0, f_1 : X \to R$ defined on a space X linked to shapes or behaviours evaluation. The simplest probabilisticc model is the Bernouilli density for one variable given by $f(x;p) := p^x(1-p)^{1-x}$). By taking logarithms one obtains $x \log p + (1-x)\log(1-p)$ which vcna be interpreted as an affine interpolation between the functions $\log p$ and $\log(1-p)$. This basic example is extended to the n Bernouilli variables (with the same probability function) given by $f(\underline{x};p) := \Pi_{i=1}^n p^{x_i}(1-p)^{1-x_i}$ with a similar interpretation in terms of n interpolations.

All structural models have an *ideal character*, which is modified by using generic vs random perturbations along this chapter. Multivariate frameworks support geometric, analytical and statistical tools. Their simplest local representations are given by $(p \times n)$-matrices corresponding ot maps $\mathbb{R}^n \to \mathbb{R}^p$.

- *Generic perturbations of (scalar, vector, covector) fields* for *regular behaviours* are performed in the corresponding ambient space which can be described in terms of dynamical systems, whose topological version (compatible with deformations) is expressed in terms of k-jets (truncated Taylor polynomials) with the corresponding infinitesimal symmetries.
- *Random perturbations of fields* for *irregular behaviours* (including unexpected events), where matrices with constant vs variable coefficients play a fundamental role. Large dimensionality requires the development of efficient strategies to reduce the complexity, identify simpler eigenspaces, and decompose the ambient space.

The geometric framework provides the support for connecting local and global aspects. In particular, the above descriptions correspond to sections of (deterministic vs stochastic) bundles. In this way, one obtains a natural extension of the deterministic approaches performed in Differential Geometry A_1 and GAGA A_3.

Even in presence of landmarks and accurate systems for sensors and actuators, Automatic Navigation of autonomous Vehicles (ANAV) is inexact. It is necessary a Real-Time (RT) localization system aable of identifying and correcting errors in terms of multiple inputs. Furthermore, the maintenance of an absolute reference system (Euclidean framework) is not a good solution, because small errors are propagated and increased by iteration. In practice, one must combine metric and angular information involving near control points and visual angles to correct positioning or drift errors, e.g.

Along this chapter we are mainly focused towards terrestrial navigation, in despite of the interest for Automatic Navigation in the Maritime (including undersea navigation), Aerial or Spatial environments. Along the stay of JF in the GIAT (Nansha, Guandong, China), some applications to aerial transportation of loud charges by small floats have been developed by the authors, but they will be detailed later. Undersea and exterior space are crucial for civil and military applications, but require additional elements of Dynamic Control which will be developed in the module B_{34} (Robot Dynamics).

To bound errors linked to the updating of en Euclidean planar reference, it is convenient take affine references given by three affinely independent points $\mathbf{A}_0, \mathbf{A}_1$ and $\mathbf{A}_2$ in the affine plane $\mathbb{A}^2$. i.e. vectors $\mathbf{v}_1 = \overline{v\mathbf{A}_1}$ and $\mathbf{v}_2 = \overline{\mathbf{A}_0\mathbf{A}_2}$ are l.i. vectors. The current position of the robot $\mathbf{R}$ is determined as the intersection of the three circles passing through $\mathbf{R}$ and any pair of the affine reference $\mathcal{R}_a := \{\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2\}$ (each circle is computed from visual angles corresponding to each pair).

Reference points can be landmarks, intensity maxima or triple junctions appearing in the scene. For 3D scenes, the affine reference $\mathcal{R}_3$ is given by four points $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$ fulfilling the vectors $\mathbf{v}_i = \overline{\mathbf{A}_0\mathbf{A}_i}$ are linearly independent. Relative psoitioning is computer from the sign of affine coordinates of the robot $\mathbf{R}$ w.r.t. the reference. Each time the robot cuts out an affine reference line $\mathbf{a}_{ij} = \overline{\mathbf{A}_i\mathbf{A}_j}$ an affine coordinate changes its sign, and one must replace a reference point by other one. In practice, it is convenient work with redundant references to reinforce the information, and to avoid degenerate situations, where points do not impose independent conditions for self-localization.

The above remarks suggest the relevance of introducing landmarks or selecting 0–dimensional features for good localization. In presence of a large number of "visual features", tracing visual lines can gives a collection of intersecting lines, which is computational managed in terms of an *arrangement* $\mathcal{A}$ of lines for the planar case [1]. The first step consists of determining the region where the robot $\mathbf{R}$ is located.

The next step consists of choosing a "goodness function" for selecting the most appropriate reference affine, between different possible choices linked to "visiblle" points.

Analytical methods in Robotics involve to the use of differential and integral methods (or their discrete versions) for usual spaces appearing in the PACW sequence $\mathcal{P} \to \mathcal{C} \to \mathcal{W} \to \mathcal{A}$ involving Perception, configurations, Working and Action spaces. All of them are (semi-)analytic spaces $\mathcal{X} = (X; \mathcal{O}_X)$ where X is the topological support and
$mathcalO_X$ the set (in fact a ring) of regular functions defined on X. Furthermore, all maps between their topological supports $X \to Y$ are topological fibrations (see below for the characterizaiton).

The central part of the above sequence of fibrations is the most meaningful for usual approaches for this chapter; it is described in terms of the mechanical transfer map $\mu : \mathcal{C} \to \mathcal{W}$ and superimposed structures such as principal budnles (ideal case) or more general fiber bundles.In both cases, they are generated by sections given by different kinds of G-invariant (scalar, co-vector, tensor) fields or their dual G.invarianty forms.

Differential methods are described in terms of relations (Ordinary Diffe-

[1] See the chapter 6 of the odule B_{11} (Computational Geometry) for details.

rential Equations, initially) between low order derivatives or variation rates of "quantities" represented by coordinates, functions or any kind of fields. A basic example is given by $U(1) \to SU(2) \to SU(2)/U(1) \simeq \mathbb{CP}^1$ given by the assignations

$$e^{i\theta} \mapsto \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta + i\sin\theta & 0 \\ 0 & \cos\theta - i\sin\theta \end{pmatrix} quad\text{and} \quad \begin{pmatrix} z & w \\ --\bar{w} & \bar{z} \end{pmatrix} \mapsto [z : w]$$

where $\bar{z}$ is the conjugate of z. Let us remark that

- The unitary group $U(1)$ (parametrized by $\mathbb{S}^1$ of unit complex numbers) corresponding to electromagnetic interaction (structural group for Electromagnetism and, in particular, Signal Theory);,
- The Special Unitary Group $SU(2)$ (parametrized by $\mathbb{S}^3$ of unit quaternions) corresponding to the structural group for weak interaction (out of nucleus) in Particle Physics, which is crucial for a vector representation of Robotics in 3D (usefuñ for aerial navigation, also)
- The projective line whici is the support for regular transformations (simplest correspondences) of perspective representations given by collineations in traffic scenes, e.g.

Their corresponding transformations for ideal Kinematics are given by regular transformations of the correspodning Lie algrbas $\mathfrak{g} := T_eG$. They are represented by (products of) Jacobian matrices acting on G-invariant fields (or their transposed acting on G-invariant differential forms). They allow an ideal description of the simultaneous evolution in the space-time of "mixed quantities" representing trajectories (integral curves of vector fields) submitted to evolving constraints (integral hypersurfaces of differential forms).

Ideal descriptions are useful to bound the local variability or to show asympotic behaviours. In practice, it is necessary to complement them with a statistical treatment of data arising from kinematic sensors, and uncertainty about the scene or events:

1. A first extension of structural models incorporates *incomplete information* which are modelled in probabilistic terms, going from coarse Bayesian methods till more sophisticated random perturbations ot structural equations (in the Stochastic Differential Equations framework, e.g.). Parametric statistical models have a natural Riemannian structure w.r..t to the Cramer-Fisher-Rao metric, which allows a direct application of Differential Geometry methods.
2. A second extension incorporates *unexpected events* which can be modelled in terms of singularities of maps or, alternately, in terms of randomized algorithms such as the Montecarlo algorithm (fast but with low successful rate; used for coarse robot localization by using distance or odometry sensors, e.g.) or Las Vegas algorithm (better results, higher complexity to be reduced by using multithread architectures)

To accomplish this program, one develops a feedback between top-down and bottom-up approaches. The former one extends the differential or Piecewise Smooth (PS) approach to an analytic approach where several solutions are possible (represented by analytical branches), which are managed in algebraic terms (truncated Taylor developments, e.g.). In a complementary way, the bottom-up approach constructs low-degree primitives or polynomial maps from clustered data, which can be interpreted in terms of truncated Taylor developments; when we "regularize" the corresponding distributions away the singular loci, one recovers a local PS-structure.

