Contributions aux schémas d'Analyse/Synthèse en Animation par ordinateur

Nicolas Courty

HABILITATION A DIRIGER DES RECHERCHES
UNIVERSITÉ DE BRETAGNE SUD
UFR Sciences et Sciences de l'Ingénieur
sous le sceau de l'Université Européenne de Bretagne

Pour obtenir le grade de :
DOCTEUR DE L'UNIVERSITÉ DE BRETAGNE SUD
Mention : Informatique
École Doctorale SICMA

présentée par

Nicolas Courty

IRISA
Institut de Recherche en Informatique et Systèmes Aléatoires

Contributions to Analysis/Synthesis Schemes in Computer Animation

Habilitation soutenue le 19 Avril 2013,
devant la commission d'examen composée de :

Multon. Franck, Professeur
Université de Rennes 2 / Président

Badler. Norman, Professeur
Université de Pennsylvania / Rapporteur

Boulic. Ronan, Maître d'Enseignement et de Recherche
EPFL / Rapporteur

Horaud. Radu Patrice, Directeur de Recherche
INRIA / Rapporteur

Arnaldi. Bruno, Professeur
INSA de Rennes / Examinateur

Gibet. Sylvie, Professeur
Université de Bretagne Sud / Examinateur

"Les gens comprennent tous l'utilité de ce qui est utile,
mais ils ignorent l'utilité de l'inutile."

Zhuang Zi, famous Taoist from the IV^{th} century.

Remerciements

A travers cette page, je souhaite remercier tous ceux qui ont participé, de près comme de loin, au travail présenté dans ce document. En premier lieu, J'ai été extrêmement honoré d'avoir pu réunir dans mon jury plusieurs personnalités scientifiques que j'apprécie à la fois pour la qualité de leurs travaux mais aussi pour les contacts humains que j'ai pu avoir avec eux, parfois depuis mon entrée dans la recherche. Merci à :

- Franck Multon, Professeur à l'Université de Rennes 2, d'avoir accepté de présider ce jury, et d'avoir aussi gentiment accepté de covoiturer sain et sauf une bonne partie de ses membres le jour de la soutenance,
- Norman Badler, Professeur, Université de Pennsylvanie, Ronan Boulic, Maître d'Enseignement et de Recherche à l'EPFL et Radu Patrice Horaud, Directeur de Recherche INRIA Grenoble, d'avoir accepté et trouvé le temps de rapporter mon travail de recherche,
- Bruno Arnaldi, Professeur à l'INSA de Rennes, d'avoir bien voulu jouer le rôle d'examinateur, et d'avoir été le premier à me faire confiance en me permettant d'effectuer ma thèse dans son équipe (Siames),
- Sylvie Gibet, Professeur, à l'Université de Bretagne Sud, pour m'avoir accepté dans son équipe (Samsara) et permis d'encadrer plusieurs thèses ensemble. J'ai toujours apprécié les nombreuses discussions scientifiques que nous avons eues ensemble, ainsi que la liberté qu'elle m'a accordée dès mes débuts au Valoria.

Ce mémoire d'habilitation comporte pour une grande part des travaux réalisés par plusieurs étudiants que j'ai eu le plaisir de co-encadrer en thèse. Merci à Alexis Héloir, Charly Awad, Pierre Allain et Thibaut LeNaour pour votre participation à ces travaux !

Ce travail a eu comme cadre le laboratoire morbihannais de recherche en Informatique Valoria, devenu Irisa depuis. Merci à toutes les personnes qui participent à la vie de ce laboratoire et à l'ambiance qui y règne : Fred 1 et 2, Yves, Pierre-François, Sébastien, Jean-François (la petite pomme de 17h !), Laetitia, Nicolas, François, Sylviane, Gildas, Franck, Dominique, Joel, Michèle, jeanne, mais aussi nos collègues mathématiciens et statisticiens : valérie, Jacques, Quansheng, evans ainsi que tous les autres !

Le travail de recherche est à mes yeux surtout un travail de collaboration: merci à toutes les personnes avec qui j'ai eu le plaisir de travailler ! Notamment : Elise Arnaud, Anne Cuzol, Nicolas Lebihan, Pierre Hellier, Soraia Musse, Thomas Burger, Antoni Jaume Y Capo, Baogang Hu, Antoine Fagette, Clément Creusot, mais cette liste n'est pas exhaustive !

Parmi ces collaborateurs, il y en a un que je côtoie depuis les cours de Paul Sablonnière à l'INSA : thomas Corpetti ! Merci à toi vieux crabe pour m'avoir accueilli en Chine, une expérience inoubliable, et pour tous les cartes sur table, qu'on dédicacera forcément à qui tu sais.

Merci aux moustaches Guillaume (et ta bonne bouteille de Médoc pour accompagner la patte d'agneau mongole) et Pierre (pour la nuit qu'on a passé ensemble dans la yourte...). Une tournée de bisous à tous les copains d'un peu partout (dans le désordre): CC et le barbu, Romain l'abbé, Filou et Titine, Vince et Sandrine (merci pour le miel, une pensée pour Bastogne), le patron de la menuiserie, Stouf, Chlo et Joa "Chipotte" Delville, Etienne 'Ragarou' et Marie, Samy 'semoule millénaire' et Assia, Didier et Angélique, Christian et Virgine, Mister Zogz, Jaï et Claudio, les Shifus (Pascal Zed', Cassissa, GongXin, JB, Antoine, et le reste de la team, les rois de ChaoYang-GongYuan), merci aux petites brochettes et à la QingDao. Merci à Hey! short rose. Merci et à bientôt les vagues de la Guérite et de Manegwen, ainsi que les chemins de la réserve de Séné. Merci aussi au Semi Auray-Vannes pour m'avoir cassé mes petites pattes arrières toutes ces années, un jour ma revanche viendra. Une spéciale dédicace à mon pote Xavier, si tu nous écoutes depuis la zonzon, on pense à toi vieux, dans ta petite chambre d'hôpital, on pense aussi à ta femme et à la

petite.

Finalement merci à Margoton d'avoir gardé le sourire pendant mes périodes autistes de deadline, et à mes deux petits chaboules Titouan et Eole.

Contents

Preamble

Context of this work This report is a summary of my previous ten years of research as an assistant professor et University of Bretagne Sud, in Vannes, France. Those years were mainly dedicated to the computer animation of character figures, and more specifically on the approaches that allow to add knowledges from data into the control schemes. Those techniques will be referred to as **data-driven** techniques.

My PhD Thesis *[1]* was concerned with the application of visual servoing in the context of computer animation, and more specifically with the control of a virtual camera and a virtual humanoid. After a postdoctorate in Brazil, where I worked with Professor Soraia Musse on the subject of crowd simulation, I was recruited at the end of 2004 in the Valoria laboratory from the University of Bretagne Sud as an assistant professor. I then participated to the creation of a new team, together with Professor Sylvie Gibet: SAMSARA (**S**ynthesis and **A**nalysis of **M**otions for the **S**imulation and **A**nimation of **R**ealistic **A**gents). Our research activities aimed at gesture modeling and human motion generation in a computer graphics context. Our works were based on an Analysis/Synthesis workflow. From some observations of human motion, our objectives were to identify common mechanisms in sensorimotor behavior and to build artificial entities endowed with similar abilities. Those objectives constituted new research perspectives for me, as far as they introduced the use of data – and their related measurements errors, in the control laws.

As an interesting applications of those methods, we focused on the problem of signing avatars, *i.e.* virtual humanoids capable of speaking Sign Languages (SL). This interest was motivated by Sylvie previous works on the subject. As it will be developed in this document, SL constitute a wealth of interesting problems, and not only from a computer animation point of view since it involves also linguistics components. This research axis was the theme of the PhD thesis of Alexis Héloir [2], co-supervised with Sylvie and Franck Multon (University of Rennes 2), which ended up in the beginning of 2008. In his thesis, Alexis tried to analyze motion capture data acquired on a real signer and use it to produce new animations produced by a virtual signer.

The ANR SIGNCOM project (2008–2011) was a good opportunity to follow up this work. The focus was given on the interactions between a virtual signer and a real human. Associated to the project, a corpus of data involving one hour of motion capture data of a real signer and the associated annotation was acquired. The motion data was containing both rigid data (associated to a skeleton) and non-rigid deformation data (associated to the face). The exploitation of this motion information together with the semantic knowledges contained in the annotation was the subject of the PhD of charly Awad [3], co-supervised with Sylvie.

Finally, the final evaluation phase with real signers highlighted some defects in the animation pipeline. Notably, precision issues in the contact and the execution of the gestures were proved to be an important factor of misunderstanding in the sentences. The reasons of those imperfections were twofold: *i)* capturing at the same time the hands, the body gestures and some facial markers at a reasonable frequency bring the motion capture technology to its current limit, and new acquisition methods and protocols should be designed *ii)* the adaptation of the motion to a virtual human geometry which usually has a different morphology is prone to errors, especially in the contacts information. This problem calls for dedicated methods that are the subject of the PhD thesis of Thibaut LeNaour, who started in 2010, and which is also co-supervised with Sylvie. Also, acquiring a new corpus of data with a much better precision is the subject of an ongoing project: SIGN3D.

My second axis of research was aimed at crowd simulation. The originality of our approach on this subject was to add data in the classical control laws. Contrary to the human motion, the crowd is usually a highly-deformable, non-rigid entity that allows, at least for high densities, an analogy with fluids. However, and oppositely to the classical mechanical simulation of fluids or granules, the individual pedestrians are motivated by their own objectives and their perception of the environment. As such, they constitute "self-propelled" particles with social behaviors. The idea of using data acquired from multiple sources of information (such as crowd videos) aims at leveraging the determinism of incomplete or too restrictive simulation models. We first focused,

together with Thomas Corpetti (CNRS), on medium to high densities crowds, where a fluid-like assumption can be reasonably taken. From real crowd videos, we extracted with computer vision techniques the apparent motion in the image (optical flow), and then use this motion information in a very simple crowd control scheme. Though satisfying, the results clearly indicated that the only apparent motion is not sufficient, and other types of information, like pedestrian density, would greatly help. Estimating those quantities from crowd videos is non-trivial, and it was a part of Pierre Allain PhD thesis [4], co-supervised with Thomas. Pierre also designed original control loops that integrated those data, and as such realized a full analysis/synthesis scheme for crowd animation.

Before going into more details, a brief introduction to analysis and synthesis schemes in computer animation will be presented. A summary of our contributions to the domain will be given, and the outline of the document will be described. In appendix A the notation conventions used throughout this document are given for reference. Also, a citation in italic (e.g. *[1]*) indicates a publication in which I am one of the co-authors.

Introduction

Analysis/Synthesis schemes in computer Animation In the computer graphics domain the terms "Computer Animation" refer to the possibilities of animating with a set of programs and algorithms some virtual and mostly geometrical representations of objects. A wide range of application of those methods is possible. If computer games and production of visual effects in the cinematography industry are the most obvious ones, the development of 3D technologies in our day to day life through more powerful computing architectures and/or mobile devices has drastically augmented the number of possible applications. Among others, let us cite the increasing interest to virtual avatars as good human computer interfaces. It is also interesting to consider the use of computer animation in the domain of simulation, where computer graphics is mostly used to visualize some possible behaviors of real objects embedded in a virtual representation. Hence, virtual reality, virtual prototyping and augmented reality constitute also good applications of computer animation methods.

Animating objects require to define the behavior of a virtual description of an object through space and time. Computer animation techniques, which technical foundations have emerged from mechanics, robotics and control theory, have somehow followed a path that might be considered as less noble in the sense that working in a perfect, exact and virtual space, where all the needed information is available without having to deal with noisy captors, may appear easier. Also, the required methods accuracies are less demanding, and possible simplification of difficult equations might be considered for the method to run at interactive frame rates. However, the last five years of research in this field have shown very nice examples of cross fertilization of this domain to others such as fluid mechanics or physical simulations. The reasons stem from the need of more and more realistic animations, which imposes a deeper understanding of the underlying physical mechanisms. Also, the computation skills required to adapt sophisticated simulations to interactive applications have also aroused other discipline's interests in the sense that more rapid (but accurate) methods allow to deploy their technique to a broader range of problems and help the inherent scalability. Hopefully, this discipline is starting to earn and to deserve a reputation for excellence, but the path is an arduous one.

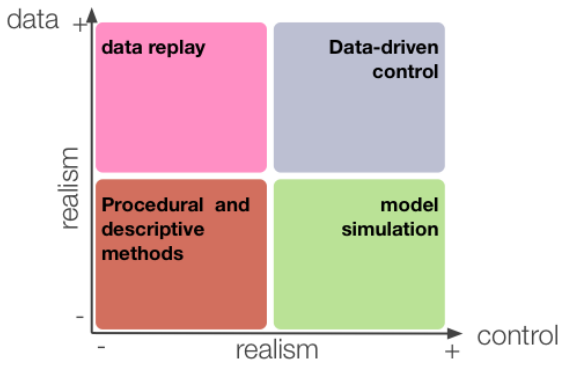


Figure 1.: Use of data versus controllability of an animation

Historically, the first animation methods can be considered as procedure driven, meaning that the space-time behavior of an object is fully described in a procedure or algorithm that directly describes or encodes the effects of the animation. Several drawbacks can be considered, especially since controlling or interacting with such systems is tedious, and the final realism is usually weak. An alternative lies in the use of (physical) model simulation, where the causes of the motion are described, instead of the effects. By **model**, we will denote the description (at an algorithmic or equational level) of the state evolution, usually subject to external forces or commands. Yet, the control of such systems is generally a difficult inverse problem (what are the causes needed

to obtain a desired effects ?), usually because of the highly non-linear transformations involved in the simulation process. **Data-driven** methods usually provide a good way to obtain realism with only a few downfalls; the idea is to be able to use some a priori knowledge about the motion (like kinematic trajectories) obtained from the observation of the corresponding phenomenon in the real world. However, three problems need to be addressed:

- **the acquisition problem**: what are the good descriptors for a motion ? Is it possible to capture them directly from the real world ?
- **the generalization problem**: how can one generalize the information contained in the observed examples ? How to characterize the extent of this generalization ?
- **the control problem**: how to use this knowledge into an animation method ? Is it possible to control in some ways the final animation ?

