B322 Motion Planning

Javier Finat and Zhicheng Hou

Índice

Previous remarks: These notes corresponds to the Chapter 2 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. Mathematical Analysis, Graph Theory, Discrete Mathematics and Probability Theory.

From the computational viewpoint, it is convenient to be familiar with Object Oriented Programming (OOP) framework or, more generally, with Python to ease connections with Functional Programming, also.

In addition of this introduction and a fifth section for recapitulation, materials of this chapter are organized in four sections. They contain a list of exercises for self-verification of understanding of materials. 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 B322

The *main goal* of this chapter is to introduce some basic ideas about **Motion Planning** for a mobile robot $\mathcal{R}$ or, more generally, several autonomous vehicles $\mathcal{V}_j$ in indoor structured or outdoor unstructured scenes. To simplify, one supposes known the geo-positioning of autonomous vehicle(s) in a connected topological space X. Otherwise (GPS is not available in indoor scenes, e.g.) and in absence of local sensor networks in the scene, it is necessary generate an "environmental map" helping navigation strategies from image and range sensors. Basic notions of Motion Planning for Anchored Robots have been exposed in the precedent module B_{31}.

In the precedent chapter $B321$ one shows the convenience of an efficient design for sensor networks giving a complete covering of the whole environment where the autonomous vehicle must operate. The USA GPS and the Chinese BeiDou provide two standards for localization and provision of services in outdoor environments. For most indoor environments, neither of both systems work in an adequate way, and it is necessary

- to design and implement a fixed sensor network covering the whole space (public institutions as museum, or private firms as large stores, e.g.); or, alternately,
- to develop an Expert System able of RT processing from available data, analysis, interpretation and decision making from the available information.

Influence regions based on distance maps are crucial for the relative localization of sensors.A standard solution is given by *Voronoi diagrams* where each sensor is interpreted as a Voronoi site for the provision of services based on the nearest Voronoi sites. This approach must be completed by using visibility maps, and extended to mobille platforms by using mobile Voronoi diagrams.

The combination of distance and angular information (in prevision of partial occlusions) is key to obtain a covering of the whole environment. Some linked services can involve to the automated management in large logistic installations till to simple navigation models following a"reactive approach" to avoid collisions and provide customized services for each agent [1].

A more sophisticated extension can include some representation of an eventually changing scene. This goal requires a coarse-to-fine 3D reconstructions B_{22}, motion analysis of trajectories performed by other agents B_{23}, and a coarse recognition B_{24} of the nearest agents behaviours. A RT processing and analysis of this problem is the key to a smarter Decision Making.

It is clear, that 3D reeconstruction based approach has a higheer computational cost and requires embarked hardware and software devices to carry out reliable and adaptive solutions. Distance and angular-based models have been

[1] Some applications to CH environments have been developed by the MoBiVAP group (under the coordination of the first auhtor) for wheelchairs in CH environments

developed in the module B_{11}. Thus, the current chapter is mainly oriented to embarked image and range-based sensors at each mobile agent. In particular, one supposes a familiarity with basic aspects of the first three modules of computer Vision B_2.

A big challenge is how the above models can be learned in a semi-automatic way by using recent Machine Learning (ML) strategies involving advanced AI solutions as those arising from DNN (Deep Neural Networks). Increasingly realistic simulations are key to develop cheaper solutions for learning and training. Some AI-based recent contributions have introduced simplified models already learned to ease query processes in regard to the selection of the nearest model. This idea requires

1. the design of efficient strategies to reduce and cluster the information from sensors in $\mathcal{P}$;
2. an introduction of hierarchies according to relative importance functions involving "features" as structuring elements to re-organize the information ;
3. generate clusters of "similar features" in the configurations space C, by identifying the most meaningful elements (using SVD or PCA methods);
4. remap and pack features in "objects" in an extended Working space W according to a catalogue of shapes;
5. re-assign priorities according to constraints and the goals to be performed; and
6. make decision and execute the corresponding actions correcting errors in the action space A.

Real Time (RT) data processing and analysis of image and range-based information must assure an efficient reuse for automatic planning and free-collision navigation having in account the constraints for each level. After re-projecting the available information (initially in frequency domain) in a commonly shared spatial representation (space-time domain). *Hierarchies* involve to all spaces of the PeCWA pipeline which are now adapted to Automatic Navigation:

1. The planning in the *Perception space* $\mathcal{P} = (P, \mathcal{O}_P)$ involves to optimal design of sensor networks, including the corresponding (synchronous vs asynchronous) low-level information processing of evolving information in terms of multivariable time series, e.g.
2. The planning in the *Configurations space* $\mathcal{C} = (C, \mathcal{O}_C)$ involves the characterization of typical evolving features (multiple junctions, intensity maxima, e.g.) with expected regular values, or the apparition of "anomalous" configurations for higher levels of decision.

3. The planning in the *Working space* $\mathcal{W} = (W, \mathcal{O}_W)$ involves to the identification of "typical" objects or behaviours, for their *optimal assignation* depending on morphological constraints on W and functional constraints on $\mathcal{O}_W$ to be fulfilled.

4. The planning in the *Action space* $\mathcal{A} = (A, \mathcal{O}_A)$ involves the design and implementation of the *Decision Making* module in charge of selecting the most appropriate action and the *optimal control* system to warrant a secure and safe navigation.

Strategies for clustering and identifying the most relevant data, features and clusters are specific for each semi-analytic space $\mathcal{X} = (X, \mathcal{O}_X)$. The problem is not trivial because it affects to quite different inputs. In particular, similarity between signal characteristics in $\mathcal{P}$, have no relation with similarity between configurations of mobile points arising from video sequences. The last one have no immediate relations with any kind of (visible, convex, visual) hulls giving mobile information packages in the Working space $\mathcal{W}$. Finally, the action to be chosen follows adaptive patterns to evolving scenes, including sudden jumps in decision making depending on priorities and unexpected events.

Differential Equations and their discrete version (given by difference equations) provide a theoretical context. It is necessary to fix initial conditions for ODE to obtain trajectories (locally unique under Lipschitz conditions), whereas PDEs require boundary conditions to incorporate constraints to the solutions. Both of them are described by evolving vectors and covectors in the discrete case [2].