The exchange between top-down and bottom-up approaches involves to all PeCWA spaces and maps of the Basic Analytical Pipeline) used to describe the succesive fibrations labelled as:

1. *representation* of clusters fibration $\rho : \mathcal{P} \to \mathcal{C}$ on features);
2. the *interaction with itself* in terms of the *transfer map* $\mu : \mathcal{C} \to \mathcal{W}$;
3. the *interaction with the environment* by using the *knowledge map* $\kappa : \mathcal{P} \to \mathcal{A}$).

The jets language [2] provides the support for a unified presentation of successive extensions, which are re-interpreted in terms of kinematic properties for trajectories and constraints, also. To develop this scheme, we need to specify some relations between different mathematical frameworks. To fix ideas, we follow a scheme of increasing complexity:

- A *classical differential approach* consists of starting with a smooth manifold M as models, and maps $f : N \to P$ as relations between models. In this case first order variations are locally represented by the differential $d_{\underline{x}}f$ of a vector map $f \mid_U$ which can be written as a linear map $T_{\underline{x}}U \simeq \mathbb{R}^n \to \mathbb{R}^p \simeq T_{f(\underline{x})}$ between tangent spaces which is of class $r \geq 1$ with $\underline{x} \in U$ is an open set of N. After fixing basis in source and target spaces, the differential map $df \mid_U$ is represented by the Jacobian matrix $J = Jac(f)$ which induces a transformation between vector fields (formal sums of $\partial/\partial x_i$ with functional coeffi1cients). Alternately, if we take its transposed J^T, one has a transformation between covectors or differential forms (given locally by formal sums of dx_i with functional coefficients). Matrix products $J \cdot J^T$ and $J^T \cdot J$ play a fundamental role for feedforward analysis in Robotics [3].
- In a complementary way, we can use a classical *semi-algebraic approach* based on Lie groups of regular transformations leaving invariant "some geometric or kinematic quantity". Along the first module B_{31} (Anchored Robots) of this matter B_2 (Robotics), we have introduced Lie groups G

[2] See the chapter 2 of A_{41} (Basic Differential Topology) for a systematic approach.

[3] Some applicaitons to grasping and handling, have been introduced in the first module, but they can be extended to other issues involving locomotion and any kind of navigation.

and their Lie algebras $\mathfrak{g} := T_eG$. They have been applied to describe different issues concerning to motions at joints and end-effector of a robot, and their common representation in terms of (successive extensions of) the mechanical transfer map $\mu : \mathcal{C} \to \mathcal{W}$ and their local "pseudo-inverses". The basic idea consists of every action on points, induces an action on configurations of such points. Along this chapter we extend this approach by giving a more accurate representation of Kinematics and Dynamics for Mobile Robots. This approach is well known for mechanical issues from the mid nineties (see [Par95], e.g.), but it is seemingly new for Visual Perception issues which will be developed along the second section.

Locally symmetric structures linked to local actions of Lie groups (or their linearization in terms of Lie algebra) give locally homogeneous structures. They simplify the description of isotropic propagation phenomena and/or their transitions towards anisotropic phenomena which appear as state changes or phase transitions, e.g. To fix ideas, we start by considering the simplest case corresponding to locally homogeneous structures.

Lie groups are also manifolds. Thus, one can apply all methods developed following the classical appiriach, including differential and integral calculus, which will be applied to optimization and control issues in Robotics. Both classical and semi-classical approaches provide motivations to develop a differential approach which is largely inspired in well known results of Differential Geometry (matter A_1) and Differential Topology (matter A_4), and their corresponding computational versions developed along first three modules of the matter B_4 (Computational Mechanics of Continuous Media). Some relevant issues are:

- *Estimation:* How estimate low order differential data from sparse or irregularly distributed date?
- *Matching:* How match local kinematic data in a global model?
- *Prediction:* How to propagate evolving models along time?
- *Correction:* How to correct errors involving capture, tracking and propagation?
- *Interaction:* How represent interaction holding along propagation?
- *Control:* How control reactions to preserve some kind of stability along time in regard to possible interactions with other agents?

In Robotics, we are interested in solving these issues starting with the differential $d\mu$ of the *mechanical transfer map* $\mu : \mathcal{C} \to \mathcal{W}$ between configurations $\mathcal{C}$ (or joints) space and the working space $\mathcal{W}$ as support for the dynamics, by one side. On the other hand, we are interested in similar issues for the differential $d\kappa$ of the *knowledge map* $\kappa : \mathcal{P} \to \mathcal{A}$ between the Perception $\mathcal{P}$ and action space $\mathcal{A}$. Both differentials ca be read in terms of a first-order approach to the basic MAD (Main Analytic diagram):

$$\begin{array}{ccc} \mathcal{P} & \rightarrow & \mathcal{A} \\ \downarrow & & \downarrow \\ \mathcal{C} & \rightarrow & \mathcal{W} \end{array}$$

where each (horizontal or vertical) arrow is a fibration between (semi-)analytic spaces. A simplified version of feedforward processes can be read by following a counterclockwise path between the four subspaces:

1. Data arising from external and internal sensors are processed and analyzed in terms of a a battery of filters, and stored in the Perception space $\mathcal{P}$.
2. Low-level recognition tools allow to identify the most probable patterns by detecting local versus global stable configurations which are stored in the Configuration space $\mathcal{C}$.
3. Stable patterns are inserted in some coarse-to-fine reconstruction of an eventually mobile representation scene in the Working space $\mathcal{W}$.
4. The resulting information is lifted to the "best choice" between possible actions to be developed in the Action space $\mathcal{A}$.
5. The interaction with an evolving environment $\mathcal{E}$ is performed in terms of the locally trivial fibration $\mathcal{P} \rightarrow \mathcal{A}$.

The existence of lifting maps is warranted by the locally trivial structure of all fibrations appearing in the MAD. Furthermore, this construction is compatible not only with the behavior of the set $\mathcal{O}_{\mathcal{X}}$ (in fact a sheaf of local rings, in mathematical terms) of "regular functions" defined on each support $\mathcal{X}$, but with any kind of superimposed fields $\mathcal{F}$ linked to differential structures (in mathematical terms a $\mathcal{O}_{\mathcal{X}}$-module).

In principle, $\mathcal{O}_{\mathcal{X}}$ denotes the set of C^r-regular functions for some $0 \leq r \leq \infty$ or $r = \omega$ (analytic case). They provide the support for multilinear superimposed structures whose Ssmplest cases correspond to the module $\Theta_{\mathcal{X}}$ of vector fields (locally given by partial derivatives) or the module $\Omega^1_{\mathcal{X}}$ of covector fields (locally given by differentials). More generally, to represent "aggregated quantities" one can consider any kind of tensors of type (r, s) (given by multilinear maps on r copies of cotangent and s copies of tangent bundles) for a simultaneous representation of constraints and possible paths or trajectories. Their estimation is performed in terms of tensor voting procedures.

Everything would be easier if all the above spaces $\mathcal{X}$ would be smooth manifolds. Unfortunately, due to internal and external constraints, neither of these spaces are smooth manifolds, but analytic spaces $(\mathcal{X}, \mathcal{O}_{\mathcal{X}})$. In other words $\mathcal{X}$ can have singularities, and functions belonging to $\mathcal{O}_{\mathcal{X}}$ can display "pathological behaviors" (changes of state, phase transitions, e.g.). Nevertheless the presence of singularities, one can "stratify" the support X by decomposing in a disjoint union of smooth "strata" verifying good incidence cionditoins at their boundaries. In other words, one can associate PS-models (PS: Piecewise Smooth) where manifolds strategies can be applied from the local viewpoint.