It turns out that those problems have to be considered dependent of each others. Hence, given an applicative context, specific constraints on the possibilities of control of the animation systems have to be considered. The simplest form of control will only consider the data playback, while more complex controllers will consider more complex combination or inferences over the data to generate potentially new trajectories, eventually fulfilling some users-defined constraints. We argue in the rest of the document that the combination of data and sophisticated control methods can bring the most realistic results, as depicted on Figure 1. At this point it can be interesting to give an example of a data-driven animation method. Let's assume that we want to animate the walk of a virtual character. A classical representation of this motion is described by a kinematic chain representing a simplification of the character skeleton. As such, this virtual skeleton is accounting for the rigid transformations involved in the walking motion. The acquisition of this information can be done by using motion capture technologies, which will infer the motion of the skeleton from a set of markers positioned over a real actor. Several walking motions can be acquired through this process. Obviously, a virtual character can be directly animated from on instance of these motions. The control problem can be translated in the following problem: if a user want to move this walking character throughout a virtual environment, how the animation process should choose among all the possible motions to reach the user's objective ? The simplest method will consider the direct combination of the original set of walking motions both in space and time to fulfill at most the user's objectives. The generalization issue then arises: is the current set of walking motions sufficient to cover each possible walking scenarios ? One could also seek, provided that the data is rich enough, to learn what walking means, both in terms of correlations between the different skeleton articulations, and also of dynamical information (how are those transformations evolving in time ?). Hopefully, the learnt model of what is a walk can be used to infer a new walking motion fulfilling the users requirements.

From a methodological point of view, it is possible to oppose the use of data and models (see Figure 2). The use of "pure" data may not require any model, and conversely a "pure" model will not necessitate any data. The combination of data to produce new data can be very simple. As an example, a weighted linear combination of data can be considered, but is likely to produce unwanted results, at least if the input set of data is rather sparse. Eventually, the distribution of acceptable weights can be inferred from the data. This could be an example of use of a non-parametric statistical method. Parametric method can also be considered, if one makes some assumptions about the underlying process that generate the motion. As examples, one can cite Gaussian processes, which will make the assumption that the underlying generative process of the data is Gaussian, *i.e.* there is a statistical Gaussian dependency among the data. More complex models, eventually described by a set of partial derivative equations, can be completed by data. This is the case with variational assimilation methods. Those levels of modeling have the good property of being able to handle various levels of uncertainties, either in the description of the data or in the description of the models. Finally, if one is very confident in a given model, the available data can be efficiently used to infer some parameters of the model. This problem is known as model identification.

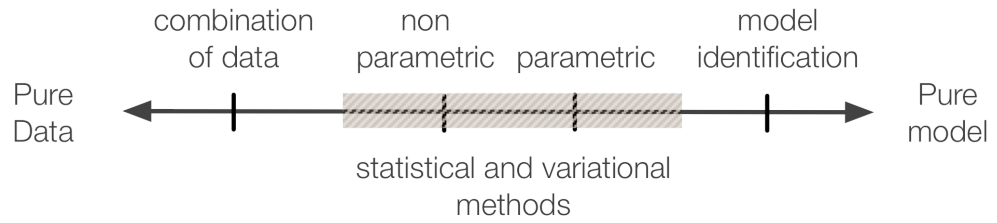


Figure 2.: Opposing data and model

Backing to the acquisition problem, the problem of estimating the needed data from real situations is not always trivial. As an example, assume we want to obtain the walking motions from videos of walking humans. Obviously, this problem is harder than capturing the walks within a motion capture setting, but the involved material is much less costly and will not require controlled acquisition conditions. In the computer vision community, this problem will be referred to as articulated figure tracking. As the problem is ill-posed, since in the image plan only a projection of the original motion is available, assumptions have to be made about the nature of what is observed. Eventually this a priori knowledge can be obtained from some data, some models or a combination of the two. Finally, we see that in a data-driven approach, the problem of estimating some data is in fact closely linked to the problem of the generation of those data. We will refer to this two different parts as the **Analysis** and **Synthesis** processes. In the first one, the objectives are to estimate from a given source of information relevant quantities, and in the second to produce or infer them. Those two processes form a loop in the sense that the data provided by the Analysis can used in the definition of some models, and because those models can be used in the estimation of the data. In fact, it is a virtuous circle, as described in Figure 3, since better data will lead to better models, and better models will help in acquiring better data, and so on.

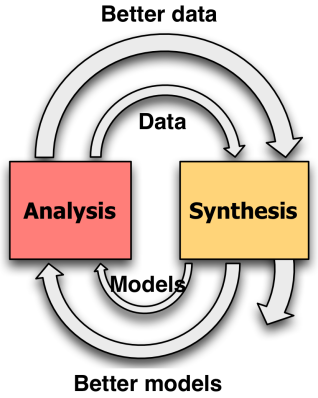


Figure 3.: The analysis/synthesis virtuous circle

Contributions of this work and document articulation The work reported in this document is a contribution in the definition of operational analysis/synthesis schemes for computer animation. We have considered two different application contexts: the animation of a virtual avatar, and the simulation of crowd phenomena, both with data acquired from the real world. This leads naturally to two distinct parts in this document. The first one will be devoted to character animation. It is composed of the following chapters:

- **Chapter 1** will describe the context of my work on human character animation. After a general introduction on the problem of articulated character animation, the specificity of the animation of a virtual signer will be described,
- **Chapter 2** is concerned with the representation of a motion, and its analysis. A particular focus will be given on the non-linear nature of this type of data,
- the control loops using this motion information will be described in **Chapter 3**. Three different types of control will be presented, with a varying degree of dependency to the motions,
- and finally the last chapter of this part will present on-going works and possible future directions.

The second part of the document is dedicated to crowds. It is composed of the following chapters:

- **Chapter 5** presents our motivation for handling the crowd simulation problem with an analysis/synthesis approach,
- **Chapter 6** shows some contribution in the analysis of crowd videos and the extraction of relevant parameters for the animation process, while
- **Chapter 7** describes a complete methodology to control crowd simulations, eventually through the use of data extracted from videos.
- this part is then concluded and perspectives will be given.

A final conclusion and a global summary of our contributions will end this document.

Part I.

Data-driven animation of human characters

1

Preamble and application context

Contents

This part of the document is dedicated to the human character animation. We give in this Chapter the application context of this study, which is devoted to the animation of a virtual signer. However, from a methodological point of view, our work has broader applications, and some of the methodologies developed in the following chapters are not only dedicated to the production of signs and expressive gestures. Therefore we will start by giving a rapid overview of problems and challenges related to human character animation.

1.1. Virtual characters: existing work and challenges

We give in this Section a short overview of the common problems encountered in human character animation. Clearly, this Section is not intended to give an exhaustive state-of-the-art, and we refer interested readers to the following book [5] for a general overview of the associated problems. Also, we will mainly focus on data-driven approaches.

Animating articulated human figure is a difficult task [5], for mainly two reasons: *i)* the human body is a complex mechanism, made of hundred of bones and muscles, with a centralized control system (the brain). The links between the perceptions and the motor system turns out to form a complex loop which is more or less well explained by physiologists and cognitive scientists *ii)* our human perception system is highly trained and used to what is a human motion, and detect easily unusual or abnormal behaviors. Hence creating realistic and plausible motions remains an open challenge within the animation community. Pure synthesis models like inverse kinematics [6, 7, 8] have been well studied in the robotics and computer animation literature. Usually, the corresponding inverse problems are underdetermined, and benefit from adding constraints in the production of motions [9, 10, 11, 12]. This stands for an interesting way to achieve more realism as far as the added constraints are carefully chosen.

Yet, the use of motion capture provides a clear advantage over the previous methods in the credibility and realism of the corresponding animation. The underlying principles are very simple: instead of trying to understand and model the motion production system, the desired motion can be directly measured on a real actor. Up to now, this motion is represented by a collection of moving points located over the body of the actor, and generally yields a skeleton-based representation of the motion. This skeleton then drives a more complex character geometry. Generally, the captured

motion is very specific to a given situation. However, and mostly because acquiring a motion is a costly and time consuming operation, one could seek to reuse this motion in different settings. This opens a wide range of problems:

- how to adapt a motion to a new situation ? For example, if the captured motion corresponds to one person opening a door, how to change the door-handle position ? This problem can be referred to as the **motion editing** problem. It can be include a combination of kinematic and temporal constraints (**space-time constraints**),
- how to adapt a motion to a different morphology ? If the actor is tall and fat, how can we adapt the motion to a small and skinny virtual character ? This problem is known as the **motion retargetting** problem.

Obviously, those problems should be solved without eliminating the naturalness of the motion which makes motion capture such an appealing tool for computer animation. Existing solutions for the creation of a new motion can be decomposed into two categories: interpolative and generative. The first category refers to methods combining (generally in a linear way) existing motions, whereas the second deals with learned models of motions:

- *Motion combination.* Within this category of methods, a new motion is produced as a combination of existing motions. The weights associated to this combination can be derived so as to minimize an energy function related to a new constraint [13, 14]. The blended motions have usually to share the same temporal variations between the different occurrences. When it is not the case, a temporal alignment has to be performed, in most cases by relying on a well known dynamical programing method: dynamic time warping (DTW) [15]. The combination can also be temporal, *i.e.* a concatenation in time of small chunks of motions extracted from several motions. This is the case in the motion graph approach [16, 17, 18], where a graph over all the possible poses (nodes of the graph) of all the available motions is built. Then, at runtime, a new motion is produced as a path optimization on this graph,
- *Statistical approaches.* In the absence of physical or analytical models of motions, statistical models have the capability of expressing the knowledge available in the data, and have revealed over the last years to be a tool of choice for enclosing the motion specific information [19, 20, 21, 22, 23, 24]. The seminal work of Pullen and Bregler [25] is the first to use a non parametric multivariate probability density model to express the dependencies between joint angles in motions. Samples drawn from these distributions are then used to generate new sequences from an input motion. Non parametric models have also been used more recently to handle the variation synthesis problem [24], where Lau *et al.* use dynamic Bayesian networks to both handle spatial and temporal variations. However, most of existing works concentrate on parametric families of statistical models. In [26], Brand and Hertzmann were the first to model a motion with hidden Markov models. The motion texture paradigm [20] uses a two level statistical model, where short sequence of motions (textons) are modeled as linear dynamic system along with a probability distribution of transitions between them. Chai and Hodgins [27, 19] also use linear time invariant models such as autoregressive models to model the dynamic information in the motions. Gaussian processes first served in the computer animation community to perform dimensionality reduction and construct a latent variable model [28].Gaussian processes have been also widely used in the context of computer vision [29]. In [22], Wang and colleagues extended the latent space formulation with a model of dynamics in the latent space. Most recent applications of Gaussian processes include motion editing [23] and style-content separation [30].

The motion is a signal information which generally conveys a meaning or an intention. This can be encoded as a textual information which completes the database. These annotations can be obtained through a manual or an automatic process. In both cases the problem faced when annotating is the temporal segmentation problem. Temporal segmentation consists in cutting motions into groups of consecutive frames, but the problem remains in detecting accurately start and end frames. In [31], Arikan et al. use a semi-supervised annotation scheme: after annotating

manually a small portion of the database, unclassified motions are annotated thanks to an SVM classifier. Chao and al. also used an annotated motion database of Tai Chi Chuan moves in [32]. Most of the time, using large database of motions will impose to have efficient storing and retrieval methods [33, 34, 35, 36]

We now turn to the more specific problems of animating a communicative virtual character.

1.2. Virtual Signers

Signed languages (SL), defined as visual languages, were initially intended to be a mean of communication between deaf people. They are entirely based on motions and have no written equivalent. They constitute full natural languages, driven by their own linguistic structure. Accounting for the difficulties of deaf to read text or subtitles on computers or personal devices, computer animations of sign language improve the accessibility of those media to these users [37, 38, 39, 40]. The use of avatars to this purpose allows to go further the restrictions of videos, mostly because the possibilities of content creation with avatars are far more advanced, and because avatars can be personalized along with the user's will. They also allow the anonymity of the interlocutor.

However, animating virtual signers has revealed to be a tedious task [41], mostly for two reasons: *i)* our comprehension of the linguistic mechanisms of signed languages are still not fully achieved, and computational linguistic software may sometimes fail in modeling particular aspects of SL *ii)* animation methodologies are challenged by the complex nature of gestures involved in signed communication. Our research were mainly concerned by this second class of problems, even though we admit that in some sense those two aspects are indissociable.

In fact, signs differ sensibly from other non-linguistic gestures, as they are by essence multichannel. Each channel of a single sign (those being the gestures of the two arms and the two hands, the signer's facial expressions and gaze direction) conveys meaningful information from the phonological level to the discourse level. Moreover, signs exhibit a highly spatial and temporal variability that can serve as syntactic modifiers of aspect, participants, etc. Then, the combination in space and time of two or more signs is also possible and sometimes mandatory to express concisely ideas or concepts. This intricate nature is difficult to handle with classical animation methods, that most of the time focus on particular types of motions (walk, kicks, etc.) that do not exhibit a comparable variability and subtleties.

We recognize in this problematic a natural application of the analysis/synthesis schemes: we first begin to show existing approaches that are based on procedural methods, and show why it could be interesting to add data to the production of signs. Notably, we will describe in details the animation challenges associated to the conception of a virtual signer.

1.2.1. Existing Virtual Signers

We first begin by reviewing some of the technologies used to animate virtual signers. Figure 4 presents in chronological order some existing virtual signers.

Several gesture taxonomies have already been proposed in [45] and [46], some of which rely on the identification of specific phases that appear in co-verbal gestures and sign language signs [47]. Recent studies dedicated to expressive gesture rely on the segmentation and annotation of gestures to characterize the spatial structure of a sign sequence, and on transcribing and modeling gestures with the goal of later re-synthesis [48].

Studies on sign languages formed early description/transcription systems, such as [49] or [50]. More recently, at the intersection of linguistics and computation, gestures have been described with methods ranging from formalized scripts to a dedicated gestural language. The BEAT system [51], as one of the first systems to describe the desired behaviors of virtual agents, uses textual input to build linguistic features of gestures to be generated and then synchronized with speech. Gibet

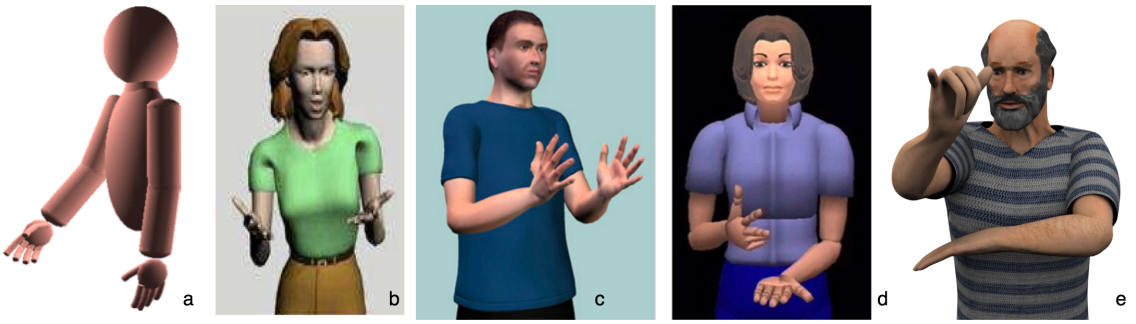


Figure 4.: Some virtual signers classified in chronological order: (a) the GESSYCA system [42] (b) Elsi [43] (c) Guido from the eSign european project [39] (d) the virtual signer of the City University of New-Yord [41] (e) Gerard [44]

et al. [42] propose a gesture synthesis system based on a quantified description of the space around the signer; using the HamNoSys [50] sign language notation system as a base, the eSign project has further designed a motion specification language called SigML [52]. Other *XML*-based description languages have been developed to describe various multimodal behaviors, some of these languages are dedicated to conversational agents behaviors, as for example MURML [53], or describe style variations in gesturing and speech [54], or expressive gestures [55]. More recently, a unified framework, containing several abstraction levels has been defined and has led to the XML-based language called BML [56], which interprets a planned multimodal behavior into a realized behavior, and may integrate different planning and control systems.