The most difficult problems concern to how to perform updating, tracking and prediction in a compatible way with unexpected events. To be applicable, Differential Systems must be relaxed in terms of deterministic vs random *perturbations* (by using Stochastic Differential Equations or MRF in the discrete case). The most complete and rigorous approach uses some variant of *Kalman filters* [3]. Here, we adopt a more pedestrian viewpoint based on simpler properties of image segmentation.

In presence of image and range information, initialization for automatic navigation selects the "most meaningful" regions or RoI according to a segmentation of planar vs volumetric representations of the evolving scene $\mathcal{S}(t)$. Furthermore the localization (position and orientation) of *Focus of Attention* (usually corners or intensity maxima) in an initial image segmentation [4]. The resulting image segmentation at each instant is the support for a semantic map including a coarse recognition of "radiometric features" contained in the planar or volumetric representations of the scene.

[2] More details about relations between differential and discrete differences systems can be read in the first chapter of B_{13} (Computational Differential Topology).

[3] For details see the chapter 6 of B_{33} (Computational Kinematics).

[4] It is given as a decomposition of the visible part of the scene in a disjoint union of radiometrically quasi-homogeneous regions (variation of the intensity level under a threshold).

The update of meaningful data along the motion requires efficient methods to represent evolving scenes $\mathcal{S}(t)$ following a coarse-to-fine strategy. The space-time evolution of FoA gives Focus of Expansion (FoE) to be tracked and updated in a *mobile segmentation*, including sudden unexpected events which can modify

1. The coarsest *radiometric level* involves to planar vs volumetric regions which are characterized in topological terms (planar vs spatial ordering, connectedness, compactness). Topological properties for objects are only preserved for short video mini-sequences (between two video shots, e.g.) where evolution is described by homeomorphisms.
2. An intermediate *geometric level* involves to geometric properties of the scene which are initially formulated in geometric terms. The hierarchy between geometric models (Projective, Affine, Euclidean) and their transformations groups provide the key for more robust representations.
3. A higher *kinematic level* concerning to the update of radiometric and geometric properties. Usual approaches based on Optical Flow are very unstable, require non-realistic hypotheses and generate a lot of noise around the objects boundaries which are very meaningful for Automatic Navigation. Our approach uses structured models based on Lie algebras $\mathfrak{g} = T_eG$ for each Lie group G of transformations used to represent the scene (projective group and its subgroups) or motion characteristics (symplectic group preserving ideal motion's equations).

As always, the most difficult problems concern to the interplay between the above items. Topology is the natural extension of Geometry; thus the second item (concerning to the Geometry) can be considered as an "instantiation" of the first one (concerning to Topology). This claim will be justified later in terms of "retraction-expansion" representations involving the Topology and Geometry of visible regions (star-shaped polygons and polytopes). Connections between Lie groups and algebras for locally symmetric spaces (to ease propagation models) are developed by taking the differential and the exponential as usual

In more advanced settings it is necessary to have in account some "delicate" properties involving relations between local and global issues which have already appeared in the module B_{31} (Anchored Robots) in differential terms. An advantage of the topological approach is linked to their "topological stability" (independence w.r.t. the dimension) of superimposed structures [5]

A daily motivation for this structural approach is linked to the automatic driving of vehicles. Highway traffic requires only an affine framework where the horizon line is fixed for each driver; sharing this information with other vehicles or the updating of this information for each driver requires a projective framework where the horizon line changes. Similarly, when one drives in

[5] This result is not trivial and requires some additional elements of Geometric Topology which have been developed in the module A_{24} of the matter A_2 (Algebraic and Geometric Topology).

an urban environment, perspective representations of scenes provide an affine geometric framework with evolving structural elements (vanishing points, e.g.). However, when one tries of parking the vehicle, it is necessary to introduce an Euclidean framework to avoid contact with other vehicles, and to incorporate non-holonomic aspects.

The precedent simple remarks show the need of developing a similar hierarchy for automatic driving in artificial devices having in account the Kinematics for the scene. The most commonly used methods for video-based information updating are based on SLAM (Simultaneous Localization and Mapping) in the configurations space $\mathcal{C}$. This method must be reprojected on a coarse perspective representation of the evolving scene to ease their interpretation in the Working space $\mathcal{W}$. A hybrid approach incorporates geometric perspective and radiometric semantic maps which are useful for the central stages o the Basic Anlaytic Pipeline (BAP).

From a classical viewpoint, perspective models have been used from the Renaissance to represent scenes. The Alberti's method (1435) provides a basic strategy to propagate simple perspective models from the knowledge of vanishing points, which has been used in Western culture from the 15th century. Pinhole camera models use central projections which are easily interpreted in terms of perspective models. However, the update of information, prediction and simulation of decisions making, and, more recently, information shared with other agents introduce additional complexity for automatic navigation issues which require a Real-Time (RT) generation and updating of perspective maps.

After image and range processing in $\mathcal{P}$, one must extract the most meaningful elements in $\mathcal{C}$, which are grouped in objects caontained in the scene (as a model for $\mathcal{W}$). An efficient design and implementation must provide a real-time interpretation and updating of the above stages at different Levels of Detail (LoD). The final stage is the selection of optimal decision for a safe and secure navigation.

From the 1980s there are several increasingly complex strategies for the above problems going from structured to more open environments. In absence of interaction between different agents, the simplest cases correspond to monitored indoor scenes (such as office or Lab environments, e.g.), where a collection of regularly distributed marks or beacons are available. The most complex cases correspond to natural outdoor scenes.

The problem becomes harder in presence of several fixed or mobile obstacles which can evolve and interact between them. In this case, uncertainty grows and one needs more adaptive strategies in a robust framework provided by the kinematic extension of geometric frameworks. A quite general strategy for collision-free navigation and smart interaction (not only a reactive behaviour, e.g.) displays a lot of open challenges, still. In this chapter we develop a "qualitative" approach. The final goal is to try of contributing to automatic driving in Traffic Scenes and Intelligent Transportation.