In the same way as for the smooth case, first order approaches of maps $f : \mathcal{X} \to \mathcal{Y}$ between (semi-)analytic spaces are loclaly given by Jacobian matrices providing a support for corresponding kinematics. A first issue is how to "propagate" and match solutions obtained for minimal vs redundant amount of data to obtain meaningful configurations, which can give a coherent representation of en evolving working space which can be continuously lifted (avoding switching procedures) to perform an action on a space-time evolving scene model. Matching local models in a more global model is easier if one has "enough symmetries"for propagating local solutions in a coherent way. The next paragraph presents some basic related ideas in the framework of locally symmetric spaces.

Before starting one must be aware that in all cases there appear different kind of errors; it is necessary not only minimize from the beginning, but also along propagation phenomena involving anticipation, prediction, simulation and decision making. In particular, locally symmetric structures used for propagation can involve to discrete, continuous and infinitesimal symmetries, depending on the framework, and the functionals we are looking for.

- In the *classical approach* very often one takes a weighted sum of L^1-distance)to remove outliers) and L^2-distance for a finer adjustment after removing outliers; this method is inspired by the use of a total energy functional given by the sum of a potential and a kinetic energy term.[4]
- In the *semi-classical approach* errors appear also in the estimation of elements belonging to Lie groups G (non-linear manifolds) and their corresponding Lie algebras $\mathfrak{g} := T_eG$ (vector spaces with a Lie bracket corresponding to matrix product). In this case, to begin with it is convenient to take weighted averaged mobile means for near localizations before propagating them to minimize registration errors (see e.g. [Gov04][5], e.g.)

To difference with traditional geometric approaches, in applied fields such as Robotics and Computer Vision, data estimation and correction of errors play a fundamental role for prediction, tracking and validation. These issues involve to every kind of transformations, including actions on intermediate entities. So, transformations of Lie groups G are estimated in terms of their linear approaches given by their Lie algebras $\mathfrak{g} := T_eG$, and lifted to G via the exponential map $exp : \mathfrak{g} \to G$. Similarly, errors are corrected by taking mobile Moving Averages.

0.1.1. Towards a locally symmetric formulation

The above remarks, suggest to analyze the situation at a point $x \in X$, and extend to small neighborhood U of $x \in X$ by using the action $\alpha : G \times X \to X$ of

[4] In fact the Newtonian total energy is a functional defined on the total space TM of the tangent bundle τ_M?'(TM, π, M) taking values on $\mathbb{R}$, hence it can be considered as a 1-form

[5] V. M. Govindu: "Lie-algebraic averaging for globally consistent motion estimation", in *Proc of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, 2004, vol. 1. IEEE, 2004

a group G on a space X. This approach is naturally extended to scalar functions $f : X \to \mathbb{R}$ defined on X or, even to more general fields. To warrant stability and robustness properties of the model, we are interested in characterizing any kind of G-invariant fields on X which allow to propagate local information and "fill out" missing data.

In more dynamical terms, bifurcation phenomena for possible solutions are translated to equivariant or G-bifurcation phenomena between G-spaces X, Y for different groups $G = (G_X.G_Y)$ acting on source X and target space Y of a map $f : X \to Y$. In more advanced models, G-bifurcations corresponding to different solutions can be propagated along different G-orbits according to adjacency hierarchies between subgroups H of G acting also on different "strata" for maps $f : X \to Y$ between locally symmetric G-spaces.

- If actions on source and targets spaces are decoupled, then one can take the direct product $G_1 \times G_2$ of groups acting on source and target spaces of f. In this case, one has the $A := R \times L$ or right-left action. A typical example is given by decoupled strategies involviong Structure and Motion (SfM vs MfS), e.g.
- If actions are coupled between them, then one must take some kind of mixed group (semi-direct product $G_1 \ltimes G_2$, e.g.) acting on the graph Γ_f of f. Furthermore, well known examples of Euclidan and (Special) Affine Groups, the most relevant for preserving relations between maps (and their possible degenerations at critical loci) are given by contact group K, which appears in different ways along this matter.

From a theoretical viewpoint, problems start with an explicit description of each one of the above spaces in Robotics. Some of them ($\mathcal{P}$ and $\mathcal{A}$), are nor even parameterizable becuase they are described in functional terms (signals and commands) corresponding to spaces of infinite dimension. From the Functional Analysis viewpoint, one can restrict to Hilbert spaces [6]. From the alegrabci viewpoint, one can use infinite-dimensional representations of $\mathfrak{s}\ell(2;\mathbb{C})$ to describe their locally symmetric structure as in Harmonic Analysis. From the topological viewpoint, one can consider generic deformations of operators or paraticular known solutions for these operators.

Anyway, all of them display some type of (infinitesimal, local, topological) symmetries which can involve their local descriptions of configurations, their kinematics or their dynamics. In particular, for a perspective map one has an initial configuration given by a collection of points and lines fulfilling incidence conditions, e.g; this configurtion evolves according to trajectories (vector fields) for points and constraints (differential forms) linked to allowed motions; hence, it can be represented as a variable tensor. Let us remember that symmetries appearing at a mechanical level are not necessarily translated to the adjacent mechanical levels such it appears in underactuated mechanisms, e.g.

[6] Intuitively, a Hilbert space can be considered as the natural extension of Riemannian structures (M, ds^2) on manifolds M.

Hence, one must be careful about lifting and descending processes for symmetries and their hierarchies[7]. Their locally symmetric character allows a "translation" of punctual pheonomena to a small neighborhood by means of the use of a (metric vs affine) "connection" representing simple propagation models. So, we obtain "generalized symmetries" having in account the geometry of the support X extending classical actions $G_X \times X \to X$ of a structural group G_X on a topological space X.

More specifically, in the first module B_{31} (Anchored Robots) we have already seen the convenience of defining $\mathcal{C}$ and $\mathcal{W}$ in terms of (subvarieties of products of) *Lie groups* G (i.e. manifolds with group structure). Their differential version is given by the corresponding Lie algebras $\mathfrak{g} := T_e G$ corresponding to the tangent space to G at the neutral element $e \in G$ (the identity matrix for groups of matrices) [8].

The Lie based approach is not new, and it has been increasingly used from the early nineties. Roughly speaking, it can be justified by the own locally symmetric nature of transformations to be performed, by the non-linear nature of most tasks which require locally symmetric manifolds M or by the reversible nature of control to be performed on such manifolds. In other words, Lie algebras allow recover a locally symmetric structure which can be observed in most tasks, larger facilities for reversibility and also for tracking and prediction by using simple propagation rules given by local symmetries. Beyond regular behavior, the use of nilpotent Lie algebras allows to identify possible degenerations (critical behaviors), relate adjacent but qualitively different behaviors and control dissipation phenomena.

Thus, a *novelty of this chapter* is the use of symmetries for different kinds of (scalar, co-vector, tensor) fields appearing in regard to environmental undertanding, interaction and control. We shall show how all of them can be managed in terms of different kinds of superimposed differential structures (vector bundles, principal bundles, fibrations, sheaves) to ease all kinds of (mechanical, information, learning) transfer between different embedded architectures in regard to the Automatic Navigation of Mobile Robots.

To ilustratge this viewpoint, let us remember that along the first module we have used essentially the Lie formalism to describe rigid motions in terms of some basic properties of (the product of a finite number of copies of) the Special Orthogonal Group $SO(n) := SL(n) \cap O(n)$, where $SL(n)$ is the Special Linear Group (preserving oriented volumes, very useful for conservative flows) and $O(n)$ is the Orthogonal Group (preserving orthogonality properties). Their interest for Robotics consistis of the Euclidean Group $SE(3)$ (preserving distances in the ordinary space) given as the semidirect product $SO(n) \ltimes \mathbb{R}^n$ of $SO(n)$ represents rigid motions for the palne and space; in other words corresponds to the joint

[7] Here, we are considering Geometric, Kinematic and Dynamic "levels" not only for Mechanics, but also for PAC, as in other modules of this matter.

[8] The extension of Lie's apoproach to Perception and Action Spaces can be performed in terms of Lie groups, but details are more cumbersome; see chapter 6 of this module for more details.

action of rotations group by the group of translations; this interpretation is only valid for $n \leq 3$. In this chapter we extend this approach to other Lie groups which are meaningful for Automatic Navigation in regard to

- the Special Affine Group $SA(n) = SL(n) \times \mathbb{R}^n$, preserving flows up to translations, and consequently to relate image flow with scene flow; and
- the Symplectic Group $Sp(2n; \mathbb{R})$ preserving ideal motion's equations as "toy model" for more advanced interaction patterns (in presence of "external forces").