Passing from the specification of gestures to their generation has given rise to a few works. Largely, they desire to translate a gestural description, expressed in any of the above-mentioned formalisms, into a sequence of gestural commands that can be directly interpreted by a real-time animation engine. Most of these works concern pure synthesis methods, for instance by computing postures from specification of goals in the 3D-space, using inverse kinematics techniques, such as in [42], [57], [58]. Another approach uses annotated videos of human behaviors to synchronize speech and gestures and a statistical model to extract specific gestural profiles; from a textual input, a generation process then produces a gestural script which is interpreted by a motion simulation engine [59].

Alternatively, data-driven animation methods can be substituted for these pure synthesis methods. In this case the motions of a real signer are captured with different combinations of motion capture techniques. Since it is not possible to record every possible sentences, new strategies are to be devised in order to produce new utterances, The next paragraph presents an example of a fully data-driven approach.

1.2.2. An example of a full data-driven approach: the Signcom project

An example of a full data-driven virtual signer is given by the Signcom project, which aims at improving the quality of the real-time interaction between real humans and avatars, by exploiting natural communication modalities such as gestures, facial expressions and gaze direction. Based on French Sign Language (FSL) gestures, the real human and the virtual character produce statements towards their interlocutor through a dialog model. The final objective of the project consists in elaborating new ways of communication by recognizing FSL utterances, and synthesizing adequate responses with a 3D avatar. The motion capture system uses Vicon MX infrared camera technology to capture the movements of our LSF informants at frame rates of 100 Hz. The setup was as follows: 12 motion capture cameras, 43 facial markers, 43 body markers, and 12 hand markers. In order to replay a complete animation, several post operations are necessary. First, the fingers' motion were reconstructed by inverse kinematics, since only the fingers' end positions were recorded. In order

Figure 5.: Photo of the motion capture settings in the Signcom project

to animate the face, cross-mapping of facial motion capture data and blendshapes parameters was performed [60]. This technique allows to animate directly the face from the raw motion capture data once a mapping has been learned. Finally, since no eye gazes were recorded during the informants performance, an automatic eye gazing systems was designed. Figure 6 gives some illustrations of the final virtual signer "sally" replaying captured motions. A corpus annotation was also conducted. Annotations expand on the mocap data by identifying each sign type with a unique gloss, so that each token of a single type can be easily compared. Other annotations include grammatical and phonological descriptions.

From recorded FSL sequences, multichannel data are retrieved from a dual-representation indexed database (annotation and mocap data), and used to generate new FSL utterances [44], in a way similar to [61]. The final system has been evaluated with native LSF signers *[62]* .

1.3. Challenges in Sign production

Though data-driven animation methods significantly improve the quality and credibility of animations, there are nonetheless several challenges to the reuse of motion capture data in the production of sign languages. Some of them are presented in the following.

Spatialization of the content As sign languages are by nature spatial languages, forming sign strings requires a signer to understand a set of highly spatial and temporal grammatical rules and inflection processes unique to a sign language. We can separate plain signs that do not use space semantically (like the American Sign Language sign HAVE which does not make any notable use of space other than which is necessary for any sign) from signs that incorporate depiction. This second group of signs includes the strongly iconic signs known as depicting verbs (or classifiers), which mimic spatial movements, as well as size-and-shape specifiers, which concern static spatial descriptions.

Moreover, indicating signs like indicating verbs and deictic expressions require the signer to interface with targets in the signing space by effecting pointing-like movements towards these targets. Indicating verbs include such signs as the LSF sign INVITER, in which the hand moves from the area around the invited party toward the entity who did the inviting . Depending on the intended subject and object, the initial and final placements of the hand vary greatly within the signing space. Deixis, such as pronouns, locatives, and other indexical signs are often formed with a pointed index finger moving toward a specific referent, though other hand configurations have been reported in sign languages, such as American Sign Language.

Figure 6.: Screenshots of the virtual signer "Sally" from the Signcom project

Small variations can make big semantic differences Sign languages require precision and rapidity in their execution, but at the same times imperfection in the realization of the signs or bad synchronization can change the semantic content of the sentence. We give here some challenging elements in the execution of signs:

- **Motion precision.** The understandability of signs require accuracy in the realization of the gestures. In particular in finger spelling the degree of openness of a fingers leads to different letters. Some of fhe different hand shapes used in FSL only differ by the positions of one finger or by the absence or not of a contact. This calls for a great accuracy in the capture and animation processes.
- **spatio-temporal aspects of the gestures.** The sign language being a language with highly spatio-temporal components, the question of timing and dynamics of gesture is crucial. In fact, three elements are of interest for a sign: first, the spatial trajectory of the hands are rather important. They do not only constitute transitions in space between two key positions, but may be constituent of the sign. This raises the problem of the coding of this trajectory. Second, synchronization of the two hands is a major component, and usually hands do not have to this regard a symmetric role, In the case of PAS D'ACCORD (not agree), the index start from from the forehead and meets the other index in front of the signer. The motion of the second hand is clearly synchronized on the first hand. Third, the dynamics of the gesture (acceleration profile along time) allows the distinction between two significations. An example is the difference between the signs CHAISE (chair) and S'ASSEOIR (to sit), which have the same hands configurations, the same trajectories in space, but different dynamics. Let us finally note that the dynamics of contacts between the hand and the body (gently touching or striking)is also relevant.
- **facial expressions and non manual elements.** While most of the description focus on the hands configuration and their motions, important non manual elements should also be taken into account, like shoulder motions, head swinging, changes in gazes or facial mimics. For example, the gaze can be used either to recall a particular object of the signing space, or either directed by the dominating hand (like in the sign LIRE, to read, where the eyes

follow the motion of fingers). In the case of facial mimics, some facial expressions may serve as adjectives (for instance inflated cheeks will make an object big, while wrinkled eyes would make it thin) or indicate wether the sentence is a question (raised eyebrows) or an affirmation (frowning). It is therefore very important to preserve these informations in the facial animation.

2

Human motion and its analysis

Contents

Using captured motion data has now become very popular in the animation community, thanks to the quality and the liveliness of the produced motions. Usually, a motion can be described as a combination of a translation and several rotations of a hierarchical structure that can be assimilated to a virtual skeleton. Hence, this type of information does not necessary belong to a linear space, and special cares have to be taken with respect to the nature of the data. This is the key idea of this Chapter, where we try to show how this specific nature can be handled with dedicated algorithms. After a description of the nature of the motion information, a contribution in the domain of the processing of rotation data is described (Section 2.1). We then show how to extract interesting knowledges from the raw motion information, either by dimensionality reduction (Section 2.2) or by learning the temporal dynamics of the data (Section 2.3).

2.1. Motion representation and filtering

Acquiring motion data can be done in various settings, ranging from optical devices to inertial or magnetic systems. Optical devices with passive or active markers usually show the best accuracy with high frequencies capture but at the expense of wearing special suits. Markerless motion capture [63, 64], based on vision algorithms, offers an interesting alternative that imposes less constraints, but yet has only been partially solved. Other equipments, such as inertial, mechanics or magnetic systems suffer from calibration problems and can encounter drifting issues over time, but are available at much lower prices. New methods based on communicating wearable sensors such as Prakash [65] are very promising and brings to life the possibilities of *on-set* motion capture systems. All these acquisition methods usually provide in the end the same type of information. We will make no assumption about the nature of the acquisition device in the remainder.

2.1.1. Motion representation

A motion $\mathbf{M}$ can be represented as a time series of rotation vectors, each rotation representing a rotation of a particular joint in an articulated figure. Those rotations are now frequently defined as unit quaternions [66], since they have proved to be relatively compact and efficient. Let us recall for clarity some facts about representing rotations with unit quaternions. The quaternion space $\mathbb{H}$ is spanned by a real axis and three imaginary axis $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ under Hamilton's conventions. A quaternion $\mathbf{q}$ is a 4-uple of real values (w, x, yz). Unit quaternions ($\|\mathbf{q}\| = 1$) can be used to parameterize rotations in $\mathbb{R}^3$, and can be considered as a point on the unit hyper-sphere S^3. As shown by Euler, any rotation map $\in SO(3)$ can be represented by an angle θ around an arbitrary axis $\mathbf{v}$. This leads to an intuitive representation of the quaternion as an ordered pair of a real and a vector, *i.e.* $\mathbf{q} = (w, \mathbf{a})$ with $w = \cos\frac{\theta}{2}$ and $\mathbf{a} = \sin\frac{\theta}{2}\mathbf{v}$. The multiplications of two quaternions is defined but not commutative, *i.e.* $\mathbf{q_1 q_2} \neq \mathbf{q_2 q_1}$. It is possible to define a distance metric between two quaternions in S^3. This metric is set as the length of the *geodesic* path between two elements on the hypersphere. This distance is given by:

$$Dist(\mathbf{q_1}, \mathbf{q_2}) = \left\|\log(\mathbf{q_1^{-1} q_2})\right\| \tag{2.1}$$

An alternative metric can be defined as suggested in [67] by the inner product of two quaternions:

$$Dist(\mathbf{q_1}, \mathbf{q_2}) = \mathbf{q_1}.\mathbf{q_2} = w_1 w_2 + x_1 x_2 + y_1 y_2 + z_1 z_2 \tag{2.2}$$

and has the advantage of being proportional to the length of the geodesic path and fast to compute. A weighted formulation of this distance is given by:

$$weightedDist(\mathbf{q_1}, \mathbf{q_2}) = w_r(1 - \|\mathbf{q_1}.\mathbf{q_2}\|) \tag{2.3}$$

which gives a weighted rotation distance between two quaternions $\in [0, w_r]$.

The definition of a motion with quaternions is finally:

$$\mathbf{M} = \{\mathbf{q_i}(\mathbf{t}) | i \in [0 \cdots n], \quad t \in [0 \cdots m]\} \tag{2.4}$$

where n is the number of quaternions used to represent a posture of the skeleton, and m the number of postures in the motion. Hence, $\mathbf{M}$ is an element of $\mathbb{H}^{n \times m}$

2.1.2. Motion filtering

The problem of motion filtering arises when the acquisition methods provide noisy data. Classical de-noising methods are based on local operators that smooth the input signal (such as a Gaussian blur or a Butterworth filter which are commonly used by animators) or on subspace techniques such as PCA [68] (and its variations) that seeks to preserve the principal features of the motion.

Though, human motion have specific features that need to be taken into account. For instance, communicative gestures such as non-verbal communication gestures are characterized by rapid and subtle changes that influence greatly the perceived meaning of the gesture *[69]* . These high frequency information produce subtle details that human beings are able to interpret and decrypt. It is thus of primary interest to be able to preserve those aspects while canceling the inherent noise of the capture system. We have proposed in *[70]* an adaptation of the well-known bilateral filter to rotation data, thus making it suitable to treat human motion. We argue that for the de-noising purpose, the bilateral filter tends to preserve some characteristic features of human motion such as rapid changes in the velocity profile.

Bilateral filtering is a well-known technique in signal and image processing. First introduced with its current name by Tomasi *et al.* [71], it has been used in several contexts such as image denoising [72, 73], computational photography [74, 75, 76, 77], stylization [78], optical flow computation [79] or even biomedical imaging [80]. Several theoritical studies have revealed its intrinsic

nature and limitations [81, 82]. In the context of image denoising, it has been shown can be related to the classical PDEs such as heat diffusion or the Perona-Malik equation [82]. The reasons for its success are its simplicity to design (it usually implies a local averaging scheme) and to parameterize (only the spatial extent and the contrast preserving strength are required). Bilateral filter has also been extended to other types of data. In [83, 84], it is used as a smoothing operator for $3D$ meshes. In [85], Paris and colleagues have used bilateral filtering to smooth a $2D$ orientation field by incorporating a mapping into the complex plane $\mathbb{C}$. In this sense, their works can be related to our technique. To the best of our knowledge, no existing method uses an adaptation of the bilateral filtering to manifold-valued signal such as $3D$ rotation time series.

2.1.2.1. Filtering orientation data

Filtering rotation data is a difficult problem that comes from the non-linearity of the unit quaternion space. Let $\mathbf{X} = \{x_i\}$ be a signal with elements in $\mathbb{R}^n$. The classical convolution operation with a filter mask $(m_{-k}, \ldots, m_k)$ of size $2k+1$ gives the following filter response at the ith element:

$$\mathbf{H}(x_i) = m_{-k}x_{i-k} + \ldots + m_0 x_0 + \ldots + m_k x_{i+k}$$

This operation does not transpose to a signal $\mathbf{Y} = \{\mathbf{q}_i\}$ of elements in $SO(3)$ because the addition is not correctly defined for two unit quaternions since the result is no longer a unit quaternion. A possible solution would be to consider the embedding space $\mathbb{R}^4$, perform computation on quaternions such as vectors of this linear space and then re-normalize the result. Though, this solution can lead to strange behaviors when data are not sufficiently dense enough [86]. Another solution considers a global linearization of the input signal [87], by using for instance the exponential mapping between S^3 and $\mathbb{R}^3$. This method also suffers from problems since there is no such global mapping (e.g. the exponential mapping is ill-defined at the antipode of the identity quaternion), and therefore some singularities may corrupt the result. The concept of local linearization was first used by Fang and colleagues [88]. It consists in decomposing the input signal into a succession of linear displacements between each consecutive samples, filter those displacements, and finally construct the filtered signal by integrating those displacements. As pointed out in [86], this integration yields drifting problems over time. Lee and Shin [86] have proposed a filter design that avoids this problem. The key idea is to consider angular displacement between each samples as a linear displacement in $\mathbb{R}^3$, filter this vector counterparts and construct the signal back through exponentiation, thus avoiding the drifting problem induced by integration in the method of Fang et al. [88]. Moreover, they demonstrated that their construction protocol leads to a linear time-invariant class of filters (LTI filters). Their framework defines the output response of a filter $\mathbf{H}$ as:

$$\mathbf{H}(\mathbf{q}_i) = \mathbf{q}_i \exp\Big(\sum_{r=-k}^{k-1} b_r w_{i+r}\Big) \tag{2.5}$$

where

$$w_i = Log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})$$

are the local linearizations of the input signal, and b_r scalars derived from the traditional filter mask coefficients m_j and defined such that:

$$b_r = \begin{cases} \sum_{j=-k}^{r} m_j & if \quad -k \leq r < 0 \\ \sum_{j=r+1}^{k} m_j & if \quad 0 \leq r < k \end{cases} \tag{2.6}$$

This construction method has been chosen in order to design our bilateral orientation filter.