0.1.1. A topological reformulation

If one has previous information about the environment and the own localization, motion planning in a space X is a topological problem. In the simplest case, given an initial element $x_0 \in X$ and a final element $x_1 \in X$ one wishes find a collision-free path $\gamma : [0,1] \to X$ such that $\gamma(0) = x_0$ and $\gamma(1) = x_1$. A basic example is given by the image of the segment $\gamma(t) = (1-t)x_0 + tx_1$ where $t \in [0,1]$ is the affine or barycentric coordinate of the segment connecting x_0 and x_1 in X. Its functional formulation is given by a continuous interpolation $(1-t)f_0 + tf_1$ between regular functions $f_i \in \mathcal{O}_X$ on X.

A first drawback consists of inteerpolating "segments" are not necessarily contained in the ambient space. Thus, a previous analysis of the topology of X or $\mathcal{O}_X$ is required. More complex to solve drawbacks are linked to the variability of the scene and agents along time. In practice, it is not possible to have a "complete" catalogue of possible behaviours, and one must adopt some simplifications involving the scene and its variability.

The description based on multipaths $\Gamma = (\gamma_1, \ldots, \gamma_k)$ corresponding to k agents $(a_1, \ldots, a_k)$ with the corresponding behaviours is called a *Multi-Agent system (MAS)*. Typical advanced examples appear in simultaneous interaction between agents (pedestrians, cars, trucks) in *traffic scenes*, which is the main topic for all this module. Their joint management is performed in terms of *Mixed Integer Programmin (MIP)*, where we adapt *Iterated Local Search (ILS)* strategies to find near-optimal solutions.

Some basic ideas of this challenging problem are shared with the Automation of Intelligent Transport and Smart Logistics. In the last case, one has a lot of inputs (relative to infrastructures and goods, e.g.) and a lot of outputs (services to be provided as regular fields) under a lot of constraints. This simple remark motivates the development of different Optimization strategies on a *Multi-Input Multi-Output (MIMO)* model. Their adaptation to Integral Logistics i carried out in terms of *Multi-Agent Path Planning (MAPP)*.

From a simplified local *geometric viewpoint*, the Affine Geometry allows the simultaneous management of k agents in terms of barycentric coordinates $(t_1, \ldots, t_k)$ fulfilling $0 \leq t_i \leq 1$ and $\sum_{i=1}^{k} t_i = 1$, to be interpreted as "relative weights". The main invariants of the Affine Geometry are linked to the "ratio" between quantities which is preserved on "parallel" pencils of lines. The main invariant of the projective completion of the Affine space in terms of the Projective space is the cross-ratio [6]

In the functional case, the expressions $\sum_{i=1}^{k} w_i f_i$ for $f_i \in \mathcal{O}_X$ for the relative weights $\underline{w}$ can be interpreted as a simultaneous interpolation of ordinary pdf (probabilistic density functions) in the probabilistic context. The simple interpolation between functions (or more general fields) does not give a function having "near" properties to the extremal ones. This is due to the non-linear character of the ambient functonal space.

[6] See the chapter B_{220} of the module B_{22} (Three-Dimensional Reconstruction) for details.

(*) To be more precise, one must take a "folding" representing "generic deformations"; if unique, it is called "unfolding". Explicit computations for unfoldings of "simple singularities" of function germs have been developed in the module A_{43} (Singular function germs). Their extension to more general foldings of non-regular map germs are developed in the module A_{44} . Due to the high complexity of matehamticla expressions we limit ourselves to the "quasi-regular" case, i.e. regular maps (submersions and immersions in the PS framework), and codimension one map germs for the singular case.

In Automatic Navigation it is difficult to follow an ideal path γ in an exact way; the problem becomes more difficult for paths defined on curved manifolds. In particular, rigid constraints show that allowable motions are composition of planar rotations and translations . Only combinatorial properties of "deformations" for closed paths are useful for motion planning. The fundamental group $\pi_1(X, x_0)$ corresponding to homotpy classes of closed paths with base point $x_0 \in X$ provides a coarse invariant to classify surfaces with g holes; in practice, holes correspond to obstacles to be avoided (columns in an indoor scene, other vehicles) [7]

In presence of k autonomous vehicles or agents a_i, one must consider a multipath $\Gamma = (\gamma_1, \ldots, \gamma_k)$. Depending on the existence of possible interactions or not, multipaths Γ can be visualized in terms of

- a surface given as the connected sum $\mathbb{T}^2 \# \ldots^g \ldots \# \mathbb{T}^2$ of g two-dimensional toruses $\mathbb{T}^2 := \mathbb{S}^1 \times \mathbb{S}^1$ (donuts), where each hole represents an obstacle to be avoided in automatic navigation; or alternately, as
- a k -dimensional homogeneous manifold given by the product of k copies of $\mathbb{S}^1$, i.e. a k-dimensional torus $\mathbb{T}^k := \mathbb{S}^1 \times \ldots \times \mathbb{S}^1$, which is useful for topological models in ITS.

Each one of basic pieces admits quite different embeddings in the ambient space. A *knot* is the image of a closed path $\mathbb{S}^1$ in the ordinary space $\mathbb{R}^3$ or in a sphere $\mathbb{S}^3$. A *link* is given by the image of two or more knots which are interlaced between them without a common intersection. A *braid* is a finite collection of interlacing paths in $\mathbb{R}^3$ or $\mathbb{S}^3$ (unit quaternions). Links and braids provide patterns for traffic management in space highway crossroads or for aerial navigation of UAV, e.g.

Other central contributions of Computational Algebraic Topology B_{12} are given by *superimposed PL-structures* (PL: Piecewise Linear) to the support given by a (eventually discrete) topological space X. In addition of (equivalence classes of closed) paths. Topological properties of the support X are computed by using combinatorial characteristics of multipaths or, alternately, (triangular vs quadrangular) meshes. Invariants of PL-structures are expressed in terms of equivalence classes of q-chains $c_q(X)$ (given as linear combinations of simplices or cuboids).

[7] More details can be found in the module A_{21} (Basic Homotopy) of the matter A_2 (Algebraic and Geometric Topology).

So, simplicial vs cuboidal representations appear as the natural extensions of the well known triangular vs quadrangular decompositions of planar tessellations. They provide unordered vs ordered deformable topological support for objects or behaviours for evolving scenes in $\mathcal{W}$. The main problem is to develop enough efficient (flexible and fast) algorithms to describe the updating of evolving PL-structures. Perspective models provide the key for quadrangular decompositions which are easily extended to cuboidal representations (by using extrusion strategies of Computer Graphics). Their updating is performed in terms of RT generated quadtrees and octrees, respectively.