Unfortunately, the classical management of elements of $SO(n)$ in terms of trigonometric functions is a source of increasing errors and "artificial singularities". Luckily, quaternions allows to avoid these inconvenients, but the cost is the introduciton of an ambiguity because the set $\mathbb{S}^3$ of unit quaternions (representing spatial rotations) displays an ambiguity linked to the antipodal map. This issue is very important in control issues, because for large or abrupt motions in UAVs, convergence towards a quaternion must be fulfilled in the same copy of $\mathbb{S}^3$ (it is not allowed a jump to the opposite and equivalent copy) [9].

0.1.2. An adaptation to navigation issues

The situation in module B_{32} (Automatic Navigation) is a little bit more complex than in module B_{31} (Anchored robots) in which concerns to much more complex evolving interactions with the environment and other mobile agents. For an effective navigation, we must display forward and feedback mechanisms able of relating the *image and range* information (deformed by perspective effects, noise and/or reaching of range sensors) with the information relative to rigid motions to be performed by a mobile platform.

To fix ideas, let us consider an "example" corresponding to the capture of a scene with a video camera. Segments alignment in perspective lines gives a perspective map which is continuously updated according to projective models for typical man-made scenes. To simplify, one supposes central camera models (like pinhole), giving eventually skewed projection. In this case,

1. *Features* (corners or intensity maxima, e.g.) are generated from the analysis of sampled images in the video sequence; next, they are identified, and evaluated (acceptation vs rejection) for their use in clusters;
2. mobile *configurations* are given by semi-dense clouds of points to be tracked, and grouped in clusters along consecutive sampled frames (by using some variant of SLAM procedures).
3. Clusters of accepted features are re-projected on quasi-homogeneous radiometric regions, which are superimposed to a perspective map as simplified model for the *Working space* $\mathcal{W}$.

[9] See Chapter 5 for related Control issues

4. Preliminary steps for effective navigation in the *Action space* $\mathcal{A}$ are focused towards the identification of the Free Collision space, and the near-optimal selection of the action to be undertaken according a system of logical rules.

(*) In this case, the apparent deformation of the scene (appearing in the perspective map) and its time evolution can be locally described in terms of the Affine Group $G_{\mathbb{A}^2}$ defined as the semidirect product $GL(2) \ltimes \mathbb{R}^2$ of the General Linear Group $GL(2)$ and the group of translations $\mathbb{R}^2$ in the plane. If we take regular transformations up to scale (the affine group preserves segment ratios up to scale), the main "normalized" ingredient is given by the Special Linear Group $SL(2) := \{A \in GL(2) \mid det(A) = 1\}$.

Hence, these transformations preserve ideally the volume of image flow, and consequently, allow its propagation by using $\mathfrak{s}\ell(2) \ltimes \mathbb{R}^2$ along a perspective map by means an adequate rescaling depending on depth. [10]

How can use this information for the design of rigid motions to be performed by the mobile platform? If we restrict ourselves to $SO(3)$ (rotations in the ordinary space), one can make it by the following simple remark: The group $SU(2) := SL(2;\mathbb{C}) \cap U(2)$ (where $U(2)$ is the unitary group), is a double covering of both of them [11]. Therefore, the idea is very simple:

1. Lift the visual information from $SL(2)$ to $SU(2)$, and descend to $SO(3)$, next. Since $SU(2)$ is a *complex* Lie group one must complexify each one of the above special lineal and special orthogonal groups. Remark that all of them are three-dimensional Lie groups; hence their complexification gives also three-dimensional complex groups. Unfortunately, their geometry is a little bit complicated. Thus, it is preferable to linearize them by choosing appropriate basis for their Lie algebras. In this way, one can apply SVD or PCA methods on $\mathfrak{g}$.

2. Inversely, one can simulate the effect of a rigid transformation belonging to $SE(3) = SO(3) \ltimes \mathbb{R}^3$ (useful for training, e.g.). It suffices lift the element of $SO(3)$ to $SU(2)$, and descend to $SL(2)$, by taking in account the effect of a central projection on the image plane . In this way, one has an "anticipated evolution" of the scene according to the planned motion to be compared with perspective map obtained at the following sampled frame.

So, our strategy consists of, instead of working with Lie groups G (non-linear manifolds), take their linear approach given by the corresponding Lie algebras $\mathfrak{g} = T_eG$, and recover G by using the exponential map $\exp : \mathfrak{g} \to G$ (local diffeomorphism). Last ones are nothing else than vector spaces with an inner product (given by the Lie bracket of matrices) verifying the anti-commutativity

[10] The situation for UAV is a little bit more complex, because furthermore variable depth maps, one must manage variable height maps. In this case, one must consider the affine group of the ordinary affine space $\mathbb{A}^3$, and $SL(3)$ instead of $SL(2)$.

[11] In fact $SU(2)$ the "universal" double covering of both groups, in such way that every map relating both of them "factorizes" necessarily thorugh $SU(2)$

property and the Jacobi's identity. In the above cases, they are 3-dimensional vector spaces.

(*) As $\mathfrak{sl}(2)$ and $\mathfrak{so}(3)$ are 3-dimensional *real* vector spaces, and $\mathfrak{su}(2)$ is a 3-dimensional *complex* vector space, one must complexify the two former ones, giving also three-dimensional Lie algebras $\mathfrak{g}_{\mathbb{C}} := \mathfrak{d} \otimes_{\mathbb{R}} \mathbb{C}$, or in more intuitive term, now coefficients are complex instead of real numbers. In a nutshell, the *general strategy* consists of the following steps:

1. Estimate infinitesimal motion at the corresponding Lie algebra $\mathfrak{g}$ by using SVD (or its corresponding statistics version PCA).
2. Correct errors by taking near k-tuples of points and apply mobile Moving Averages.
3. Translate the information to the group G by using $exp : \mathfrak{g} \to G$ is a local diffemorphism in a small neighborhood of the neutral element.
4. Propagate by translating along G according to the reuslt $\tau_G = \mathfrak{g} \times G$ (every Lie group is parallelizable, i.e, its tangent bundle is topologically trivial).

(*) Splitting properties of Lie groups in terms of Lie algebras representations is the key for the reduction to low dimension objects (groups vs algebras, and their corresponding homomorphisms). In this way, one reduces the problem of updating and tracking information along Automatic Navigation to simple computations with small matrices appearing in natural decompositions of Lie algebras [12] in terms of "basic pieces" given by representations of $\mathfrak{sl}(2)$.

Nevertheless, it is necessary to have very efficient methods to interpret these computations, not only in terms of the support (given by perspective maps and semantic maps), but to their space-time evolution according to propagation actions. Their local description in terms of infinitesimal symmetries corresponding to Lie algebras, provideds a general context for a joint treatment. To fix ideas, we restrict ourselves to propagation based on perspective maps, initially, where $\mathfrak{sl}(2)$ is the basic piece.

For any perspective map, the action induced by Lie algebras is estimated by using standard SVD techniques for decomposing each infinitesimal perception of a motion $X \in \mathfrak{g}$ in a weighted combination $\sum_{i=1}^{3} \lambda_i X_i$ (where λ_i are the eigenvalues of SVD) in terms of the corresponding standard basis $\{X_i, X_2, X_3\}$ for $\mathfrak{g}$ [13].

The action of the group $SL(2)$ in the plane is translated to linear objects (pencils of perspective lines) by taking the transposed matrix. Their translation to G-invariant functionals defined on curved objects is more involved and requires some basic elements of Geometric Invariant Theory, which are difficult

[12] For Lie algebras, one must replace diagonal decompositions by semisimple Lie algebras, and finite sums of vevcctor spaces by infinite sums of representations of $\mathfrak{sl}(2)$. The first general results are due to E.Cartan; for an introduction see the chapter 0 of [Gri78].

[13] If one desires, the exponential $exp(tX)$ gives the desired element of the group G

to implement. Thus, we shall restrict ourselves to basic properties involving lines and conics to represent second order momenta. Conic curves can be easily estimated by using central momenta of order two, and their degenerations are well known.

The above ideas show a general strategy for translating visual information to motor information for an effective navigation in terms of rigid transformation to be accomplished by actuators. Furthermore, one can add range information in terms of a sparse depth map, which is superimposed to the semantic segmentation (by means RGB-D cameras, laser or Lidar, e.g.).