2.1.2.2. Bilateral motion filtering

We first begin this section by recalling the classical form of the traditional bilateral filter. We then present its adaptation to orientation data.

2.1.2.3. Bilateral filtering in a linear space

In its original form [71], the bilateral filter is a weighted average of a sample i of the original signal $\mathbf{X} = \{x_i\}$, given by:

$$\mathbf{BF}(x_i) = \frac{\sum_{r=-k}^{k} W(i,r)x_{i+r}}{\sum_{r=-k}^{k} W(i,r)} \tag{2.7}$$

where $W(i,r)$ are the weights of the filter and are given by a combination of functions of the temporal distance $W_t(i,r)$ and the geometric distance between the samples i and $i+r$: $W_g(i,r)$. W functions are smoothly decaying functions, usually Gaussian functions. In this case, $W(i,r)$ writes:

$$\begin{aligned} W(i,r) &= W_t(i,r) * W_g(i,r) \\ W_t(i,r) &= \exp(-\frac{d^2(i,i+r)}{2\sigma_t^2}) = \exp(-\frac{|r|}{2\sigma_t^2}) \\ W_g(i,r) &= \exp(-\frac{d^2(x_i,x_{i+r})}{2\sigma_g^2}) = \exp(-\frac{|x_{i+r}-x_i|^2)}{2\sigma_g^2}) \end{aligned}$$

The idea behind this definition is that both near samples and samples with close-by values will have more influence on the final result. σ_t and σ_g set their relative strength and are generally used to privilege one of these two aspects.

2.1.2.4. Bilateral filtering on rotation data

In order to adapt the bilateral filter to orientation data, we first need to choose a metric between rotations. It is common to use the length of the *geodesic* path between two elements on the hypersphere (geodesic distance). This choice is important as it conditions some of the filter properties (see below). This distance is given for two unit quaternions by:

$$d(\mathbf{q}_1, \mathbf{q}_2) = \left\|\log(\mathbf{q}_1^{-1}\mathbf{q}_2)\right\| \tag{2.8}$$

We now adopt the construction method of Lee and Shin [86] described in the previous section to build our filter. The m_j coefficients used in equation 2.6 are given by:

$$m_j = W(i,r) = \exp(-\frac{|r|}{2\sigma_t^2}) \exp(-\frac{\left\|\log(\mathbf{q}_i^{-1}\mathbf{q}_{i+r})\right\|)}{2\sigma_g^2})$$

Those coefficients characterize the Bilateral Orientation filter. They have to be computed with respect to a sliding window over the signal. As the distance between each samples of the signal has to be evaluated several times, it can be convenient to pre-compute all these distances as a band matrix $\mathbf{D}$ as depicted in Figure 7. The symmetry of the distance function allows to store only the upper-diagonal part of the matrix[1].

The filter construction presented in [86] guaranties that the filter is linear time-invariant. In our case, this proposition does not hold anymore since the filter coefficients depend on the input signal. Nevertheless, we demonstrate that our bilateral orientation filter keeps interesting properties:

Proposition 1 *The bilateral orientation filter is coordinate-invariant, that is to say that for any* $\mathbf{a}, \mathbf{b} \in S^3$, $\mathbf{BOF}(\mathbf{aq_ib}) = \mathbf{aBOF}(\mathbf{q}_i)\mathbf{b}$.

Proposition 2 *The bilateral orientation filter is time-invariant,*

[1] In this case, $\mathbf{D}(i,k)$ becomes $\mathbf{D}(k,i)$ if $i > k$.

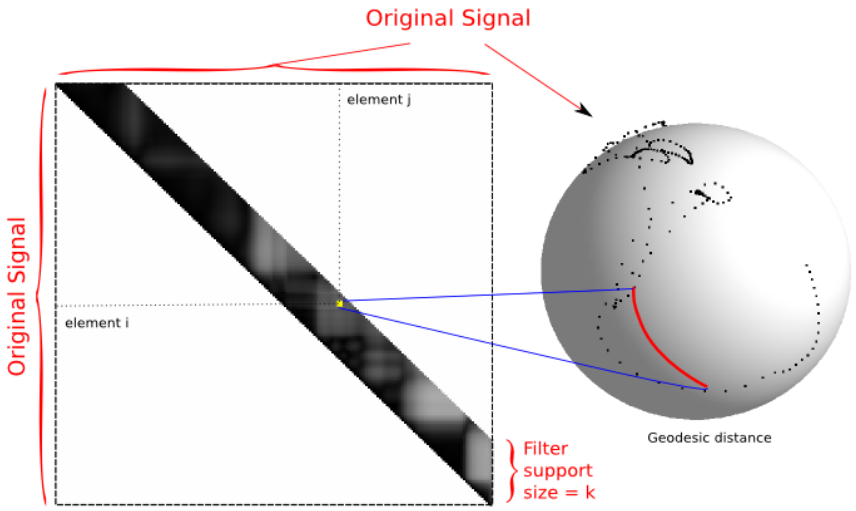


Figure 7.: **Precomputation table**. This figure shows a band matrix which contains the distance information used by the filter. Each element correspond to a geodesic distance between samples. Normally, the quaternionic signal belongs to the unit hypersphere in $4D$, but has been represented here, without loss of generality, as the unit sphere in $3D$.

Proofs can be found in *[70]* . In the case of human motion processing, we simply filter every joint orientation with our filter. As the joints are organized into a hierarchy of joints (the wrist depends on the elbow which depends on the shoulder, etc.), the coordinate invariance property is strongly desirable since the result of the filtering operation will be the same whereas the joints orientations are expressed in local or global coordinate frames.

2.1.2.5. Illustrations

We now present some results obtained with our new filter on real motion capture data. These data represent a complete human body with two hands for a total of 75 joints. The performed motion correspond to a sign language motion. It is depicted in Figure 9.a. The BO filter was first tested on the rotation of the left hand. The original signal is presented in 8.a. We added manually to this signal a quaternionic Gaussian noise with variance $\sigma = 0.18$ radians (Figure 8.d). For comparison purposes, we first filter the noisy signal with a Gaussian filter (Figure 8.b) with variance $\sigma_t = 1.0$, then with the Bilateral Orientation filter (Figure 8.c) with temporal variance $\sigma_t = 1.0$ and spatial variance $\sigma_g = 0.1$. Both filters were applied five times to the noisy signal. Boundary conditions were handled by mirroring the signal at both extremities. It is interesting to notice how the overall shape of the signal and the principal features have been recovered through the filtering process. Figure 8.e and .f shows the angular velocity of the original signal compared to the final signal. It is computed as $|| \log(\mathbf{q}_i^{-1}\mathbf{q}_{i+1})|| \ rad.s^{-1}$. While Gaussian blur exhibits less peaks in the signal and a globally less important magnitude, the BO filter preserves the overall aspect of the velocity profile, at the expense of amplifying in some cases the speed magnitude.

We then processed the entire motion (75 rotation time series corresponding to every joints). The length of the motion was about 150 frames. Our implementation yields a computation time on a standard laptop of 300 ms. Figure 9 illustrates the impact of filtering the rotational components of the motion on the resulting trajectories of the end effectors (in our case, the hands) expressed in the cartesian space. Figure 9.a gives a short outline of the test motion. The test hand trajectory has been represented in red. This trajectory cumulates in some sense all the errors on the previous articulations along the kinematic chain (*i.e.* elbow, shoulder, etc.). This propagation effect

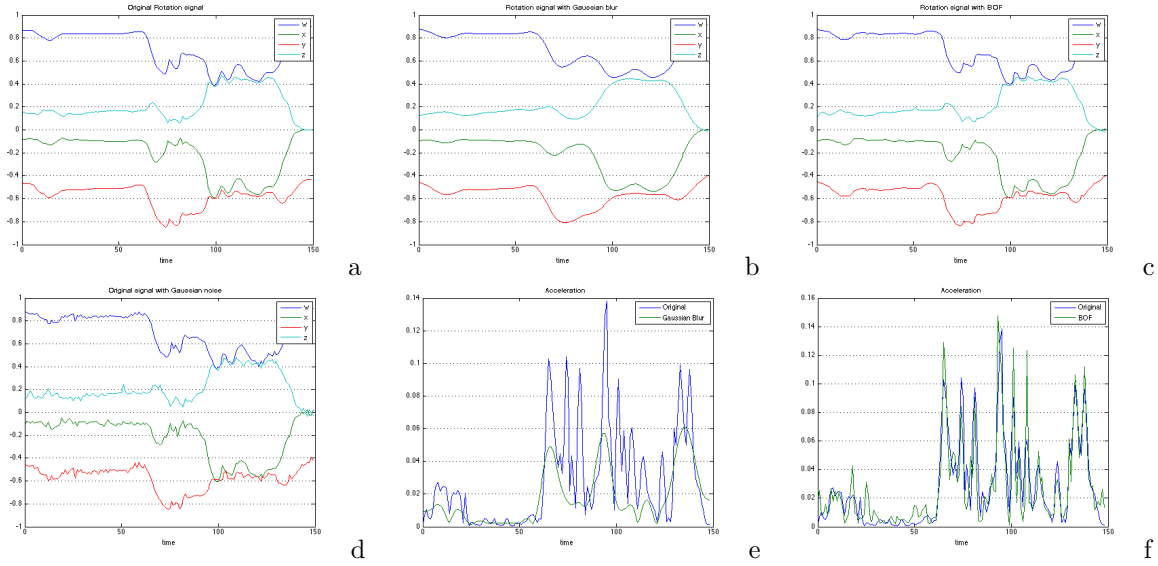


Figure 8.: **Filtering real data**. (a) Original signal (d) plus Gaussian noise (b) filtered with a Gaussian Kernel $\sigma_t = 1.0$ (c) filtered with BO filter $\sigma_t = 1.0$, $\sigma_g = 0.1$ (e,f) Comparisons between original and filtered signals angular velocities.

magnifies at the same time the effect of the filtering process. For example, processing the whole motion with Gaussian filtering leads a global diminution of the motion's energy, thus leading to a smoother trajectory but with less amplitude (Figure 9.c). In this last case, some details of the hand motion are lost (bottom left of the trajectory). Those details are in fact small and quick repetitions that are used in the case of sign language to outline a particular idea. We can see that in the case of Bilateral Orientation filter (Figure 9.d), this pattern is conserved. Moreover, the global amplitude compares better to the original signal.

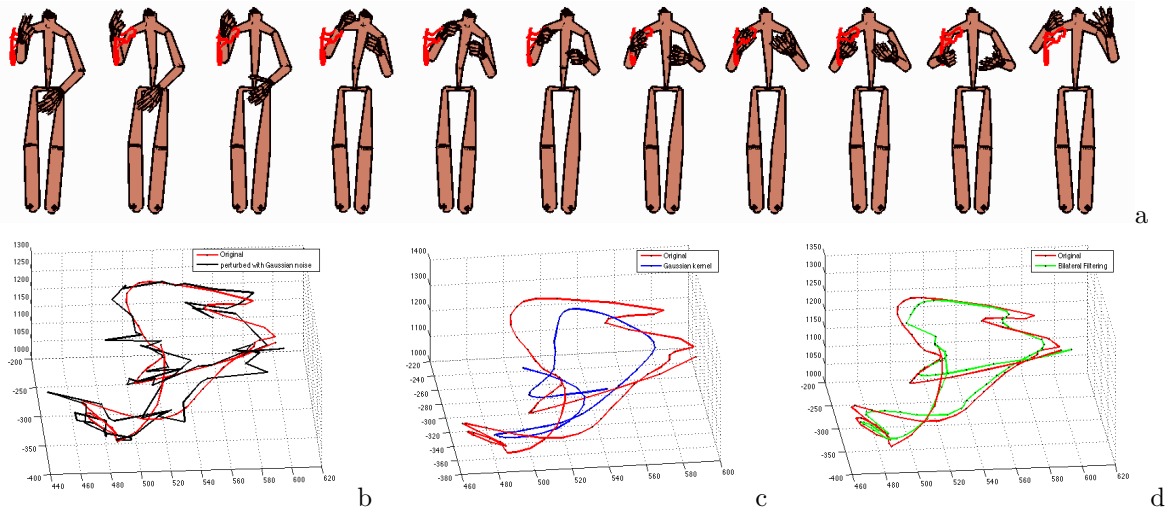


Figure 9.: **Motion strip of the test sequence**. (a) Ten poses along the test motion. The red curve shows the right hand trajectory which is used in the following illustrations. (b,c,d) Comparisons between original signal and reconstruction of the right hand trajectory (in cartesian coordinates): (b) effects of gaussian noise (c) Smoothing using a Gaussian Kernel (d) Smoothing using bilateral filtering.

2.2. Subspace decomposition

Usually, the amount of data produced by the acquisition system at a reasonable frequency ($100Hz$ is a minimum for communicative gestures) is huge and presents a lot of redundancy. A possible way to to handle the inherent problems is to perform dimensionality reduction on them. The objectives are twofold: *i)* working on smaller sets of data while keeping most of the informative part *ii)* decorrelate the different dimensions of the signal so that it is possible to work on them independently. The dimension reduction problem is often solved using descriptive statistical tools. Those tools typically yield a subspace that is more suitable for expressing the data: smaller dimension, orthogonal axis, most notably. The extension of known linear statistical tools to the non-linear case is not eased by the fact that many elementary results in the former case do not hold when dealing with more general spaces. For instance, the problem of finding the mean value of data lying on a sphere can no longer be expressed through probabilistic expected value, but has to resort to the minimization of geodesic distances. We first present an application of a classical variance based dimensionality reduction technique (Principal Component Analysis) to the problem of style translation. We then proceed to the use of its non-linear counterpart (Principal Geodesic Analysis) which allows to handle more formerly rotation data.

2.2.1. Principal Component Analysis (PCA) and its application to style variations

In a conversational situation, style conveys useful hints to verbal and nonverbal features of the discourse such as nuances, intensity, emphasis points, speaker genre, cultural background, and emotional state. Consequently, automatic generation of expressive human motion requires methods that are capable of seamlessly handling a wide range of different styles along the animation. We consider that style is the variability observed among two realizations of the same gestural sequence. This definition is voluntarily low level, signal oriented as our investigations are motivated by motion signal analysis. We worked on multiple realizations of a sequence of French Sign Language (FSL) *[89]* gestures. To do so, we asked a professional signer to perform several motion capture records of a predefined FSL sequence by varying several aspects of the discourse: mood, emphasis and speed. The temporal variability of the data was handled with Dynamic Time Warping (DTW). DTW allows to find a mapping, usually non-bijective, between two time series. This mapping realizes the best alignment between the two series, which can be seen as a non-linear temporal registration. Given this mapping, one can also produce a new motion by deforming consequently a given motion (time stretching). However, in the presence of motions, one can consider the complete time series of all the joint rotations along time. Our idea *[89]* was to only consider the time series formed by the projection of the motion in the principal subspace associated to the motion to limitate the influence of the spatial variability induced by the style.