The topological management of multipaths Γ corresponding to k agents on a PL-approach to the scene requires starting and ending time, task duration and reward-penalty rules in regard to tasks execution. A computational approach is performed in terms of *Multi-Agent Path Planning (MAPP)*, which is a variant of the classical *Multiple Travelling Salesmen Problem (MTSP)*. Some meaningful extensions of this description involve to

- *Internal data* provided by embarked sensors of discrete vs continuous nature; dense vs sparse distribution, spatial vs frequency domain, between others, to have information about the own state of the mobile platform in th Perception space $\mathcal{P}$.
- *External data* about the *scene* $\mathfrak{S}(t)$ supported on evolving clouds of "basic features" (points or oriented segments) . The scene can be known vs unknown, indoor vs outdoor; structured (man-made, e.g.) vs unstructured (sonar for submarine exploration, e.g.); monitored in the short range (beacons, markers, BT, RFID sensors, e.g.) vs monitored only by remote sensor networks (satelitar information for Earth or planets, e.g.)
- *Obstacles* in the Working space $\mathcal{W}$ Denoted as $b_\beta(t)$ for the planar case, or $B_\beta(t)$ for the volumetric case; they can be static (constant w.r.t time t) or mobile (depending on t), known vs unknown or variable shape, etc The union of obstacles is denoted as $\mathfrak{B}(t)$, with complementary *collision free-space* denoted as $\mathfrak{F}(t) := \mathfrak{S}(t) - \mathfrak{B}(t)$.
- *Interaction* with the scene $\mathfrak{S}(t)$ or with other mobile agents $\mathbf{a}_i(t)$ (human or robots) at different levels: avoiding collisions, sharing information, collaborative behaviors, assistance, between others.
- *Mechanical issues* in regard to the superposition of (geo)metric information for the scene, or low-level descriptions of the corresponding kinematics.

In an introduction as this one can not give a detailed analysis of all these topics. To simplify, we will restrict along the first section to the case of once a mobile platform $\mathcal{R}$. Several extensions will be given in the other 3 sections.

0.1.2. Some remarks about interaction

The next step consists of incorporating some basic properties arising from the interaction with other agents in a HRI (Human Robot Interaction) framework as extension of the HCI (Human Computer Interaction) paradigms. Some remarks in regard to interaction issues are the following ones:

- *Internal data:* We have in account the information arising from multiple sensors in an evolving Perception space $P(t)$, which have been described in the precedent chapter [8]. In particular, we paid attention initially only to range and image sensors for once a mobile platform, which will be enlarged to other sensors in the last 2 sections. In the mobile case, furthermore the individual processing and analysis, the first tasks to be accomplished concern to the fusion of time evolving information. We will use multivariate variants of ARMA.
- *Modelling evolving configurations* $C(t)$ (including obstacles) by using adaptive envelopes (cuboidal, convex, α-shapes) generated from semi-dense clouds of points (SLAM) vs sparse clouds of segments (to extract motion characteristics). Learning obstacles from both clouds is an advanced research problem in a changing Configurations space $C(t)$. Thus, we shall restrict ourselves to regular configurations of segments, such those appearing in simplified perspective scenes.
- *Evolving scene representation* of the working space $W(t)$ to different levels including objects and behaviours in $W(t)$. They are modelled by suing planar vs volumetric, visible vs virtual or augmented (in prevision of events) evolving primitives. We use a two-layered perspective vs semantic segmentation which is managed in terms of symbolic representations (dynamical graphs, e.g.)
- *Basic patterns for interaction* in an evolving Action Space $A(t)$ by following centralized vs decentralized paradigms, collaborative vs opponent behaviors, coupled vs decoupled tasks for the Decision Making module.

All the above processes are mutually dependent,, and require a specification of *adaptive geometric frameworks* and relations between them depending on the required accuracy for (coarse-to-fine, relative vs absolute) localization criteria and to perform automatic navigation tasks. In this chapter we adopt a description in terms of data arising from signals, clusters of "features" in C, and their grouping in evolving geometric primitives in W. More advanced structural kinematic models to be developed in the next module B_{33} (Robot Kinematics).

Anyway, one must be aware that all localization and navigation issues are prone to errors which must be corrected on-line at each stage. This correction means that, if errors are beyond a tolerance threshold, we must adapt the current

[8] An on-line reference is http://planning.cs.uiuc.edu/node1.html

trajectory to a nominal one; up to risky or catastrophic events which must be recognized, also by a supervisory module in regard to tasks to be performed.

(*) More sophisticated issues concern to the simulation of signals, configurations, structured objects and possible decisions to be made. A good feedforward design is crucial to develop cheaper AI-based simulation for learning and training in quite different scenarios for Automatic Navigation. The design and implementation of software tools will be developed in the module B_{44} (Simulation and Animation) of B_4 (Computer Graphics).

In the meantime, we use already available tools of Advanced Visualization B_{16}, whcih give a support to databases, processing and analysis, insertion of basic primitives, and low-level interaction models for evolving objects. For more advanced applications, it is necessary to develop multi-scale adaptive solutions for quite different issues, going from simplified traffic scenes till the semi-automatic management of merchandises in very large stores or logistic centers.

0.1.3. From topological to metric issues

The alternance between more and less structuted environments even for traffic scenes, suggests an exchange between topological and metric properties at each level of the Basic Analytic Pipeline (BAP). The most general topological properties to be evaluated are connectedness (involving number of components), compacity (to warrant convergence) and separability (to discriminate components). All of them involve paths for basic objects and behaviours, even under uncertainty conditions due to partial occlusions, reflectance or bad illumination conditions. The last ones can require luminance-based approaches and low-level restoration to complete objects, which are artificially separated due to bad environmental conditions.