Inversely, motion planning in terms of rigid transformations can be lifted to $SU(2)$; their descent to $SL(2)$ provides a method to propagate the information contained in the current perspective map in another extended perspective map to be validated (or not) in the next iteration. In this way, one obtains a Lie-based closed loop not only for information update, but for anticipating the following projected scene, also.

In practice, visual information at each instant is managed in terms of two layers consisting of a perspective map for geometric information, and a semantic segmentation for the scene included in the FoV (Field of View) of the camera. Both of them are related by depth (and height for UAV) map. More recently, semi-dense clouds $\mathcal{N}_p(t)$ of meaningful points, incorporating scalar information (depth, height), have been extended to disperse cloud $\mathcal{N}_s(t)$ of segments (grouped in the correspoinding Hough space, e.g.).

Advanced models to update, tracking and prediction of point clouds are performed in terms of some extension of Kalman filters [14] applied to mobile perspective maps. For lower-level approaches, similar tasks for radiometric information are performed in terms of SLAM (Simultaneous Localization and Mapping) or particles filter (also called Monte-Carlo sequences). Relations between the update of continuous (perspective maps) and discrete models (mobile point clouds) are not still well understood. If we superimpose meshes, they involve to different kinds of image and scene flows which ideally are ideally preserved by $SL(2)$ and $SL(3)$, respectively.

0.1.3. Space-time evolving scenes

To start with, one supposes initially an underlying smooth structure $M(t)$ or a PL-approach given by a mesh $\mathcal{M}(t)$ for an evolving scene $\mathcal{S}(t)$. Inside such scene there are a finite collection of evolving PS-objects $B_\alpha(t)$ contained in the scene $\mathcal{S}(t)$ which provide an underlying space-time model. Their space-time evolution will be denoted as $2D+1d$ for *image flow* ϕ_t in video sequences, and as a $3D+1d$ for *scene flow* Φ_t for evolving volumetric representations. In this chapter we extend usual functions-based approaches to co-vector or more generally tensor fields to stimate “mixed quantities” (see §1,4,1 for an introduction).

[14] See chapter 6 of B_{33} (Robot Kinematics) for details.

In Cartesian coordinates, flows are denoted as $\phi(x, y; t)$ and $\Phi(X, Y, Z; t)$, respectively, expressing surface or volumetric variations. Obviously, the planar flow is a projection $\pi_{\mathbf{C}_t}(\Phi_t) = \phi_t$ of the scene flow for each mobile position $\mathbf{C}_t$ of the camera's center. Lifting a planar to a volumetric flow is an "ill-posed problem" in the Hadamard sense. One must add constraints linked to the scene and/or motion characteristics as it occurs for Optical Flow. An advantage of our approach consists of the incorporation of information about the scene and its projection, differential invariants for a volumetric and planar flows (corresponding to an adapatation of Gauss and Stokes theorems).

Above we have seen how *Differential and Integral Methods* can be formulated in terms of different frameworks concerning the *space-time evolution* of evolving scene and obstacles. One can choose a smooth, a PL or even a discrete support; depending on the choice, we will use ordinary (co-)vector fields, their PL-approach or their discrete versions given by pde (partial difference e quations). In all cases the basic notion is that of vector field for small displacements or its dual given by co-vector fields (called differential forms in the smooth case) for constraints.

To represent different agents with their own kinematic characteristics, one uses a finite number of *vector fields* $\mathbf{X}_i$ to describe motion characteristics. Similarly, one uses a finite number of *covector fields* ω_j to evaluate the behaviour of vector fields and to represent evolving constraints in the Cartesian space. From a formal viewpoint, both of them can be considered in a simultaneous way in terms of tensor fields which are nothing else than tensor products of a finite number of vector and covector fields.

This reasoning scheme can be applied to *control* issues and *optimization strategies* for decisions making which are developed with more detail in Chapter 5. The simplest vector fields $\mathbf{X}_i$ are locally given by like-gradient potential fields ∇f_i which provide a coarse approach for solving general motion equations. In our context, they are given by algebraic expressions in terms of vector fields or their dual expressions in terms of covector fields (the same as differential forms for smooth manifolds M).

(*) More generally, to include higher PDE (or their PdE versions), they can be written as polynomial expression in the corresponding "jets space" [15]. A systematic study of their solutions will be developed in modules B_{23} (Kinematics) and B_{24} (Dynamics). Thus, along this chapter, we restrict ourselves to simpler issues concerning to the behaviour of (generic perturbations of) scalar fields or like-gradient or potential (co)vector fields.

Two crucial aspects concern to the *estimation and prediction of trajectories* for a low number of "meaningful points (control points, beacons, markers), and AI strategies for *multimedia contents learning* in video sequences [16]

[15] Intuitively, a k-jet $j^k f$ of a function f is (an equivalence class of) the truncated Taylor development at order k of f.

[16] For a recent survey on Statistical Learning and Multimedia Learning see the chapters 42, 44 and 56 of [Pha23].

A common approach consists of a *space-time decoupling*. In other words:

1. extract general characteristics of the mobile scene going from background segmentation till more complete semantic segmentation; next
2. perform a content analysis for isolated images, depending on the processing and analysis capability for such images.

This strategy ignores "internal links" (entrelacement) between the "most meaningful" objects which are found in the video sequence. A first formalization of this idea was made in terms of space-time surfaces (Faugeras and Papadopoulos, ECCV'96); see last section of this chapter. These surfaces are understood as "slices" of the volumetric flow along time. Their estimation requires additional elements to identify tangential and normal components along these surfaces.

Motion estimation is directly related to the discrete vs continuous information arising from available sensors, including fusion of information in $\mathcal{P}$, reprojection on configurations in $\mathcal{C}$, and reinterpretation of clusters in terms of "objects" in $\mathcal{W}$. Thus, a key issue is to provide smooth approaches for succesive fibrations between PeCWA spaces of the BAP, by starting with discrete information provided by sensors. There are several strategies to generate related PL-models from discrete data. We use alignment between signals, configurations and objects which are finally embedded in perspective maps, with the corresponding automatic generation of generic meshes (away discontinuities of depth map).

PS-models simplify PL-models (drastic reduction of cells). However, they require an efficient implementation of regularization models. In addition, smothing strategies to pass from PL-models to PS-models can require advanced Optimization techniques for non-convex models. The simplest smoothing strategies use some variants of statistical parametrics (Bernouilli, Gauss, Poisson) linked to inputs. More advanced models use regularization techniques (Tikhonov, e.g.) which can be seen in the module B_{23} of the matter B_2 (Computer Vision).

From a statistical viewpoint, classical estimation models for time evolving phenomena (corresponding to ODE in the smooth case) are given by time series with ARMA (AutoRegressive Moving Average) as paradigm. This approach is extended to time multiseries, where each agent can have its own time (one requires a common time scale for the management of delays).

In a complementary way, the estimation of space-time distributions $\mathcal{D}$ (corresponding to PDE in the smooth case) is performed in terms of multi-correlation models. The existence of a "dictionary" between basic notions provides bridges for connecting top-down and bottom-up approaches. A non-trivial example is given by the analogy between curvature and covariance matrices, e.g., involving second order variation rates of "quantities".

Observed phenomena are not strictly random, nor even at beginning in some cases one starts with this kind of models. Data estimation in an evolving scene and tracking mobile objects under uncertainty or incomplete information must incorporate self-adaptive optimization procedures to improve coarse-to-fine PS models.

In more advanced stages, one relaxes smoothness conditions about varieties and morphisms by more realistic hypotheses relative to objects and their transformations. Relaxation procedures are convenient to incorporate Piecewise Linear (PL) or even discrete models, including events or experimentally observed discontinuities involving the information treatment in terms of fields.

In view of uncertainty, often we use Markov Random Fields (MRF) as a discrete analogue of ordinary vector fields. The use of pde (partial difference equations) on Graphs representing symbolically evolving regions in Semantic Maps simplifies the global treatment of underlying information.