2.2.1.1. Considering the time series of the reduced coordinates in the PCA space

We rely on the assumption that there exists a *fundamental motion* which is common to multiple styled realizations of communicative gesture sequence (CGS). An actor may perform a predefined motion sequence according to different moods, speeds, or expressivity clues, but, even when asked to be as neutral as possible, the actor will still convey his own *kinematic signature*. Still, each realization of a CGS will at least contain a common subpart that conveys the semantic of the CGS. Identification of this subpart motivates our investigations towards a low dimensional representation subspace for CGS. The construction of a style robust distance function is motivated by the assumption that the meaningful part of the gesture is embedded in the subset which presents the greatest variance. To determine this subset, PCA was used. We briefly recall the principle of PCA:

Let $\mathbf{x_i} \in \mathbb{R}^n$ be a set of elements, the PCA amounts to:

- find $\overline{\mathbf{x}}$ the mean of the data and substract it to all the values
- find a subspace $\mathbf{V}_k = \mathtt{span}(\mathbf{v}_1, \cdots, \mathbf{v}_k)$ that maximizes the variance of the data.

A recursive definition of $\mathbf{v}_k$ can be found as a maximization problem:

$$\mathbf{v}_1 = \arg\max_{|\mathbf{v}|=1} \sum_{i=1}^{n} (\mathbf{v}.\mathbf{x}_i)^2 \tag{2.9}$$

$$\mathbf{v}_k = \arg\max_{|\mathbf{v}|=1} \sum_{i=1}^{n} \left((\mathbf{v}.\mathbf{x}_i)^2 + \sum_{j=1}^{k-1} (\mathbf{v}_j.\mathbf{x}_i)^2 \right) \tag{2.10}$$

When considering the definition of a motion using time series of rotation vectors, this classical formulation is not well adapted since it does not model explicitly the specific non linearity (periodicity for instance) of the rotation data. As previously seen, a rotation in 3D space can be represented as a unitary quaternion. Seen as a fourth dimensional vector, the non linearty stems from the fact the this vector norm must be unity. Instead, one can work on a linearized version of this information, obtained through the use of the exponential map [90]. Hence, we form $\mathbf{x_i} \in \mathbb{R}^n$ as the vector $\{\exp \mathbf{q_i}(t)\} \in \mathbb{R}^{3m}$ if m is the number of rotations in the skeleton.

2.2.1.2. Application to style changing

In *[89]* we only consider the temporal aspects of the style: we assume that in order to register a new motion with a given style, it is sufficient to change the associated timing information. This mapping is usually obtained by a dynamic time warping procedure, which calls for a correct metric between two poses. As exposed in the previous paragraph, this metric was obtained thanks to the projection of the motion in a low-dimensional subspace that is common to every realizations of the CGS. Figure 10 illustrates the time alignment obtained by aligning the angry styled CGS performance onto the reference CGS performance. On one hand, the spacial variations introduced by angry style are not negligible (notice the differences between the overall postures and the amplitude of the movements). On the other hand, angry style introduces rhythmic repetitions of some specific movements. The postures depicted in Figure 10 are equally sampled and the curves represent the evolution of the influence of the first the eigenposture vectors along frames. This figure highlights the capability of our procedure to provide a smooth and accurate registration despite the spacial variations and repetitions introduced by the angry style.

2.2.2. Principal Geodesic Analysis (PGA)

We present here a version of the PCA which generalizes the concept of PCA to manifold valued data: Principal Geodesic Analysis (PGA) . PGA has first been introduced by Fletcher et al. [91]. It can be seen as a generalization of PCA on general Riemannian manifolds. Its goal is to find a set of directions, called geodesic directions or principal geodesics, that best encode the statistical variability of the data. In the Euclidean space, those geodesics are straight lines, thus leading to the classical definition of the PCA. It is possible to define PGA by making an analogy with PCA (and the definition given in the previous Section).Fletcher gives a generalization of this problem for complete geodesic spaces by extending three important concepts:

- **Variance**: expected value of the squared Riemannian distance from mean,
- **Geodesic subspaces**: geodesic submanifold,
- **Projection**: projection operator π_p onto that geodesic submanifold.

A geodesic of a Riemannian manifold is a one parameter subgroup that can be easily defined with the exponentiation map with the form: $\exp^{(t\mathbf{v})}$ where $\mathbf{v}$ is a direction in the tangent space and t a scalar. The k-dimensional geodesic submanifold $\mathbf{V}_k = \exp_{\overline{x}} \ \ span(\mathbf{v_1}, \cdots, \mathbf{v_k})$ is given by the

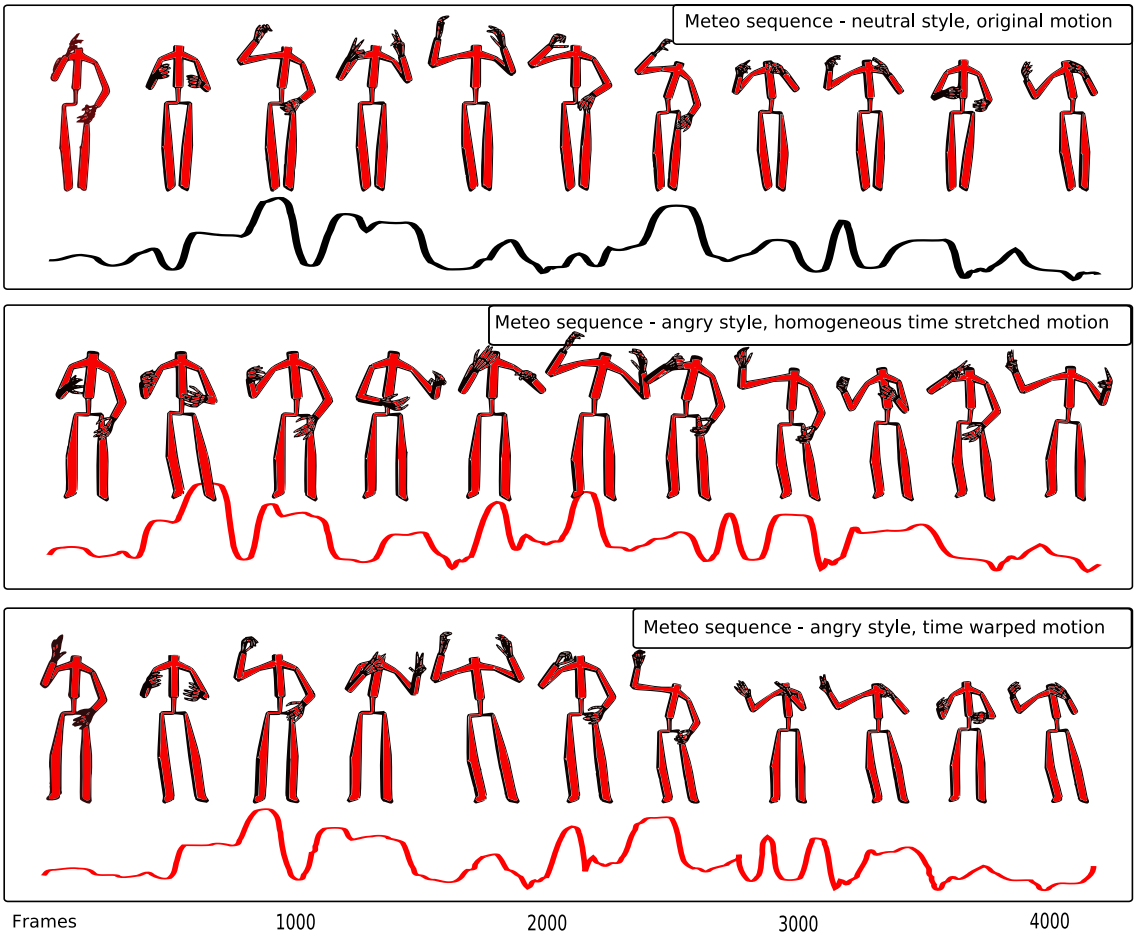


Figure 10.: Style changing by motion warping. From top to bottom, the two upper sequences are captured from original performances according to neutral and angry style, respectively. The third sequence represent the original angry sequence warped along the original neutral sequence.

new following recursive definition:

$$\mathbf{v_1} = \arg\min_{||v||=1} \sum_{i=1}^{n} ||\log(\pi_H(\overline{x}^{-1}x_i))||^2 \tag{2.11}$$

$$with \quad H = \exp_{\overline{x}} \; span(\mathbf{v}) \tag{2.12}$$

$$\mathbf{v_k} = \arg\min_{||v||=1} \sum_{i=1}^{n} ||\log(\pi_H(\overline{x}^{-1}x_i))||^2 \tag{2.13}$$

$$with \quad H = \exp_{\overline{x}} \; span(\mathbf{v_1}, \cdots, \mathbf{v_k}, \mathbf{v}) \tag{2.14}$$

In order to implement this decomposition method for a given manifold, one has first to find a closed-form solution for the projection operator. Though, in [91], Fletcher gives an algorithm that compute an approximate version of the PGA. It relies on a global linearization in the tangent space at the mean of the input data. More recently, we have proposed in *[92]* an exact computation of the PGA for rotation data. However, for a complete motion data $\in SO(3)^n$, a closed form solution of this projection operator still remains to be found. Let us note that there exists some possible solution to this problem in the form of a non-convex optimization problem [93], but we have not tested this approach. Instead, we choose to rely on a first order approximation of the exponential

map, as described by Fletcher in [91]. The quality of this approximation strongly depends on the curvature of the manifold, as discussed in [94].

We employed the PGA framework in different works *[95, 96]* to describe the variability of the inner joints orientations during a motion. The geodesics can be looked upon as the eigenposes of the skeleton during the sequence. We sometimes included the root joint's orientation in the analysis *[96]* , but we will not consider this case in the following. Hence, the pose of the skeleton is represented by a vector of the direct product $SO(3)^n$, where n is the number of joints of the skeleton. Applying the approximate PGA to the poses data from a motion with $m \in \mathbb{N}$ frames yields:

1. The intrinsic mean of the data, $\mu \in SO(3)^n$
2. $k \in \mathbb{N}$ tangent directions $(\mathbf{v_j})_{1 \leq j \leq k}$, where each $\mathbf{v_j} \in \mathfrak{so}(3)^n \approx R^{3n}$ uniquely defines a geodesic of $SO(3)^n$
3. A set of coordinates $T = (t_{i,j})$ where $1 \leq i \leq m$ and $1 \leq j \leq k$, where the i^{th} row is the approximate projection of the i^{th} pose over the k geodesics

The i^{th} pose can then be recovered partly using the k leading geodesics with:

$$p_i = \mu \prod_{j=1}^{j=k} \exp(t_{i,j}.\mathbf{v_j}) \tag{2.15}$$

This parametrizes the approximate poses manifold using the canonical coordinates of the second kind (ccsk). One can think of the reconstruction formula as a weighted composition of the k first eigenposes. Note that here the exponential over the direct product $\mathfrak{so}(3)^n$ is used. By only considering the $k \leq n$ first modes of the PGA, we obtain a new reduced parametrization of a motion in terms of geodesics coordinates. We show next how such a reduced pose parametrization can be used to perform inverse kinematics.

2.2.2.1. Inverse Kinematics in the PGA reduced space

The reduced parametrization with geodesics coordinates allows us to define a function $f : \mathbb{R}^k \to \mathbb{R}^{3d}$ that maps a set of geodesics coordinates $\mathbf{x} \in \mathbb{R}^k$ to the global space of positions of $d \in \mathbb{N}$ end-effectors: $\mathbf{y} \in \mathbb{R}^{3d}$. This function is the composition of the reduced pose parametrization by the ccsk $h : \mathbb{R}^k \to SO(3)^n$ and the classical direct kinematics function, which maps a skeleton pose to the global position of the d end-effectors, $g : SO(3)^n \to \mathbb{R}^{3d}$. The derivative of g at the pose $\mathbf{x} \in SO(3)^n$ simply maps instant rotation vectors for each joint to the linear velocities of the end-effectors.

Since $h(x) = \mu \prod_{j=1}^{j=k} \exp(\mathbf{x}_j.\mathbf{v_j})$ is a product of differentiable functions (the exponentials) in a Lie group, h is therefore differentiable. Each partial derivative of h with respect to x_j can be easily computed due to the ccsk parametrization. We are eventually able to compute the whole Jacobian matrix $\mathbf{J_f}$ of the function f using chain rule. We then use this Jacobian in a least square optimization method, such as the well-known Levenberg-Marquardt algorithm, in order to find the geodesic coordinates $\mathbf{x_j}$ that best match the given end-effectors constraints $\mathbf{y}_0 \in \mathbb{R}^{3d}$:

$$\mathbf{x}^\star = \underset{\mathbf{x} \in \mathbb{R}^k}{\arg\min} \left(||f(\mathbf{x}) - \mathbf{y}_0||^2 \right) \tag{2.16}$$

The benefits of using this method are threefold:

- The optimization is done in a much smaller space than traditional IK (usually 30 degrees of freedom): this not only speeds up the process, but also better constraints the IK problem.
- The geodesics being principal poses modes, the optimization naturally exploits correlations between joints to reach the objectives, resulting in a more natural pose.
- The geodesics formulation allows a quick computation of the Jacobian $\mathbf{J_f}$ used in the optimization, thus eliminating the need for numerical differentiation.

The main drawback is that the geodesics yield a limited reachable space: our IK works better in a neighborhood of the input data. This pitfall is common to all Inverse Kinematics approach in reduced space [28, 21]

2.2.2.2. Application to motion compression

We present shortly how the PGA can be used in a particularly performant motion compression technique *[95]* which exploits both temporal and spatial coherence to achieve high compression ratios with few perceptual distortion. After the principal geodesics have been extracted from the input motion using approximate PGA, global end-joints trajectories can be compressed using any linear compression method. The root orientation is eventually compressed using the multiscale representation. The decompression phase consists in decompressing the global trajectories as well as the global root orientation, then expressing the end-joints positions in the root joint's frame, and eventually performing PGA-based IK to recover poses. Our experiments show that the use of a compact pose model allows to successfully recover poses given only end-joints positions. As the end-joints and root joint's trajectories present high temporal coherence, they can also be compressed efficently in order to further improve compression rates. A particularly appealing aspect of our technique is that the pose model may also be used for editing compressed motions by employing the very same algorithm.