A challenge is related to the *automatic completion of information*. One can use sensors at different resolutions, by starting with low level restoration, and incorporate initially supervised methods linked to approximate shapes and motion characteristics. To accelerate completion, we restrict ourselves to simple evolving envelopes in simplified perspective models given locally by frontal or angular perspective models (corresponding to one or two vanishing points at finite distance). From the statistical viewpoint, it is necessary to develop a topological extension of the Geometric Information Theory (GIT) which is labelled as TIT (Topological Information Theory)

All the available information is submitted to *uncertainty* conditions, and propagation models are "degraded" after 2 or 3 iterations. By this reason, it is necessary to have *robust reference models*, and specify the *error models*, (having in account the need of giving a RT response for traffic scenes). From the AI viewpoint, it is necessary to perform a previous training which uses *supervised learning* of linear patterns to be combined by means *superposition principles*.

The *correction of errors* puts the accent on metric properties (minimize distance between current and nominal or expected behaviour), and an estimation of different kinds of errors and how to minimize them in terms of control functions. Minimization of errors must be performed on the kinematic layers, also. Accuracy requirements are not the same for an isolated terrestrial robot than for a flock of UAV, e.g. Furthermore, some constraints are specific of a higher level and do not descend to the geometric level.

For *known static man-made scenes* motion planning is “easy”. In absence of occlusions, it suffices to align current data about the environment with the “nearest part” of nominal plane or volumetric representation. Even so, the problem is not elementary, because it requires efficient search algorithms in regard to beacons, markers, or any kind of signals in the scene. If one intends a navigation based on visual alignment in $\mathcal{W}$, the problem becomes more complex because it requires image segmentation, generation of perspective maps and estimation of the geometric transformation performing the alignment for both of them. These issues are considered in the first section.

A statistical treatment of mobile information is classically incorporated in the third section, by using terms of stochastic process vs Markov Random Fields (MRF), depending on the availability of high vs low structured patterns. The Symplectic Geometry provides a structured framework to integrate all these issues in a common kinematic framework. The corresponding Information System is developed in terms of *Kinematic Information theory* (KIT) to be developed, which can be considered as an extension of Geometric Information Theory [Ama16] to Kinematics.

Let us remember that usual approaches to the *Geometric Information Theory (GIT)* use a mixture of Statistics and Riemannian Differential Geometry on the space of (initially parametrized) statistics of multivariate distributions. In particular, the introduction of the Cramer-Fisher-Rao metric ds^2_{CFR} induces a structure as a Riemannian Manifold (M, ds^2_{CFR}) on the space of multivariate normal distributions, with the corresponding metric connection ∇_{CFR}, which is nothing else that the CFR adaptation of the Levi-Civita connection ∇_{LC}.

(*) In our case, due to the systematic use of perspective models (supported by an evolving Affine Geometry), it is necessary to replace the metric connection by an affine connection ∇_a and its corresponding statistical version. The maininconvenient of this extension is the existence of infinite affine connections, which requires the introduction of a “variational principle” linked to “some kind of curvature operator” (locally represented by the covariance matrix).

In the KIT framework, the initial kinematic paradigm is given by th Analytical Mechanics initially developed by Lagrange and Legendre at the beginning of the 19th century. From the middle of the 19th century (Hamilton), the introduction of the now called *Symplectic Geometry* (Hamilton-Jacobi) plays a similar role on the Poincaré Phase space P (total space of the co.tangent bundle) to the metric on the original manifold M. The incorporation of the variational viewpoint (the first systematic approach is due to L.Euler), was performed by Lagrange. This incorporation gives the so-called Euler-Lagrange or integral approaches. In absence of external forces, integral adn differential (Hamilton-

Jacobi) are equivalent between them.

(*) In presence of external forces (corresponding to different kinds of interactions), it is necessary to perform "small perturbations", in terms of non-vanishing terms in the right-hand of structural equations. Unfortunately, the equivalence between differential and integral approaches is not longer true. Thus, one must choose the corresponding framework. By coherence with simpler methods developed in the module B_{23} (Motion Analysis) of B_2 (Computer Vision), along this chapter we privilege differential methods labelled as Hamilton-Jacobi (in despite several important contributions are due to Lagrange and Legendre, also). So, the KIT initially modelled as the extension to Statistics to the Symplectic Geometry.

0.1.4. Symbolic management

Most of the above issues involving different terms of the Basic Analytic Pipeline (BAP) linking PeCWA spaces, can be formulated in terms of Optimization problems, and the corresponding dual Control strategies for tracking and correcting behaviours. *Optimization* in a space X is carried out w.r.t. a finite set of constraints, which are locally given by equalities $f_i = 0$, inequalities $g_j \leq 0$ and strict inequalities $h_k < 0$. Their simultaneous consideration is a hard problem to solve, specially in the non-linear case. Thus, a basic strategy consists of decoupling, by removing some of them, and re-coupling in successive stages.

For each removed hypothesis the problem becomes more and more ambiguous and, consequently, difficult. In absence of previous information about the evolving scene $\mathfrak{S} = [S(t)]_[t \in I]$, a first problem concerns to the automatic generation of the "nearest" simplified representations for the environment from the information fusion arising from different sensors; a second problem is the management (inserting vs removing nodes in dynamic graphs) of eventually mobile objects in the environmental layer. Efficient solutions require the introduction of hierarchies to process the information for each one of the spaces appearing in the Basic Analytic Pipeline (BAP)

1. Hierarchies for *sensor fusion* in the *Perception* space $\mathcal{P}$ suggest the use of similar sensors working at different resolution levels. A coarse-to-fine approach provides a general coarse framework, where refinements are included in a selective way (only for RoI) in depth along "vertical direction". Propagation is performed at each resolution level in an horizontal way. A symbolic representation of the corresponding tree can be visualized in terms of upper block triangular matrices whose rectangular boxes of the same "row" represent propagation along time for each abstraction level, and whose "columns" represent successive refinements of RoI.

2. Hierarchies to *update and track features* in the *Configurations* space $\mathcal{C}$ are introduced from an initial configuration C_0 of isolated junctions and intensity maxima in a *Visibility Graph*. Their space-time evolution is described in terms of pdf (probability density functions) linked to normal

distributions around each "feature". Updating of the "features graph" is performed by using queries on the nearest neighbors for each node 8By using a variant of the k-NN algorithm).