0.1.4. AI for automatic navigation (*)

From an experimental viewpoint, one must develop an interplay between multivariate time series (for evolving data), and stochastic perturbations of ideal submanifolds N in the Phase Space P (Poincaré) corresponding to the total space TM or T^*M of the co-tangent bundle of an ideal manifold M (space-time surface, e.g.). Hence, the central problem of this paragraph is to sketch the main *learning strategies for submanifolds* in P. We start with a top-down approach..
ç

The simplest models for evolving scenes are given by PS-manifolds M, and PS-maps $f : N \to P$ between them (PS: Piecewise smooth). They provide the framework for Deep Learning methods in AI, with *Learning Manifolds* as a central topic for applications in different areas going from Recognition of static objects and their motion characteristics in the mobile case. The Phase space P (introduced by Poincaré) is given by the total space TM of the tangent bundle $\tau_M = (TM, \pi, M, \mathbb{R}^m)$ or its dual T^*M.

The Phase space P it has a natural structure as manifold induced via π^{-1} (by using the local triviality of the tangent bundle(of the smooth structure of M. In addition, P has a natural symplectic structure. Thus, recognition of smooth characteristics of a motion in the Phase space is formulated as Learning Manifolds in the Phase space P, which is decoupled in terms of learning vector and covector fields.

From a theoretical viewpoint, one starts in a PS framework, and one enlarges in a two-fold way, by incorporating PL-objects and maps (for connecting discrete data with continuous models), and extending the PS to the Semi-Analytic framework to i ncorporate "events" which are modelled as singularities of varieties X or, more generally, maps $f : X \to Y$ between varieties.

For linear models (given by paraperspective or weak perspective in W, e.g.), motion characteristics are subject to r constraints $\mathbf{w}_i$ (evolving hyperplanes represented by linear forms $T_xM \to \mathbb{R}$ on the tangent space), and described in terms of s vector fields $\mathbf{v}_j$ (whose integral curves give ideal trajectories $\gamma_i(\underline{q})$ for i control points. Each one of them can have a variable weight w_{ij} which is represented by a scalar field. Constraints on W are naturall lifted to constraints on C and P in the PeCWA pipeline.

Linear combination of weighted products of scalar, vector and covector fields gives a tensor of type (r, s) for the motion, which is called the structural tensor for the motion. Tensors are estimated by using *Tensor Voting* procedures. Their evolution in the space-time is described in terms of Tensor Flows. Extensions of classical tensor Voting procedures can be read in the chapter 39 of [Pha23]. From the AI viewpoint, the implementation of *TensorFlow* under PyTorch provides the first integrated approach for the estimation of multiple motions under weighted constraints [Goo16].

The joint management of multiple trajectories γ_j for $1 \leq j \leq s$ is performed in terms of *distributions* $\mathcal{D}$ of vector fields $\mathbf{v}_j$. The joint management of multiple linear constraints H_i for $1 \leq i \leq r$ is performed in terms of a *differential system* $\mathcal{S}$ of covectors given locally by linear forms $\mathbf{w}_i : T_xM \to \mathbb{R}$ on the tangent space. Initially, we suppose that distributions and systems are integrable.

If we adopt a *bottom-up* approach, the simplest motion characteristics are given by the first and second order finite differences of localization (e.g. position and orientation) for a finite number of control points which are tracked along the motion B_{23}. At the lowest level, one can use time series modelling for the coarsest motion characteristics, with multivariate ARMA (Auto Regresssive and Moving Average) patterns for linear vs angular speed and acceleration. Typical statistical models for motion tracking are based on some variant of the Kalman filters. B_{336}

Hence, AI-based modelling in the DNN framework, requires the use of (at lrast) three consecutive layers for each module, able of extracting information from variable weights at "small units". A typical choice in the classical case (before the massive use of AI), is given by radiometric superpixels corresponding to quasi-homogeneous regions in consecutively sampled images of a video sequence, e.g. Currently, one can work at pixel level on supercomputers. However, an embarked system in an automatic car has a more limited computation capability. Thus, even for off-line training one can use a supercomputer, it is convenient preserve the methodology linked to "older" solutions, in order to incorporate "singular events" which have not been included in the learning stages.

0.2. Outline of the chapter B323

As usual, furthermore this introduction and a fifth section for recapitulation, materials are organized in four sections to be given along one month (one per week). They contain a list of exercises for self-verification of understanding of materials.. Materials are organized in the following sections:

1. *Differential Geometry for Mobile Robots*, where geometric foundations are developed according to ideal PS objects and maps.

2. *Motion estimation*, where one develops a Lagrangian approach for connecting with the preservation of motion characteristics (momenta, motion's equations, flow volume).

3. *Tensor flows for visuo-motor integration*, where one develops a joint treatment of trajectories and weighted constraints, with the corresponding evolving hierarchies extending classical Flag Manifolds.

4. *A differential approach to the PeCWA* as a natural extension of the PAC (Perception-Action Cycle), where one shows the utility of decoupling with Configurations and Working space in the extended PeCWA pipeline to improve the performance of Navigation systems

0.2.1. Some methodological issues

Along this chapter we prior a *top-down* methodology, where theoretical models are based on the PS manifolds for the Phase space P (Poincaré). The Phase space P can be modelled in terms of the total space TM of the tangent bundle τ_M or the total space T^*M of the cotangent bundle τ_M^*, whose sections are locally given by vectors (for trajectories) and covectors (for constraints) in the linear case.

If we adopot a "deterministic" approach, motion's equations are obtained in the Newton context, as the derivative of an energy functional. Hamilton-Jacobi equations give an expression for N control points. The incorporation of internal and external forces, and the introduction of torques "to compensate" undesirable phenomena (linked to instability, e..g) gives matrix expressions for the Robot Dynamics which will be exploited in the module B_{24}. Thus, along this cahpter we adopt a simpler approach whcih is based on the preservation of motion's equations.

The local parametrization of the Phase space P and the total space TP of its tangent bundle τ_P is described in terms of the *Lagrangian generalized coordinates* $(\underline{q}, \underline{p}, \underline{r})$ fulfilling the *contact structural constraints* (relations in the cotangent bundle of P) given by

$$\underline{p}dt - d\underline{q} \quad \text{and} \quad \underline{r}dt - d\underline{p}$$

In absence of external forces, ideal *motion's equations* appearing in the differential approach (Hamilton-jacobi) and integral approach(Euler-Lagrange) corresponding to the minimization of the total energy functional, are expressed in terms of the preservation of the two-form on P given by the *symplectic form* ω whose canonical form is locally expressed as $\omega \mid_U = \sum_i dq_i \wedge dp_i$. The symplectic group $Sp(n)$ acting on tge Phase space P is characterized by the preservation of ω. The symplectic 2-form ω is related with the *contact form* α given locally by $\sum_i p_i dq_i$ by $\omega = -d\alpha$.

(*) Lagrangian and Legendrian (resp. Legendrian) varieties are the integral solutions of maximal dimension in P (or its contactification) corresponding to the motion's equations linked to the preservation of the symplectic (resp. contact) form. Their projection on the base space given by the PS-manifold M gives Lagrangian (resp. Legendrian) waves which can be visualized as eventually singular "leaves" of a foliation on M. Classification of the corresponding singularities has been performed in the module A_{44} (Singular map-germs) of the matter A_4 (Differential Topology), and provide ideal models for "events" in space-time evolving phenomena. In this chapter, we adopt a more basic approach by restricting to the regular case or to "generic singularities" (codimension one).

Vector and covector fields acting on the base manifold M or the Phase space P (modelled as "sections' of the corresponding tangent and cotangent bundles) can be considered as two layers involving evolving trajectories and constraints. Their overlapping is carried out in the corresponding bigraded piece of the tensor algebra.

Usual regular transformations are locally described in terms of the Jacobian matrix for vector fields (or its transposed for covector fields) for each map relating two "states" for each analytical space $\mathcal{X} = (X, \mathcal{O}_X)$ or for each map $F : \mathcal{X} \to \mathcal{Y}$ between two analytic spaces of the PeCWA pipeline.

Less attention is paie in the application of TensorFlows in DNN to non-regular operations in the tensor algebra consisting of contracting or expanding indexes. They can correspond to the decreasing or increasing of the number of agents vs constraints operating in the evolving space-time representation of the environment (including the removal of a layer for some kind of vectors or covectors, e.g.).

Along the third section, we develop this extension (well known from the Classical Differential Geometry of the Italian school at the end of the 19th century), by incorporating some basic combinatorial tricks for their combinatorial treatment. So, furthermore regular transformations of tensors "travelling" in an abstract space of tensors, we have regular operations, which allow contract and expand information in a similar way to submersions and immersions in Differential Topology. Some "immediate applications" are linked to automatic recognition and generation of new video contents in AI.