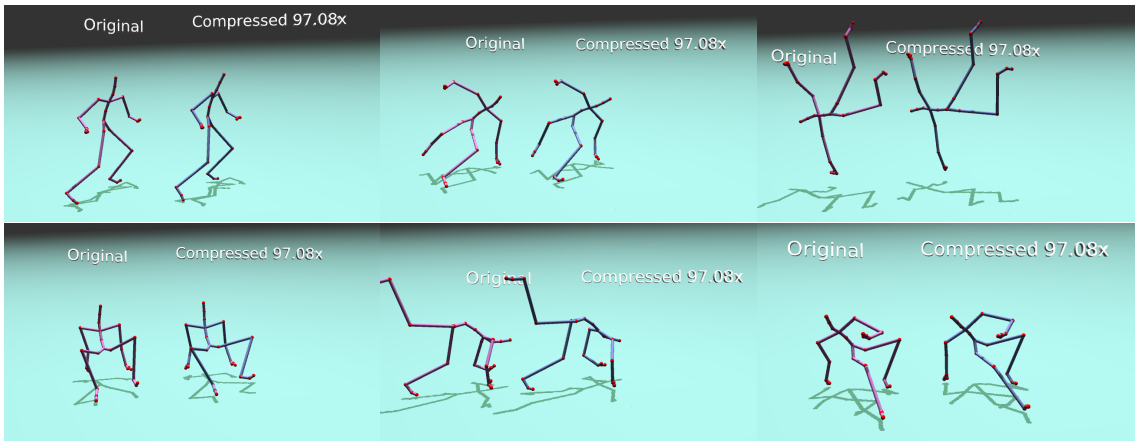


Figure 11.: Hip-Hop motion from the CMU database compressed with our technique

We finally note that we also used more recently the concept of PGA in a pure machine learning setup *[97]* . As the subject of this work is not directly aimed toward computer animation, we left its description in annex B.

2.3. Analysing the motion dynamics

A dimensional reduction method is applied to the motion capture data (PGA). The motion signal $\mathbf{x}_t$ is the corresponding trajectory in the PGA latent space and is defined over the entire motion duration, *i.e.* $t = 1 \cdots T$. This trajectory can now be used to learn the dynamical behavior of a captured motion. We present to statistical methods that reach this goal: identification of a linear time-invariant model and Gaussian processes.

2.3.1. Linear time-invariant models

The first method considers to fit a m-order linear time-invariant model [98, 19] to the trajectory:

$$\mathbf{x}_t = \sum_{i=1}^{m} \mathbf{A}_i \ \mathbf{x}_{t-i} + \mathbf{B} \ \mathbf{u}_t + \epsilon_t, \tag{2.17}$$

where the set of $\mathbf{u}_t$ can be assimilated as input control variables and ϵ_t is a Gaussian noise. Given the low-dimensional representation of the original motion capture data $\mathbf{x}_{0:N}^{ref}$, the matrices $\mathbf{A}_1 \dots \mathbf{A}_m$, B and the set of input control variables $\mathbf{u}_t$ can be calculated with simple procedures (least-square solver and SVD decomposition of the residual) [98]. In practice, A_i matrices are estimated by computed the least-square solution.

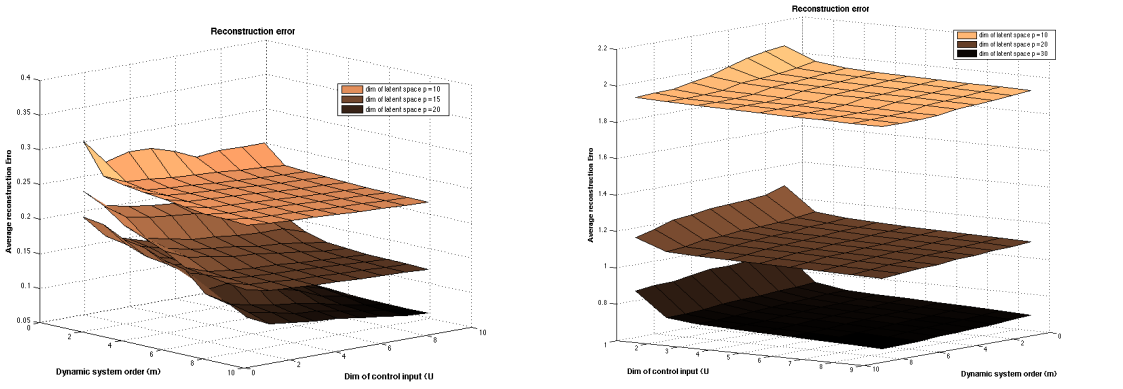
$$\hat{\mathbf{A}}_1 \dots \hat{\mathbf{A}}_m = \arg \min_{\mathbf{A}_1 \dots \mathbf{A}_m} \sum_{t=m}^{T} \| \mathbf{x}_t^{ref} - \sum_{i=1}^{m} \mathbf{A}_i \ \mathbf{x}_{t-i}^{ref} \|^2, \tag{2.18}$$

The order m of the system can be decided with standard statistics criteria such as BIC. Denoting

$$\mathbf{z}_t = \mathbf{x}_t^{ref} - \sum_{i=1}^{m} \hat{\mathbf{A}}_i \ \mathbf{x}_{t-i}^{ref}, \tag{2.19}$$

and $\mathbf{Z} = [\ \mathbf{z}_{m+1} \mid \dots \mid \mathbf{z}_T\ ]$, performing an SVD decomposition of Z leads to B, matrices of eigenvectors and the control input $\mathbf{u}_{m+1:T}$ are subsequently derived. Here again the dimensionality of the control space can be chosen such that there's a drop in the eigenvalues of $\mathbf{Z}$.

We show examples of the reconstruction error in Figure 12. As expected, the higher the dimension of latent space, the more impact has rising the number of dimensions of the control space.



a b

Figure 12.: Reconstruction error for different dimensions of the latent space and the control space and also the order of the dynamic model for (a) a walking motion (b) a signed language motion

This type of autoregressive models is useful to build a computational model of the motion dynamics, and can also be used as a statistical prior in methods such as the Monte Carlo approach to motion production presented in the next Chapter. We present now another possible but yet closely linked representation based on Gaussian processes.

2.3.2. Gaussian processes

The PGA trajectory is now assumed to be a realization of a Gaussian process X_i with covariance function C_i. This process is assumed to be ergodic, meaning that its statistical properties can be

inferred from one finite realization of it. If the observed realization is of sufficient length, we can indeed consider that it contains the same information as several different realizations of the process. We also assume that the underlying process is stationary, meaning its joint probability distribution does not change when shifted in time, reducing for the Gaussian process to the property that the two first moments do not depend on time.

In practice, a parametric model is first chosen for the covariance function and its hyperparameters are estimated from each realization $\mathbf{X}_i$. An example of parametric covariance model is the following one:

$$C_i(t,t') = \alpha_i \exp\left(-\frac{|t-t'|^2}{\rho_i}\right) + \sigma_i \delta_{tt'}, \tag{2.20}$$

where ρ_i will be called the length-scale which determines how quickly the covariance falls, δ is one if $t = t'$ and zero elsewhere, and the associated σ_i traduces the nugget effect (small scale variations, corresponding to noise). This model is used for all applications in this document, and the parameters are estimated with a maximum likelihood approach for each PGA component.

Once the parameters of the covariance functions C_i are known for all PGA directions, a model of motion is available in the PGA space. New motions can then be synthesized from this model. If one aims at simulating motions with the same statistical properties as the reference motion, a realization of a Gaussian process with covariance C_i can easily be obtained for each component, and a new motion can be reconstructed from the PGA approach. However, in order to improve the resulting motion, constraints have to be introduced into the simulation procedure. The problem can then be formulated as the conditional simulation of a Gaussian process, with kinematic constraints as constraint values. Those aspects, related to motion control, will be detailed in the next Chapter.

2.4. Summary

In this Section the classical representation of a motion with rotations and its analysis were exposed. Motions constitute high-dimensional multivariate time series, which analysis is mandatory to reveal the hidden structure of the correlations both observable in space and time. Yet, the non-linear nature of the rotation data imposes several restrictions to the classical signal processing techniques and call for the use of dedicated methods. We have proposed filtering algorithms or dimensionality reduction techniques adapted to this type of data. Also, the temporal analysis was discussed with two statistical methods that can encode some prior information on the motion. The next Chapter is dedicated to the use of this information in the production of new motions.

3

Motion control: how to use data

Contents

In this chapter we give an overview of our contributions in the domain of motion control. By motion control, we are referring to the methods capable of producing motions from some motion references. In this sense, they can be considered as supervised by the data. The extent to which the data is incorporated in the command loop greatly varies between the methods. We first present an alternative solution to the classical problem of inverse kinematics 3.1. We then discuss an original method which produces new motions based on statistical dynamic prior on the motion dynamics based on stochastic simulation 3.2. We end this chapter by describing a method which combines in space and time motions to produce new sign languages utterances 3.3.

3.1. Kinematic methods

We begin this Chapter by describing a contribution *[96]* to the problem of inverse kinematics. Given a kinematic chain described by a fixed number of segments linked by joint angles, the forward and inverse kinematics problems can be derived. The first one amounts to computing the pose of the figure given the values of the joint angles. The second one is the process of determining the parameters of the kinematic chain in order to achieve a desired configuration. The latter has been extensively studied in computer animation due to its huge number of applications, such as connecting characters to the virtual world (feet landing on top of terrain, or hands lining up with doorknobs), as well as in robotics, where manipulator arms are commanded in terms of joint velocities. While the forward problem has a unique solution, the inverse problem does not in the

general case. Because of this, in the inverse problem, one needs to make explicit any available *a priori* information on the model parameters. Traditional ways of solving inverse kinematics rely on analytical or numerical methods, where some constraints are added to choose one of the solutions. Carrying out such methods becomes rapidly difficult for a large number of model parameters.

A very general theory to solve inverse problems is obtained when using a probabilistic point of view, where the *a priori* information on the model parameters is represented by a probability distribution over the model space (i.e. the state space). This *a priori* probability distribution is transformed into the posterior probability distribution, by incorporating a theory (relating the model parameters to some observable parameters) and the actual result of the observations (with their uncertainties). The probabilistic formulation of the inverse problem requires a resolution in terms of **samples** of the posterior probability distribution in the model space. This, in particular, means that the solution of an inverse problem is not a model but a collection of models (that are consistent with both the data and the *a priori* information). The generation of this collection of possible figures can be accomplished by means of an efficient Monte Carlo method.

Following this direction of work, we propose to solve the inverse kinematics problem using a Monte Carlo approach. We present how **sequential** Monte Carlo methods (SMCM) can be advantageously used for inverse kinematics. The inverse kinematics is thus re-formulated in a filtering framework. This allows us to derive a simple and efficient algorithm that can be seen as a filter whose state is the entire complex articulated figure. The sequential aspect of the procedure is one of its keypoints. The algorithm produces a complete motion, from the initial position to the target position as a result of the optimization procedure where each intermediate pose corresponds to an optimization step. The motion velocity is directly dependent on the filter parameters, making the algorithm flexible. The produced motions may then be used by an animator or an interactive animation system.

The contributions of our method to the domain of articulated character control are threefold:

- our method does not require any explicit numerical inversion to solve the control problem,
- any type of constraints can be added to the system in a simple and intuitive manner provided that an *evaluation function* can be provided (no derivation *wrt.* articular space is required),
- this method can be implemented in a few lines of codes and tested easily without the needs of complex optimization algorithms.

Let us remark that another strong motivation for this work comes from motor neuroscience where Körding and Wolpert [99] have highlighted the bayesian nature of sensori-motor learning and the role of uncertainty in the realization of a motor task.

3.1.1. Theoretical background on Inverse Kinematics

In this section we consider a kinematic chain $\mathcal{C}$ composed of n joints and defined by the length of its different segments $\{l_1, \ldots, l_n\}$. $\mathcal{C}$ is parameterized by the following rotation vector $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \in SO(3)^n$ (which defines the *articular space*).

It is possible to define the forward kinematic operator $\mathbf{H}$ that computes the configuration of the end effector of the chain. Usually this configuration $\mathbf{P}$ is defined by a position and an orientation, *i.e.* $\mathbf{P} \in SE(3)$ (*task space*):

$$\mathbf{H} : \mathbb{R}^3 \times SO(3)^n \longmapsto SE(3) \tag{3.1}$$

$$\{\mathbf{r}_1, \mathbf{q}_1, \ldots, \mathbf{q}_n\} \longmapsto \prod_1^n \mathbf{M}_i(l_i, \mathbf{q_i}) = \mathbf{P} \tag{3.2}$$

where $\mathbf{r}_1$ is the root position of the chain and $\mathbf{M}_i(l_i, \mathbf{q_i})$ the homogeneous transformation matrix representing the rotation of a segment of length l_i by the quaternion $\mathbf{q}_i$. For clarity purposes this operation will be summarized by:

$$\mathbf{P} = \mathbf{H}(\mathbf{Q}) \tag{3.3}$$

Problem Statement The goal of inverse kinematics technics is to find a vector $\mathbf{q}_i$ such that $\mathbf{P}$ is equal to a given desired configuration $\mathbf{P}_d$. This problem amounts to the following non-linear inverse problem which does not always have a unique solution and is not always well-behaved:

$$\mathbf{Q} = \mathbf{H}^{-1}(\mathbf{P}_d) \tag{3.4}$$

Numerical resolution Most of the previous works on inverse kinematics solve equation (3.4) by using a local linearization method which amounts to converging to the solution by computing small variations $\dot{\mathbf{Q}}$ in the articular space that ensure the regulation from $\mathbf{P}$ to $\mathbf{P}_d$:

$$\dot{\mathbf{Q}} = -\lambda \mathbf{J}^{+}_{\mathbf{Q}}(\mathbf{P} - \mathbf{P}_d) \tag{3.5}$$

where $\mathbf{J}^{+}_{\mathbf{Q}}$ is the pseudo inverse of the Jacobian of $\mathcal{C}$ evaluated around the configuration $\mathbf{Q}$, and λ a scalar which sets the rate of convergence. Evaluation of the Jacobian (matrix of partial derivatives $\{\frac{\partial P_j}{\partial \mathbf{q}_i}\}$) is usually done with finite difference methods. The computation of the pseudo-inverse is one of the critical parts of the inverse kinematics. Some works [6, 7, 100] have explored the possibility to use the transpose of the Jacobian $\mathbf{J}^t$. As this solution relies on the assumption of the convexity of $\mathbf{H}$ (which is far from being the case), it has proved to be less efficient in the general case and exhibits smaller convergence rates. One of the most common techniques relies on the Singular Value Decomposition (SVD) which has the advantage of being robust to ill-posed problem since singularities can be detected and treated during the process. It is also current to consider a slightly modified version of the pseudo-inverse that guaranties that singularities are avoided; it is referred to as the damped or singularity robust (SR) pseudo-inverse [101, 102, 10].