3. Hierarchies for *evolving objects* in the *Working* space $\mathcal{W}$ are described in terms of coarse-to-fine objects recognition at different LoD. Flattening graphs provide a coarse description for polyhedral objects to be validated as embedded subgraphs of the *Aspects Graph* for standard objects. In man-made scenes, a symbolic representation is given by structural elements of a perspective representation (vanishing points, perspective lines, e.g.). For non-structured scenes, the most maingful information is given by the space-time evolution of FoA, given by FoE.

4. Hierarchies for *Decision Making* in the *Action* space $\mathcal{A}$ use different optimization criteria involving increasingly complex criteria (linear, convex, non-linear smooth). Statistical Decision Theory is the key.

The big challenge is how all these hierarchies can be learned, and how can be modified in a semi-automatic way. Related topics are developed in the chapter B_{326}.

0.2. Outline of the chapter B322

In a nutshell, along the first section, one develops a static approach which are supported by geometric (perspective maps) and topological models (regional segmentation); both of them incorporate uncertainty levels involving the environment. The second section introduces several dynamical aspects, involving paths, interpolation and some applications to more realistic environments, where basic types of interaction are allowed. Third section is focused towards situations where several agents operate in a "similar way". Last section introduces heterogeneous clusters which interact between them with a higher uncertainty level and some unstability phenomena linked to the lack of control for mechanisms. Some advanced illustrations are introduced to motivate further developments.

More specifically, in addition of this introduction and the fifth section of complements, this chapter has four sections labeled as

1. *Evolving representations of scenes*, starting with $2D$ cenital vs $2{,}5D$ perspective representations as initial paradigms. Their time evolution is described in terms of planar affine transformations (with PL-approaches for curved elements). In absence of cenital views, RT estimation of perspective models provide local models for Visual Navigation. A final subsection devoted is devoted to a short description of basic algorithms.

2. *A topological approach to Motion Planning* is developed in the section 2 to provide a common framework for semantic and perspective maps. Homotopy methods provide some keys for information reduction in an invariant

way. Extrusion of perspective maps from depth or height functions allows a RT generation of 2,5D representations. Finally, one introduces a multivector approach adapted to cuboidal maps for a joint management of 2,5D perspective models.

3. *Statistical methods for Navigation* are crucial for clustering, sampling, extracting and remapping meaningful features in parametrized models. To achieve these goals, one introduces second order methods and one describes some basic Optimization procedures which are applied to ITS (Intelligent Transportation Systems). One describes some basic algorithms for automatic extraction, which are applied to mobile scenes.

4. *Learning of evolving perspective maps* is the most complex section. In view of the very large diversity of motion patterns, we start with a based-models structured approach (having in account basic Analytic Mechanics for Kinematics). Next, one relaxes these conditions to incorporate the learning of low-dimensional patterns for evolving isolated objects (cars, people, e.g.). Their fusion is developed in simplified cuboidal representations for the scene. The last subsection proposes an extension of learning procedures which is based on joint evolving kinematics.

Several topics will be developed in depth along successive chapters. By this reason, in this chapter we consider only simplified or “toy models”, as an initial description of more realistic models which will be incorporated later. The next paragraphs of this subsection illustrate some methodological issues, jointly with some illustrations showing the interplay between different approaches.

0.2.1. Some specific troubles

Complete information about a scene is an unrealistic hypothesis. One must incorporate uncertainty or incomplete information about the scene in all Navigation stages. Hence, smart navigation requires some capability to generate low-level environment representations for indoor as for outdoor scenes. It is very common the existence of “shadow zones”, where communication is degraded or, simply, is lost.

In these zones, autonomous robots must be able of developing their tasks, including possible cooperation with other robots. Furthermore, the use of local terrestrial networks (as WiFi, BT, RF, e.g.) has a limited bandwidth which degrades information contents even if the coverage is “complete”. A related challenge is the incorporation of AR modules to smart mobile agents for completing information. It requires more advanced AI models and tools for a simple management of “similar” situations. Again, it is necessary to develop a more topological approach.

In some cases, even if the communication with a central node is lost for some agents (due to partial occlusions, e.g.), the presence of other agents with whom the information is shared, allows to recover an indirect communication by selecting a new leader (who receives the most complete communication). This implies the need of developing tools for reconfigurable floats which is performed in the framework of MAS (MultiAgent Systems). An illustration to Intelligent Transportation systems is developed in the subsection §3,3.

In the last section we specify computational tools and provide some meaningful examples relative to motion planning with a special regard to dynamic groups. In particular, we paid a special attention to some qualitative aspects of cooperative vs collaborative behaviours with several examples to illustrate our approach which appear in a recurrent way in other chapters of these notes.

0.2.2. Methodological issues

As usual, we follow an increasingly complex strategy, where we prior top-down approaches along the two first sections, and bottom-up approaches along the third and fourth sections. *Top-down approaches* have a geometric vs kinematic character, where kinematics is understood as "geometry in motion" (in an initially ideal Symplectic framework). Next, along the second section one "relaxes" rigidity conditions of geometry, by replacing it by a more topological approach, where deformations and abrupt changes (corresponding to visual events) are allowed.

In a complementary way, *bottom-up approaches* incorporate mobile data in terms of multivariate time-series, where ARMA (Auto-Regressive Mean Average) methods play a central role. The main problem to be solved consists of identifying "typical patterns" appearing along the motion. Covariance matrices to estimate the curvature, are replaced by correlation matrices to estimate motion patterns, as a previous step to be refined in more advanced stages by covariance based methods. The most difficult problems concern to the design of efficient methods for learning the above issues, which is the main topic of the fourth section. Next, we remember some basic ideas to motivate this strategy.

Mobile Platforms were originally designed to navigate 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 clean-up, e.g.) or unaccessible environments (planetary exploration, e.g.). To perform these tasks it is necessary to integrate an accurate odometry along the tasks execution which involves all sensors and commands. Their first integration is performed in a "deterministic" framework given by manifolds (i.e. smooth or differentiable varieties) and locally trivial structures defined on them (vector bundles, e.g.).