0.2.2. Fields and flows for navigation

Local Differential Analysis allows compare and analyse events w.r.t. structural elements of evolving continuous scenes $\mathcal{S}(t)$. The same strategy can be applied to discrete configurations of elements contained in objects $B_\beta(t)$ varying along time. In both cases, their space-time evolution is described in terms of scalar, (co-)vector and tensor fields. Each one of them has its own flow which is characterized as the set of evolving solutions in some representation of the space-time:

- The gradient flow ∇f of $f : M \to \mathbb{R}$ relates two level surfaces for f.

It can be applied to depth, height or attitude, image intensity, energy function, Hamiltonian or Lagrangian functionals in Analytical Mechanics, probablity density functions (pdf), e.g.

- Flows $\phi(\xi;t)$ of a field ξ is locally given by a pack of solutions of the ODE representing locally ξ; flow of a covector field is given by a uniparametric family of constraints (dual of a vector field) evolving in a simultaneous way. They are applied to non-gradient fields (dense set), work performed by a system, contact constraints, optimization issues, between others.
- Tensor flows correspond to the formal product of a finite number of scalar, vector and covector fields, representing "evolving quantities" whcih are varying in a simultaneous way. Typical examples are given by structure tensors for 3D Reconstruction and Motion, simulation of complex articulated mechanisms; evolving texture maps and reflectance maps in radiometric analysis, can be also considered as tensor fields.

The above description follows an increasing order of difficulty, and maps between fields allow to relate different "distributions" for evolving quantities. Lie derivative is an extension of directional derivative which can be defined for any tensor, leaving invariant its type. Furthermore regular transformations between tensors of the same type, one has contraction and expansion operators which allow to relate planar and volumetric objects along time.

A typical *example* is given by trajectories of k agents (persons, cars, e.g.) as integral curves of distributions $\mathcal{D}$ of k vector fields X_i, and apparent displacements of structural elements (walls or ground floor, e.g.) as integral surfaces of systems $\mathcal{S}$ of differential forms ω_j in an evolving scene given at each point by a manifold $M(t)$. When the behavior of two or more agents is similar, one can replace them by once a agent (acting as the leader,e.g.) and increase the flow size; this can be interpreted as a contraction of the associated volume form (a differential form) along several fields.

The comparison between static objects (manifolds $M(t)$, e.g.) is performed in terms of maps $f : N \to P$, where N (resp. P) is the source (resp. target) space. Differential Geometry provides a support for Kinematics on the total space of tangent and cotangent bundles. In more down-to-earth terms, the first order approach to f is described in terms of linear maps between (co)tangent spaces given by the differential map $d_x f : T_x N \to T_{f(x)} P$ at each point $x \in N$, and locally represented by the Jacobian matrix; it allows to express how vector fields X_i and differential forms ω_j are transformed between both spaces.

In this way one obtains, not only a linear approach for objects and maps, but also for Kinematics. So, one can identify flows in terms of (forward or inverse) images of flows. Anyway, differential tools allow identify not only regular behaviors, but "sudden transitions" between them. Last ones are described in terms of Kinematic Singularities for the system [17]. This viewpoint has been already introduced in the module B_{21} in regard to Kinematic Singularities of anchored

[17] The simplest example appear in scalar fields with critical Morse points as singularities.

robots; here, we extend it ot incorporate transitions in agent´s behavior or in the whole scene (turning a corridor or a corner in a street, e.g.

0.2.3. The interplay between frameworks

In this paragraph, we sketch some basic ideas relative to the interplay between discrete vs continuous, statistical vs smooth, geometric vs kinematic frameworks. Mobile Platforms were originally designed for automatic navigation in indoor structured scenarios. A later relaxation of initial conditions has allowed to extend some functionalities to explore and navigate outdoor scenarios, including exploration or realization of tasks in hazardous, toxic (including waste cleanup, e.g.) or unaccessible environments (planetary exploration, e.g.). To perform these tasks it is necessary to integrate information about the environment (by using odometric resources, e.g.) which provide a feedback for tasks execution.

Hence, this process involves the whole PeCWA pipeline as a factorization of the PAC (Perception Action Cycle), linking sensors in terms of input signals. and actuators in terms of commands. Their integration is performed in terms of coarse-to-fine fibrations corresponding to smart systems. Along successive fibrations there is an interplay between "deterministic" and "probabilistic" frameworks. The intrplay is obtained by by "relaxation" of initial models or by using "regularization" strategies. An extension of Differential Geometry based on deformations vs stochastic perturbations of manifolds and superimposed structures provides an initial structural framework.

To increase the performance and ease the computational implementation, one must relax some of these hypotheses and replace them by other nearer to the discrete information provided by sensors. Furthermore, one must incorporate discontinuities and the uncertainty linked to the interpretation of processed information. Anyway, smooth models (given by manifolds M and their superimposed structures) provide a robust framework to adjust more casuistic and less structured information. A typical adaptation to the discrete under uncertainty conditions, consists of replacing ordinary distributions $\mathcal{D}$ of vector fields by some probabilistic discrete version usually given in terms of Markov fields, e.g..

By the same reason, the capture and generation of evolving volumetric models for the scene in $\mathcal{W}$ start from an initially already known simplified 3D reconstruction which can be generated in an automatic way; typical examples are given by perspective maps, e.g.. Next, one tries of adjusting the fundamental tasks (motion planning, effective navigation) to a "continuous deformation" where one has only a partial information about the environment (corresponding to a uniparametric family of evolving perspective maps, e.g.). Propagation models along the motion direction provides a "deterministic" support for the environment, where different kinds of events (agents operating in the scene) can be incorporated as they would be on an additional layer.

The above reasoning scheme for an evolving environment can be translated to Differential Geometry framework in terms of a distribution $\mathcal{D}$ of vector fields X_i to describe egomotion and motion of different agents, and a system $\mathcal{S}$ of differential forms ω_j representing constraints to evaluate geometric or radiometric "quantities". After linearizing at each point, the support for last ones can be thought as evolving hyperplanes which "separate" relevant information for the scene or its kinematics.

- In regard to the scene, typical examples are given in terms of "dominant planes" corresponding to architectural elements (walls or floor, e.g.) or their radiometric properties (color or textures, e.g.). Their automatic labeling and recognition must be "so independent as possible" of environmental conditions (lightening, relative orientation).
- In regard to its kinematics, typical examples are given by the simplest linear (first or second order) ODE; contact forms ($dq - pdt$, $dp - rdt$) involving generalized coordinates (q, p, r); reflectance maps for self-adapting to evolving lightening conditions, control and torque differential forms for self-adapting to evolving conditions. Their automatic recognition requires a bounded number of local patterns able of providing a PL-approach in terms of linear low-order ODE or PDE.

The space-time evolution of all these data can be described in terms of Lie derivatives for any kind of tensors, including scalar fields f on the ordinary space (height, depth, e.g.) or tangent space (total energy, Hamiltonians, Lagrangiands), vector fields X_i (linked to motion) or differential forms ω_j (work performed by a system, e.g.) as the most relevant "particular cases". Last ones are denoted as $\mathcal{L}_X f$ (directional derivative also called Frechet derivative), $\mathcal{L}_X X_i$ and $\mathcal{L}_X \omega_j$, respectively [18]; in this way, it is possible to describe the space-time evolution of planar or volumetric elements corresponds to the (real or apparent) motion of objects in the scene.

The discrete version (necessary by raw data provided by sensors) of information is developed in terms of PL-models which can be simplified in terms of cuboidal maps. A n-dimensional cube (resp.cuboid) is the image of the standard cube $[0, 1]^n$ by a PL (resp. continuous) map. Cuboids can be matched together along common faces giving "cuboidal complexes" (formal combinations of cuboids) with usual formal operations linked to the boundary operator. Ideally, a cuboidal map is a continuous map between complexes.

The discrete nature of information, partial occlusions and uncertainty about measures generates discontinuities along tracking which require to complete the information (interpolation, propagation, regularization) to recover continuous models. The statistical version of the above arguments, allows to introduce several procedures for information selection (sampling techniques, e.g.) and uncertainty management for clustered information arising from sensors. Along this

[18] The Lie derivative of a vector field or a differential form is the natural extension of the directional derivative of a scalar or vector function.

chapter, this extension is accomplished in terms of a probabilistic version of manifolds, (co)vector fields and superimposed global structures (vector bundles in the simplest case).