Adding constraints Since the number of degrees of freedom in our kinematic chain is (most of the time) greater than the size of the task space, the number of solutions is usually infinite. With the pseudo-inverse approach the minimal norm solution is chosen by the algorithm, but it can be useful to control with additive constraints the choice of this solution. This operation is possible thanks to the use of projection operator $(\mathbf{I}_n - \mathbf{J}^{+}\mathbf{J})$ that allows us to project a constraint (or secondary task) on the null-space of $\mathbf{J}$. The new solution is given by:

$$\dot{\mathbf{Q}} = \mathbf{J}^{+}_{\mathbf{Q}}\dot{\mathbf{P}} + (\mathbf{I}_n - \mathbf{J}^{+}\mathbf{J})\frac{\partial h}{\partial \mathbf{q}_i} \tag{3.6}$$

where $\mathbf{I}_n$ is a $n \times n$ identity matrix and h is usually expressed as a generic cost function that needs to be analytically derived *wrt.* articular parameters. This formulation ensures that the secondary task will have no effects on the regulation from $\mathbf{P}$ to $\mathbf{P}_d$. This secondary task has been extensively used, notably for enforcing joint limits [103] or control the position of the center of mass [9] for instance. Recent works deal with adding several levels of constraints to the system [11, 104, 12]. In this particular theoretical framework (also called Prioritized inverse kinematics), the difficulty is to balance and order the different constraints (which is usually performed manually) without *a priori* knowledge on how the global task will be realized.

3.1.2. Inference problem and SMCM

3.1.2.1. Formulation overview

We propose a statistical inverse kinematics solver. It is based on a Bayesian formulation of the problem, that enables us to combine motion prior, skeleton constraints (i.e. joint limits) and kinematic constraints. We denote $\mathbf{x} = \mathbf{x}_{0:M} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_M\}$ the sequence of poses from the initial pose of the chain $\mathbf{x}_0$ to its final pose $\mathbf{x}_M$ satisfying the kinematic constraints. The goal is to infer the most likely trajectory $\hat{\mathbf{x}}$ given the set of kinematic constraints $\mathbf{z}$. We have:

$$\hat{\mathbf{x}} \quad = \quad \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{z}) = \arg\max_{\mathbf{x}} \frac{p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x})}{p(\mathbf{z})} \tag{3.7}$$

where $p(\mathbf{z})$ is a normalizing constant. The involved components are the motion prior $p(\mathbf{x})$ and the constraint likelihood $p(\mathbf{z}|\mathbf{x})$. The motion prior carries the *a priori* knowledges about the intrinsic nature of the motion, as well as biomechanical constraints ; whereas the likelihood calculation gives an evaluation on how good is the pose with respect to the kinematic constraints that have to be satisfied.

Two main families of methods are used to solve this inference problem. The first one relies on a direct estimation of the *maximum a posteriori* $\hat{\mathbf{x}}$ through an optimization procedure of $-\log(p(\mathbf{z}|\mathbf{x})\ p(\mathbf{x}))$. An example of this approach has been recently used for constraint-based motion optimization [19]. Monte Carlo methods are the second family. They approximate the probability density itself $p(\mathbf{x}|\mathbf{z})$ and *then* estimate $\hat{\mathbf{x}}$ through the use of *maximum a posteriori* or *minimum mean square error* estimates. Traditionally solvers from these two families are non-sequential, that make them improper for an efficient on-line animator use.

For this purpose, we propose to use a *sequential* Monte Carlo technique. The resulting algorithm has the advantage of being at the same time easy to implement, robust and adapted to online practice. The formulation (3.7) has to be modified to suit a sequential approximation of $p(\mathbf{x}|\mathbf{z})$. Let suppose that constraints $\mathbf{z}$ can be decomposed into a set of constraints $\mathbf{z}_{0:M}$; each $\mathbf{z}_k$ has to be satisfied at point of time k. Typically, this can be translated into a progressive hardening of the kinematic constraints. Then $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ is expressed using $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$ $(k \leq M)$:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{0:k-1})\ d\mathbf{x}_k}, \tag{3.8}$$

where

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\ p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})\ d\mathbf{x}_{k-1} \tag{3.9}$$

To derive this expression, one has to suppose the *hidden state process* $\mathbf{x}_{0:M}$ to be Markovian. The new involved components are : the motion prior, now described as an *evolution prior* $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and an *instantaneous constraint likelihood* $p(\mathbf{z}_k|\mathbf{x}_k)$. Those two densities define the *model* of the system.

Doing an analogy between constraints and observations, we can recognize here a filtering problem. The filtering recursion (3.8 - 3.9) yields closed-form expressions only for specific cases. The most well-known case is the Kalman filter for linear Gaussian models. Non optimal extensions of the Kalman filter, based on a Gaussian approximation of the filtering distribution (Extended Kalman filter, Unscented Kalman filter [105], [106]), have been devised for non linear systems. In the general multi-modal case, such an approximation is not satisfactory. For general non-linear non-Gaussian models, the recent development of sequential Monte Carlo approaches [107, 108] has lead to new efficient algorithms. Before commenting upon the specific model we propose for inverse kinematics, we describe these methods in the next subsection.

3.1.2.2. Sequential Monte Carlo methods

The idea behind sequential Monte Carlo algorithms is very simple. These techniques propose to implement recursively an approximation of the sought density $p(\mathbf{x}_k|\mathbf{z}_{0:k})$ (called the filtering distribution). This approximation consists in a finite weighted sum of N Diracs centered on hypothesized locations in the state space – called particles – of the initial system $\mathbf{x}_0$. At each particle $\mathbf{x}_k^{(i)}$ $(i = 1 : N)$ is assigned a weight $w_k^{(i)}$ describing its relevance. This approximation can be formulated with the following expression:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) \approx \sum_{i=1:N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}_k). \tag{3.10}$$

Assuming that the approximation of $p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1})$ is known, the recursive implementation of the filtering distribution is done by propagating the swarm of weighted particles $\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1:N}$. At each time instant (or iteration), the algorithm can be decomposed into three steps :

1. *exploration of the state space:* The set of new particles $\{\mathbf{x}_k^{(i)}\}_{i=1:N}$ is drawn from an approximation of the true distribution $p(\mathbf{x}_k|\mathbf{z}_{0:k})$, called the *importance function* and denoted $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})$. The closer the approximation to the true distribution, the more efficient the filter.
2. *evaluation of particles relevance using the observations (i.e. calculation of the new importance weights):* The importance weights $w_k^{(i)}$ account for the deviation w.r.t. the unknown true distribution. To maintain a consistent sample, the importance weights are updated according to a recursive evaluation as the new measurement $\mathbf{z}_k$ becomes available:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})\ p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k})}, \qquad \sum_{i=1:N} w_k^{(i)} = 1. \tag{3.11}$$

3. *mutation/selection of the particles:* From time to time, it is necessary to perform a resampling step. This procedure aims at removing particles with weak normalized weights, and multiplying particles associated to strong weights, as soon as the number of significant particles is too small. Consequently, resampled particles tend to be concentrated in areas where important features exist.

These three steps (sampling / calculation of the importance weights / resampling) constitute the general framework of sequential Monte Carlo filter. Then, different instances of this general algorithm can be defined according to the choice of the importance function and/or the choice of the resampling strategy (see [108]). In particular, the simple method we use is built with the following rules: (a) to set the importance function to the evolution law, i.e. $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{0:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$; (b) this implies the calculation of the weights using $w_k^{(i)} \propto w_{k-1}^{(i)}\ p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$. The application of this algorithm for inverse kinematics problem is described in the next section.

3.1.3. SMCM for Inverse Kinematics

In this section we present the inverse kinematics filter. After defining our notations in a first part, the second part describes the design of motion prior and likelihood when modeling the inverse kinematics problem. The last part is dedicated to the corresponding algorithm.

3.1.3.1. Notations

Let consider a kinematic chain parameterized by a vector of rotations. Each rotation, expressed as a unitaryquaternion, corresponds to one joint and may have one, two or three degrees of freedom. Quaternions lives on the hypersphere S^3. We denote $\phi(\mathbf{q}\,;\mathbf{m},\Sigma)$ the Gaussian quaternionic density of variable $\mathbf{q}$. This density is called QuTem distribution in [90]. It corresponds to the Gaussian distribution of covariance Σ in the tangent space at the quaternion mode $\mathbf{m}$ wrapped onto a hemisphere of S^3 [90].

To each joint is associated a quaternion and its QuTem distribution. The covariance matrix of this QuTem distribution designs the kinematic properties of the joint (number of degrees of freedom). This is depicted figure 13 where realizations on S^3 for three different covariance values of this distribution are shown. For instance, modeling a one degree of freedom (DOF) joint amounts to consider only one possible axis of rotation (and its opposite). This property is modeled by a diagonal covariance matrix with only one non-zero eigenvalue (Figure 13.c). Generalizing this idea, a 2 DOF joint will exhibit a diagonal covariance matrix two non-zero eigenvalues and full ball-and-socket joint will have three non-zero eigenvalues. This system allows to model for instance one DOF joint with a small variations allowed on the remaining DOF (just as in the human body where the biomechanical nature of the joints allow this), provided that the two other eigenvalues are much more smaller (the example of Figure 13.c is a good illustration of this). Appendix A details how to sample from the QuTem distribution.

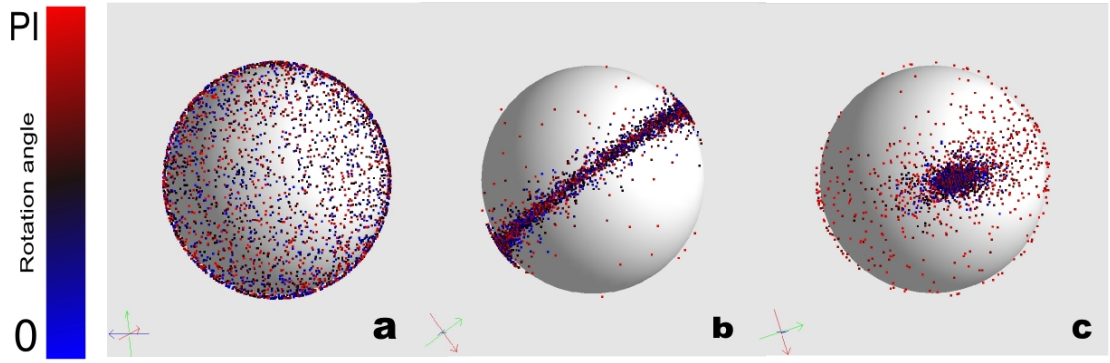


Figure 13.: **Quaternion distribution** In this figure the equivalent representation axis–angle of a quaternion is adopted. Points on S^2 represent a rotation axis while the varying color stands for the rotation angle along the axis; 1000 samples over a QuTem distribution on S^3 with (a) $\sigma_1 = \sigma_2 = \sigma_3 = 1$ (3 DOF joint) (b) $\sigma_1 = \sigma_2 = 1, \sigma_3 = 0.05$ (2 DOF joint) (c) $\sigma_1 = 1, \sigma_2 = 0.1, \sigma_3 = 0.05$ (1 DOF joint)

Supposing that each quaternion of the kinematic chain follows a Qutem distribution, the distribution of the quaternion vector $\mathbf{Q}$ is denoted $\Phi(\mathbf{Q}\,;\mathbf{M},\Sigma)$. We assume that this last distribution defines a Gaussian distribution over the pose space $SO(3)^n$. $\mathbf{M}$ and Σ are deduced from the QuTem parameters.

3.1.3.2. Model design

The goal of inverse kinematics is to estimate the value of the vector $\mathbf{Q}$ such that the resulting kinematic chain satisfies the kinematic constraints and the joint limits. One may also want to fix other constraints such as balance constraints for instance. As said before, we propose to reformulate this problem in a filtering framework. The rotation vector is now seen as a random variable evolving in time until the final task is reached. The notation $\mathbf{Q}_k$ describes the random vector of quaternions at iteration k.

We choose to simply set the state vector $\mathbf{x}_k$ of the filter as the rotation vector $\mathbf{Q}_k$. The motion trajectory $\mathbf{x}_{0:M} = \mathbf{Q}_{0:M}$ will be the result of our algorithm, under the assumption that the optimization iteration time k also corresponds to the motion decomposition time. The sets of various constraints are taken into account in the design of the evolution prior and likelihood.

Evolution prior The evolution prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ carries the *a priori* knowledge about the intrinsic nature of the motion, as well as biomechanical constraints. As in the kinematic framework no *a priori* motion has to be verified, our model is simply a random walk model under the condition that the new sampled pose $\mathbf{x}_k$ enforces the joint limits of the skeleton. We propose the following general design:

$$\boxed{p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \text{random walk} \setminus \mathbf{x}_k \text{ enforces joint limits}}$$

This is equivalent to assume that the configuration remains constant along time. Angular displacements are only supported by a Gaussian noise model. This leads to:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad = \quad \Phi(\mathbf{x}_k\,;\mathbf{x}_{k-1}\,,\,\Sigma_{\mathbf{x}}) \setminus \mathbf{x}_k \text{ enforces joint limits}$$

The covariance matrix $\Sigma_{\mathbf{x}}$ contains the kinematic properties of each joint as explained in section 3.1.3.1. The following method is applied to sample from (3.12). Once a sample is drawn from Φ, an accept/reject procedure is applied to satisfied the condition on the joint limits: while

sampling over the current configuration, if a joint does not enforce its corresponding limit, a new orientation value is sampled. This rejection/acceptance process guarantees that no impossible configurations will be considered. A drawback of this method is that it is not totally efficient (and may lead to the worst, but highly improbable, case of endless rejection).

Also, it is possible to use here the dynamical model learned from on motion. One can use the LTI model presented in equation 2.17. In this sense, the optimization is also data-driven, and the produced trajectory matches at best the learnt motion dynamic.

Likelihood The likelihood calculation $p(\mathbf{z}_k|\mathbf{x}_k)$ gives an evaluation on how good is the configuration with respect to the kinematic constraints that have to be satisfied. It is designed as:

$$\boxed{p(\mathbf{z}_k|\mathbf{x}_k) = \exp(-||\text{distance to task}||_{\Sigma_\mathbf{z}}) \prod \text{other constraints}}$$

where $|d|_\Sigma$ is the Mahalanobis distance $d^t\Sigma^{-1}d$, and $\Sigma_\mathbf{z}$ is the covariance of the noise.

As an example, If a unique kinematic constraint is imposed, the likelihood of a given state $\mathbf{x}_k$ is evaluated by calculating the distance between the end effector configuration – computed using the forward kinematic operator $\mathbf{H}$ described in equation (3.1) – and the desired configuration $\mathbf{P}_d$. The likelihood model is therefore:

$$p(\mathbf{z}_k|\mathbf{x}_k) \propto \exp \quad (-||d(\mathbf{P}_d \ , \ \mathbf{H}(\mathbf{x}_k))||_{\Sigma_\mathbf{z}}) \tag{3.12}$$

where $d(.,.)$ is the distance function in the task space.