Hence, we suppose initial models for variation rate of information relative to signals and commands are smooth maps, i.e. functions or coefficients of distributions describing models superimposed to expected variables are differentiable ones. Hypotheses about smoothness imply that the first mathematical framework is given by Differentiable Manifolds. Later, we relax these hypotheses and replace them by other nearer to the discrete information provided by sensors, or the uncertainty linked to the interpretation of processed information. In last case, one can replace ordinary distributions of vector fields by some probabilistic version (usually given in terms of Markov fields).

By the same reason, first models suppose that the environment is already known and try of adjusting the fundamental tasks (motion planning, effective navigation) to situations where one has a complete information about the environment in a "deterministic" environment. Later, we introduce several procedures for uncertainty management arising from sensors, even in differentiable case. This extension is accomplished in terms of a probabilistic version of manifolds, fields and superimposed structured.

Furthermore, even when the global problem is well-defined, 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 "visual features" (landmarks, beacons, intensity maxima, e.g.) detected from visual sensors. The connection at different levels involving PeCWA spaces is represented in terms of relational maps. So, in addition of the informaiton processing and analysia t each level, it is necessary to design and implement how this information is reconverted to their use for the next PeCWA spaces in terms of analytical maps $\mathcal{X} \to \mathcal{Y}$ by minimizing errors and uncertainty at each level.

The computational management is carried out in terms of symbolic representation given by analytical graphs $\mathcal{G}$. The usual superimposed structures $\mathcal{E}(\mathcal{G})$ provide the support for multiple purposes on $\mathcal{G}$. In particular, they allow the design of feasible tasks (compatible withmetric and kinematic constraints and manoeuvrability capability). Feasible paths are usually approached by PL- or PQ-trajectories performed by "meaningful" control points of the mobile platform involving C and W.

In addition of feasibility, one must implement algorithms to find *near-optimal* trajectories in terms of scalar fields (potential functions, e.g.) vector fields defined on the support G of $\mathcal{G}$ (involving flows along the graph), under the corresponding constraints (linked to evolving covectors or differential forms in the PS context). Kinematic and Dynamic aspects involving the behaviour of the c.o.g. $\mathbf{G}$) or linked to the end-effector of kinematic chain will be developed in the modules B_{33} and B_{34}. From the topological viewpoint, we limit ourselves to simulate and validate solutions for ideal motion equations.

A coarse approach to the above problems is developed in a PL framework with two layers, where the semantic map (linked to regional segmentation) is superimposed to the geometric layer in W corresponding to each perspective map. In despite of the linear character of primitives, the generation of a perspective map is not a linear problem. In particular, the relative depth is a highly non-linear function as the inverse of space or time disparity between homologue elements. Some easier PL models to be estimated are given by weak perspective and paraperspective models which provide a discrete approach for scene representation, relative location and navigation tasks [9].

The intrinsic PS nature of egomotion requires to develop some differential refinements for the PL-approaches of the initial perspective models. Furthermore the non-linear character of perspective representations (depth as inverse of disparity) and groups of transformations involving them (classical groups are manifolds, not vector spaces), motion's equations are usually non-linear. Furthermore, there appear non-holonomic effects which is necessary to evaluate and correct in an automatic way; typical examples are giving by parking operations, e.g.. Thus, it is necessary to develop more involved Algebraic and Differential Topology methods for solving motion planning issues, at least from a theoretical viewpoint.

As always, the most difficult challenges concern to semi-automatic learning of all the above issues, which require advanced elements of Deep Learning. A classical approach to evaluate space-time evolution use RNN (Recurrent Neural Networks), initially introduced in the ninetiees. Transformers are an extension of RNN which incorporates basic features of dynamical systems. However, even if we forget semantic aspects (involving mainly to changes in radiometric properties of RoI), there are a lot of qualitative problems to be solved involving topological aspects of PeCWA spaces. Some of them are the following ones:

1. Complete information arising from sensors, by removing noise and lowering uncertainty in $P(t)$.
2. Select, track and predict the most meaningful "evolving features" in $C(t)$.
3. Reproject the available information on evolving coarse-to-fine representations of $W(t)$.
4. Select the near-optimal action to be accomplished in $A(t)$ according to constraints.

All these issues require and efficient design for Optimization criteria, and the incorporation of Expert Systems to reduce information for each PeCWA space. Furthermore, it is necessary to evaluate how constraints are propagated along the "transfer map" $X \to Y$ between the base spaces of semi-analytic fibrations $\mathcal{X} \to \mathcal{Y}$. To simplify, we suppose the corresponding distributions $\mathcal{D}$ of vector fields and systems $\mathcal{S}$ of covector fiedls are integrable (or "very near" to one integrable). Even so, the topology of integrable distributions or systems is far from being trivial, and one must be careful with the validity of perturbations.

0.2.3. Data Analysis for Navigation

Along the first chapter B_{321} o this module a collection of sensors have been introduced to capture evolving information about a changing scene. Roughly speaking, the most relevant ones are labelled as image- or range-based sensors.

[9] Details in the two first chapters of B_{22} (Three-Dimensional Reconstruction)

The former ones have a "passive" character (conventional video cameras, e.g.), whereas the latest ones have an "active" character (infrared, laser, lidar, acoustic, RFID, etc), i.e. sensors capture a response of the environment w.r.t. the emitted signal. Evovling data are stored in terms of multivariable time series.

A non-trivial problem is how integrate all this information in a common framework. As all of them are given by different types of signals, the most immediate answer would be in some variant of the $2D$ or $3D$ Fourier domain. However, their interpretation is very difficult, data structures in the frequency domain are difficult to manage. Indeed, discrimination capability is very low and their real-time updating in regard to motion issues display a lot of unsolved problems, still. Roughly speaking, we prior different data structures in terms of lists for P, tables for C, connected lists for W, and simulation of their overlapping for their joint management in A.

Thus, to ease the visualization and interaction, we reconvert frequency-domain information a *space-time domain* in $\mathcal{P}$ which is nearer to human visual perception of an evolving environment. This choice implies to reinforce the role of geometrical information for "features" in $C(t)$, and their "translation" to topological representations for objects and scenes in $W(t)$. The functional approach is recovered again in regard to Decision Making strategies in $A(t)$ for near-optimal action to be performed.