A more difficult problem is how to update events linked to the structure of the scene or, alternately, to objects:

- Sudden modifications in the scene structure which can be due to turn around a corner, e.g. or the (dis)apparition of elements belonging to walls or the ground due to partial occlusions, e.g.. At the lowest level, they are incorporated in terms of updating of perspective maps (involving an ideal mobile 3D reconstruction) and semantic maps (to ease interpretation). SLAM (Simultaneous Localization and Mapping) provides the support for their integration in an extended Structure from Motion (SfM) framework.
- From the dynamical viewpoint, it is necessary to incorporate events relative to the (dis)apparition of mobile agents (persons, cars, e.g.). Following our approach, they are managed in terms of coarse cubical representations (bounding boxes), which are superimposed to perspective representations in a restricted shape-from-motion (sfm) framework, where only cubical hulls are considered to ease their RT integration in a mobile scene.

So, cubical complexes (like-Manhattan scenarios) developed along the Chapter 1 play a fundamental role; in particular, contraction and expansion operators can be managed in a very easy way for cuboidal scenes and for bounding boxes for objects. From the differential viewpoint, the simltaneous management of a low number of evolving constraints is managed in terms of pencils, nets or webs for systems of differential forms depending on 1, 2, or 3 parameters in the ordinary space. In particular, evolving depth or height maps in a scene correspond to pencils or nets; similarly, in the Kinematic framework we can use energy and entropy fields to improve the management of objects in scene; if we wish specify more complex relations with the geometry of scene or objects inside, we will need more refined energy functionals (Wilmore nergy, e.g.) giving us an example for a web (three-dimensional famlly of differentialforms). In particular, optimization procedures must have in account both components.

0.2.4. Some strategies for resolution

Furthermore, even when the global problem is well-defined (a non trivial problem in presence of eventually mobile obstacles and any kind of loops), one follows an incremental strategy consisting of decomposing the global problem in local ones, and solving local problems separately. Often, local problems are referenced to previously known landmarks or, alternately, to meaningful elements which can be easily discovered and tracked by sensors, with a special regard to the visual information inputs.

Some relations between different modules can be described in terms of “relational maps” which allow to connect different partial subproblems. A typical example corresponds to information request by mutually related agents; it must be possible to share odometric information to reduce uncertainty levels by several collaborative agents. The interplay between local and global issues plays a fundamental role and it is ubiquitous in issues as diverse as the management of partially occluded regions, variable aspect of deformable objects, regular maps between objects or their superimposed structures (such as vector bundles, e.g.)

In more formal terms, when one requires higher accuracy or behaviors are inherently complex, one must use *local analysis* based on successive differentials of functions. So, for any function $f \in C^r(\mathbb{R}^n, \mathbb{R})$ with $r \geq 2$, initial differential data are given by the first and second order differential $df(p)$ and the Hessian $Hess(f)(p)$ at each point, which correspond to linear and quadratic approaches in the Taylor development series. Both expressions are widely used for the analysis of kinematic representations, because they correspond to first and second order variation rates of the observed parameters. It is very important to identify critical values (where the differential is vanishing) and their type (depending on the signature of the Hessian quadratic form at each singular point) to predict and understand the behavior in terms of kinematic events.

Global Analysis uses models and tools for patching together local inputs in a global framework. Usual data to be matched are given by fields, i.e. maps which assign a vector quantity to one or several functions which can vary in a simultaneous way. There are several kinds of fields which are labeled as scalar, vector or tensor fields:

- *Scalar fields* are represented by the different values which takes a function on a domain; they are locally represented by functions $f \in C^r(\mathbb{R}^n, \mathbb{R})$. In the discrete case, the range of values is limited and it can be given by a locally finite amount of values in a bounded interval.

- *Vector fields* are represented by a C^r-map which assigns to each function another function which depends on variable coefficients; in terms of local coordinates $\underline{x} = (x_1, \ldots, x_n)$ it can be written as an operator $\sum_{i=1}^n g_i(\underline{x})\partial/\partial x_i$ which maps each function f in $\sum_{i=1}^n g_i(\underline{x})\partial f/\partial x_i$ where $g_i(\underline{x}) \in C^r(\mathbb{R}^n, \mathbb{R})$ for $2 \leq r \leq \infty$ (the constraint about r is due to existence and uniqueness conditions for solutions of Ordinary Differential Equations). A typical example is given by the gradient field ∇ which to each function $f \in C^r(\mathbb{R}^n, \mathbb{R})$ assigns the vector function $\nabla f := (\partial_1 f, \ldots, \partial_n f)$ (it gives an ordinary vector when it is evaluated at each point). A *vector distribution* is a finite collection of vector fields.

- *Tensor fields*: Beyond vector fields, one can consider their evaluation at each particular point p_0 which given a real number which is called a *differential* or, more precisely, a differential 1-form which is denoted by means α. i.e., if ξ is a vector field, $\alpha(\xi) \in \mathbb{R}$, i.e. it is a dual version of a vector field. For example, the product ${}^T\nabla f \quad (\underline{x} - \underline{x}_0)$ represents a differential 1-form. The "exterior" product (an extension of the notion of vector product) of k differential 1-forms gives a k-form which allows to evaluate k

quantities varying simultaneously on a manifold M. A *tensor of type* (r, s) is a formal product of r differential forms and s vector fields whose behavior is compatible with coordinate changes in order to obtain a global object defined on a manifold.

It is clear that the above fields follow an increasing order of complexity, and the compatibility properties between local data, they allow to construct global objects on manifolds (or more generally, not necessarily smooth varieties; see below). By matching the above fields one obtains "sections" of superimposed structures verifying topologically trivial conditions (as product of varieties) which are called *vector bundles*. These conditions can be relaxed to achieve a more realistic adaptation to real situations, giving principal bundles or, more generally, topological fibrations. [19]

0.3. References for this introduction

References appearing below are not exhaustive, nor the most recent ones. They must be understood as an invitation to complete and improve the information given in the precedent paragraphs, and ease to the reader the completion of his/her own knowledge "reconstruction" according to his/her interests.

0.3.1. Basic bibliography

We include only classical textbooks or handbooks. More detailed references can be found at the end of the chapter. As usual, each chapter ends with a fifth section devoted to recapitulation including conclusions, practices, challenges and more detailed references. [Mur94] contains the first approach to the use of differential methods for manifolds in Robotics.

[Chr08] H.I. Christensen, and G.D. Hager: "Sensing and Estimation", Chapter 4 of [Sic08], 2008.

[Fan23] R.Fan, S.Guo and M.Junaid Bocus (eds): *Autonomous Driving Perceptio: Fundamentals and Applications*, Springer, 2023.

[Goo16] I.Goodfellow, Y.Bengio and A.Courville: *Deep Learning*, The MIT Press, 2016

[Koh97] T.Kohonen: *Self-Organizing Maps (2nd ed)*,Springer-Verrlag, 1997.

[Lat91] J.C.Latombe; "Robot Motion Planning", Kluwer, 1991.

[Mur94] R.M. Murray, Z.Li and S.S.Sastry: *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[Oks03] B.Oksendahl: *Stochastic Differential Equations. An Introduction with Applications*, Springer, 2003.

[Pha23] Hoang Pham (ed): *Springer Handbook of Engineering Statistics (2nd ed)*, Springer, 2023.

[19] See my notes on Differential Topology for additional details.

[Sic08] B.Siciliano and O.Khatib (eds): *Handbook of Robotics*, Springer-Verlag, 2008.

0.3.2. Software resources

Only open source references are included. Some of them could be obsolete. Any suggestion to complete and improve the following is welcome.

- g2o is an open-source C++ framework to optimize graph-based nonlinear least-square problems in SLAM context. [20]
- The TDA (Topological Data Analysis) package package provides an R interface for the efficient algorithms of the C++ li-braries GUDHI, Dionysus, and PHAT.

Final remark: Readers which are interested in a more complete presentation of this chapter or some chapter of this module B_{32} (Automatic Navigation), must write a message to javier.finat@gmail.com or to zhicheng.hou@gpnu.edu.cn.

[20] https://openslam-org.github.io/g2o.html