Multivariable time series are the key for the data treatment for all PeCWA spaces appearing in the Basic Aanlytical Pipeline (BAP). At raw data level, a general strategy is based on ARMA (Auto-Regressive Mean Average) models and its variants [10]. Their joint management uses perturbation methods applied to matrix representations. To simplify, we restrict ourselves to basic models of *parametric statistics* given by multivariate normal distributions. In this way, one obtains more robust models for TIT (Topological Information Theory), before extending it to KIT (Kinematic Information Theory).

Distributions $\mathcal{D}$ of r vector fields corresponding to structural elements provide a support for self-organization, which must be compatible with "occasional" elements corresponding to "events". Their joint management can be visualized as a r'-dimensional flow, with $r' \leq r$ to be "integrated" (to ease its interpretation). In the simplest linear cases, one selects the "most relevant" components (by using SVD or PCA).

(*) In absence of reliable information, different relevance is managed in terms of ImpSaC (Importance Sampling Consensus) strategies for distributions which have been introduced in the module B_{23} (Motion Analysis) of B_2 (Computer vision). A "statistical relaxation" of this method is MLESac (Maximum Likelihood Estimation Sampling Consensus). Their computational management will be performed in symbolic terms by using Petri nets; see the section 4 of the chapter $B324$ for more details.

In a complementary way, systems $\mathcal{S}$ of covectors (given by differential forms

[10] A more structured approach for packaged data will be performed by using different variants of Kalman filters in the chapteer 6 of B_{33} (Computational Kinematics).

in thee PS context) provide the support for evolving constraints. In a similar way to the precedent case, the simultaneous management of the above s constraints can be thought as en "evaluation" of the volume form corresponding to their formal product (the wedge product in the PS framework). Ideal fluids are characterized by the preservation of volume forms (Liouville).

(*) In particular, if $S_i := supp(\omega_i)$ is the support of a constraint (visualizable as a one-form differential ω_i), successive constraints are set-theoretically supported by $\cap_{i=1}^{k} S_i$; in the PS framework, it can be interpreted as the dual of the exterior products $\omega_1 \wedge \ldots \wedge \omega_k$. Similarly to the case of distributions of vectors, one can apply ImpSaC and MLESaC methods for their estimation.

0.2.4. Joint motion of clusters

The approach performed at the end of the precedent paragraph has an ideal character. usually, one has not an accurate representation of vector and/or covector fields. The information flow for each PeCWA space appearing in the BAP (Basic Aanlytica Pipeline) displays a lot of irregularities involving different kinds of signals in the Perception space P, optical flow in Configurations space C, and scene flow in the Working space W. All of them lower the reliability of the decision to be made in the Action space A.

It is necessary minimize noise in P, remove outliers in C, and improve the robustness of packaged information in W. All these issues involve to individual clustering for each type of signal in P, grouping criteria in C, shape and behaviour characterization in W, and the corresponding remapping on A under constraints involving the mobile device and the scene.

Individual motion planning for once a mobile device in structured scenes is a classical topic which is well understood from the eighties; a good reference is [Lat91] [11] The increasing availability of low-cost mobile robotic devices poses several challenges involving groups composed by k agents. To fix ideas, we limit ourselves to *collaborative models*, where different agents share their own information to achieve a better understanding of themselves, their environment and actions to be undertaken. Some commonly used strategies to improve th information for each PeCWA space are the following ones:

- *Interpolation* of signals in P. Structural models for expected eigenvectors are crucial to obtain more robust models.
- *PL-propagation under uncertainty* of initial configurations in C from the current localization. The use of conoids for each cluster with similar motion characteristics provides natural bounds.
- *completion of information* relative to objects or bahaviours in the Working psace by using enhancement or restoration techniques under partial information.

[11] J.Latombe: *Robot Motion Planning* (2nd ed), Kluwer, 1991.

- *Expert systems* to assist *Decision Making* based on systems of logical rules, acccording to the usual hierarchy for logical rules (logic of classes, propositional, descriptive).

In more advanced settings one must introduce competitive behaviours instead of looking only to the collaborative ones. In both cases, one can have a “predominant” leader L_i for each cluster, and several followers F_{ij} (with similar behaviour to its leader L_i). Their identification includes the development of clustering strategies by following evolving patterns. From a more practical viewpoint, it is necessary to work at different levels going from automatic identification of coarse regions (by using windowing and/or sector decomposition of images, according to coarse motion characteristics) till the estimation of kinematic features of each evolving cluster.

As always, the hardest problem concerns to the design of a semi-automatic strategy for learning, based on “enough” cases-of-use. To lower the cost of training the corresponding Neural Networks, it is convenient to develop software tools for interactive simulation of scenarios with multiple agents and their corresponding behaviours. These issues will be developed in the chapter 6 of this module.

0.3. References for this introduction

References introduced here are not exhaustive nor the most recent ones. They must be understood as an invitation to the reader to acquire a deeper understanding of the subjects which have been sketched above. Each reader must be able of constructing his/her own knowledge representation according to the specific problem to be solved.

0.3.1. Basic references

We include only classical textbooks. More detailed references can be found at the end of the chapter. As suaul, each chapter ends with a fifth section devoted to recapitulation including conclusions, practices, challenges and more detailed references.

[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.

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

0.3.2. Software resources

In deterministic frameworks, the $A*$ provides the first robust algorithm, with efficient implementations at the end of xisties and early seventies. However, these hypotheses are not realistic for usual environments where mobile robots must operate.

An efficient implementation of solvers for Multiple Traveling Salesmen Problem (MTSP) is necessary for k agents operating in a common environment.

Low-dimensional problems can be solved with grid-based algorithms that overlay a grid on top of configuration space, or geometric algorithms that compute the shape and connectivity of Cfree.

Exact motion planning for high-dimensional systems under complex constraints is computationally intractable. Potential-field algorithms are efficient, but fall prey to local minima (an exception is the harmonic potential fields). Sampling-based algorithms avoid the problem of local minima, and solve many problems quite quickly. They are unable to determine that no path exists, but they have a probability of failure that decreases to zero as more time is spent.

Sampling-based algorithms are currently considered state-of-the-art for motion planning in high-dimensional spaces, and have been applied to problems which have dozens or even hundreds of dimensions (robotic manipulators, biological molecules, animated digital characters, and legged robots).

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