Other constraints may be added into the model. The methodology to do so is the following: each constraint has to be expressed in terms of a cost function whose value is 0 if the constraint is satisfied and large otherwise. Supposing that j different constraints (assumed to be independent) are modeled by the cost functions $C_1 \dots C_j$, associated to noise covariances $\Sigma_1 \dots \Sigma_j$ then the likelihood is defined as:

$$p(\mathbf{z}_k|\mathbf{x}_k) \propto \exp(-|\text{distance to task}|_{\Sigma_\mathbf{z}}). \prod_i \exp(-|C_i|_{\Sigma_\mathbf{i}}) \tag{3.13}$$

Examples of constraints and their corresponding cost function are given in the results Section. Let us finally note here that setting the amplitude of the noises with respect to each constraint can be seen as a discrimination between *important* and *optional* constraints, which is related in a sense to the prioritization of constraints in traditional inverse kinematics. The corresponding algorithm can be found in *[96]* .

3.1.4. Some results

We show some results obtained with our method. Additional results can be found in *[109]* . We consider a complete human figure with 40 joints that were designed to respect predefined kinematic properties (number of degrees of freedom and joint limits). Snapshots of a resulting animation are shown in Figure 14. For this example, we added to the state space the root position (the pelvis in our case) so that the whole figure can move in the $3D$ space. For the cartesian coordinates of this link, an additive Gaussian noise was applied (conversely to the multiplicative quaternionic noise presented in the previous section). The feet were constrained to stay on the floor, while the left and the right arms were given two different targets. It is interesting to notice in the produced animation how the motions of root of the body contribute to the solution.

In this other example the considered chain is a forearm with a hand. The elbow, as well as the wrist, have been given 3 DOFs. The fingers are constituted of 3 segments. The basis of each fingers has 2 DOFs allowing abduction/adduction and flexion/extension. The remaining joints are set to have only one degree of freedom. In this animation, each fingertip is given a target (empirically

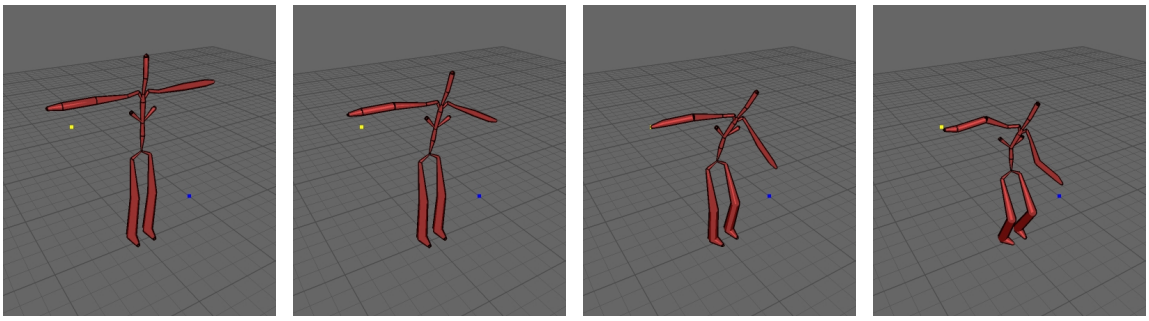


Figure 14.: **human figure animation** In this animation, feet are constrained to lie on the floor, the right hand is linked with the yellow dot while the left arm has the blue dot as target. Notice how the knees bend for the task to be achieved

determined). Two sets of targets are chained together during the animation. Figure 15 shows images from this animation. In order to increase the realism of the produced animation, we added the biomechanical constraint linking the last two joints of each fingers except the thumb:

$$\theta_{last} = \frac{2}{3}\theta_{previous}$$

where θ stands for the flexion/extension angle. At this point, let us note the difficulty of handling such a kinematic configuration and the previous constraint in a classical numerical inverse kinematics scheme where this problem would be decomposed into several problems (corresponding to several distinct linear chains) with likely conflicting solutions. Conversely with our framework this problem is treated as a global optimization problem.

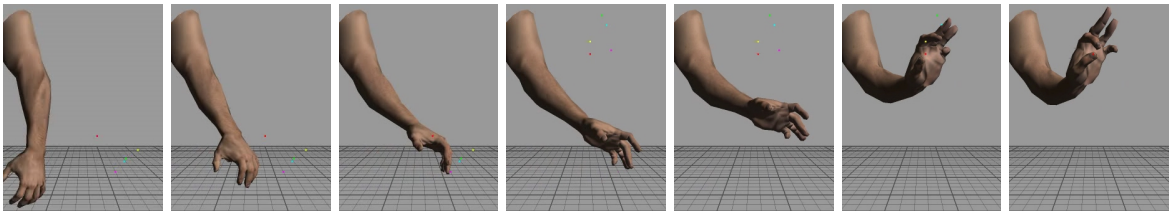


Figure 15.: **Hand animation** In this animation the fingers were given a target position represented as colored dots in the images. The two strips correspond to two different tasks that were chained along the animation.

3.2. Incorporating a dynamic prior to the motion production

In this Section, a methodology, developed in *[96]* , that allows to generate new motion from a dynamic prior expressed in the form of a Gaussian process (described Section 2.3.2) is proposed.

Our method can synthesize new motions that share the same statistics up to order two of a reference motion. Assuming that the inherent variability of a motion is a realization of a stochastic process, our method first learns its structure by treating it as a Gaussian process. Then, new realizations of motions can be obtained by stochastic simulation, which guarantees that the obtained motion has the correct statistics. Nevertheless, this is not sufficient to assert the correctness and realism of the motion. The aim of the proposed method is to allow to add kinematic constraints to the system. The contributions of the method are in this direction and are twofold: *i)* using a double kriging operation, we show how it is possible to constrain the stochastic simulation to reach

given values at given instants, which amounts to keyframe the simulation *ii)* a novel real-time algorithm performing sequentially is proposed to conduct this operation.

This method, depicted in Figure 22, starts by applying a dimensionality reduction technique to the data.

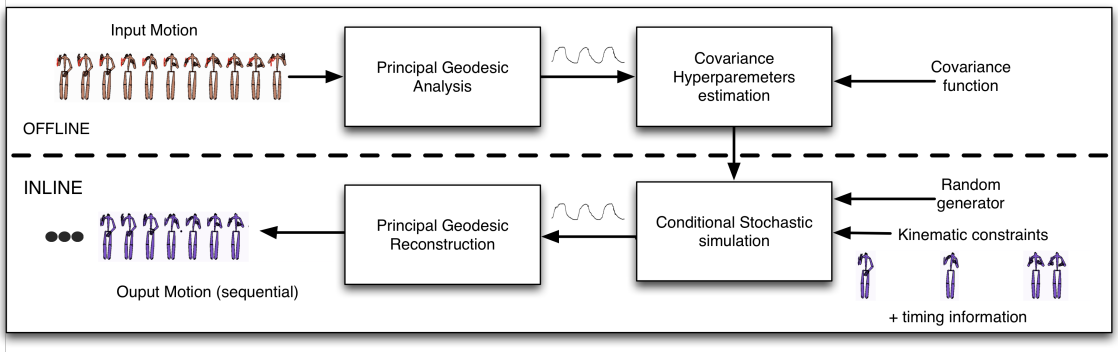


Figure 16.: **Overview of the proposed method.** During an offline phase, an example motion is first decomposed with principal geodesic analysis. The resulting trajectories are used to estimate the hyperparameters of a given covariance function. At runtime, the conditional stochastic simulation uses this covariance function, a random generator and some constraints to produce a new motion.

3.2.1. Prediction from Gaussian processes

Given p observations $X(t_1), \ldots, X(t_p)$ at times $t_1, \ldots, t_p$ of a given Gaussian process with known mean and covariance function C, one can look at the prediction of $X(t)$ for a given time t. In this section, we show that the Kriging approach and the Gaussian Process regression method solve this problem in the same way.

3.2.1.1. Kriging

Kriging [110] is a linear interpolation method issued from the geostatistical community. Mukai and Kuriyama [14] used this technique in the context of computer animation to find an optimal set of weights for blending motions. In the kriging approach, the estimation $\hat{X}(t)$ is expressed as a linear combination of the p known values $X(t_1), \ldots, X(t_p)$ as follows:

$$\hat{X}(t) = \sum_{i=1}^{p} \lambda_i(t) X(t_i), \tag{3.14}$$

where $\boldsymbol{\lambda}(t) = (\lambda_1(t), \ldots, \lambda_p(t))^T$ stands for the kriging coefficients.

It is possible to express those coefficients with the following equation:

$$\boldsymbol{\lambda}(t) = \Sigma_{(p)}^{-1} \Sigma_{(t)}, \tag{3.15}$$

where:

$$\Sigma_{(p)} = \begin{pmatrix} C(t_1, t_1) & \cdots & C(t_1, t_p) \\ \vdots & \ddots & \vdots \\ C(t_p, t_1) & \cdots & C(t_p, t_p) \end{pmatrix} \tag{3.16}$$

and:

$$\Sigma_{(t)} = (C(t, t_1), \ldots, C(t, t_p))^T. \tag{3.17}$$

These coefficients are obtained under the constraints that the estimation is unbiased and that the variance of the kriging error given by:

$$Var(X(t) - \hat{X}(t)) = C(t,t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)} \tag{3.18}$$

is minimized.

3.2.1.2. Gaussian Process regression

A similar approach is known as Gaussian Process (GP) regression in the machine learning community and vision communities [111]. The GP approach aims at solving the same prediction problem: given p observations $\mathbf{X}_{(p)} = (X(t_1), \ldots, X(t_p))^T$, one looks at the estimation of $X(t)$ at a given unobserved time t. GP's approach solve this problem using the assumption that the process is Gaussian, and building the conditional distribution $p(X(t)|\mathbf{X}_{(p)})$ which is itself Gaussian. The joint distribution of $X(t)$ and $\mathbf{X}_{(p)}$ writes indeed:

$$\begin{bmatrix} \mathbf{X}_{(p)} \\ X(t) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{(p)} & \Sigma_{(t)}^T \\ \Sigma_{(t)} & C(t,t) \end{bmatrix}\right) \tag{3.19}$$

where $\Sigma_{(p)}$ and $\Sigma_{(t)}$ are defined by (3.16) and (3.17).

The conditional distribution $p(X(t)|\mathbf{X}_{(p)})$ is then obtained from a little matrix algebra [111], and it comes that this distribution is Gaussian described by:

$$p(X(t)|\mathbf{X}_{(p)}) \sim \mathcal{N}(\Sigma_{(t)}^T \Sigma_{(p)}^{-1} \mathbf{X}_{(p)}, C(t,t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)}). \tag{3.20}$$

The mean $\Sigma_{(t)}^T \Sigma_{(p)}^{-1} \mathbf{X}_{(p)}$ of this distribution is clearly the same as the kriging estimate in equation 3.14, and the variance $C(t,t) - \Sigma_{(t)}^T \Sigma_{(p)}^{-1} \Sigma_{(t)}$ corresponds to the variance of error given by (3.18). The Gaussian Process regression is then another expression of kriging.

3.2.2. Stochastic simulation

In the following we assume that Z is a Gaussian process with mean μ and covariance function C. The objective is to simulate trajectories $\mathbf{Z}^{(sim)} = (Z^{(sim)}(t_1), \ldots, Z^{(sim)}(t_N))$ of length N of this process. The trajectories have to be independant and respect the statistical properties of Z:

$$E(Z^{(sim)}(t)) = \mu \ \forall t, \tag{3.21}$$

$$Cov(Z^{(sim)}(t), Z^{(sim)}(t')) = C(t,t') \ \forall t, t'. \tag{3.22}$$

Knowing the covariance function C, the covariance of a trajectory $\mathbf{Z}^{(sim)}$ is then a matrix denoted $\Sigma_{(N)}$ of size $N \times N$, with:

$$\Sigma_{(N)} = \begin{pmatrix} C(t_1,t_1) & \cdots & C(t_1,t_N) \\ \vdots & \ddots & \vdots \\ C(t_N,t_1) & \cdots & C(t_N,t_N) \end{pmatrix} \tag{3.23}$$

3.2.2.1. Non-conditional stochastic simulation

In this section we present how to simulate a trajectory $\mathbf{Z}^{\text{NC}}$ respecting the properties (3.21-3.22). One possible and simple simulation method is based on the Cholesky decomposition of the covariance matrix $\Sigma_{(N)}$. We first sample a vector $\mathbf{y} = (y_1, \ldots, y_N)^T$ composed of N independant realizations of the standard Gaussian distribution, so that $\mathbf{y} \sim \mathcal{N}(0, \mathbf{I}_{(N)})$. Then we set:

$$\mathbf{Z}^{\text{NC}} = L_{(N)}\mathbf{y} + \mu, \tag{3.24}$$

where $L_{(N)}$ is obtained from the Cholesky factorization of the covariance matrix: $\Sigma_{(N)} = L_{(N)} L_{(N)}^T$ (provided that $\Sigma_{(N)}$ is positive semi definite). From this decomposition it is easy to verify that $E(\mathbf{Z}^{\text{NC}}) = \mu$ and $Cov(\mathbf{Z}^{\text{NC}}) = \Sigma_{(N)}$.

One possible concern with this method is, from a computational point of view, the Cholesky factorization of $\Sigma_{(N)}$ which is $o(N^3)$. However, this operation can be conducted only once when $\Sigma_{(N)}$ is known.

3.2.2.2. Conditional Stochastic Simulation

In some cases, it can be interesting to force the simulations to reach given values $Z(t'_1), \ldots, Z(t'_p)$ (experimental data, keyframes specified by animators, etc.) at given time instants $t'_1, \ldots, t'_p$. In this section we explain how to respect these constraints while maintaining properties (3.21-3.22).

One could think of simulating new trajectories using the kriging estimate (equ. (3.14)) or sampling from the posterior defined by the GP regression (equ. (3.20)), for all times t between observed values [111]. Resulting trajectories would then reach observed values. However, these methods do not create trajectories respecting the property (3.22). The covariance structure is indeed not respected, and simulated trajectories are then smoother than those simulated with the right covariance structure C.

Note that recently, a method to sample new trajectories solving a global maximum a posteriori estimation conditioned to observed valued has been proposed by [23]. However, with such an approach there is no guarantee neither that the statistical properties of the reference motion are preserved.

A possible way to obtain trajectories that both respect the required covariance property and reach fixed values is to use a double kriging operation [112]. Let us recall that the simple kriging allows to find an estimate $\hat{Z}(t)$ at time t that differs from the unknown $Z(t)$ by the kriging error $Z(t) - \hat{Z}(t)$. This error is unknown but can be simulated by means of a secondary process having the same properties as Z. A trajectory $\mathbf{Z}^{\text{NC}} = (Z^{\text{NC}}(t_1), \ldots, Z^{\text{NC}}(t_N))$ is first simulated using the non-conditional simulation technique described in the previous subsection. A new trajectory $\hat{\mathbf{Z}}^{\text{NC}} = (\hat{Z}^{\text{NC}}(t_1), \ldots, \hat{Z}^{\text{NC}}(t_N))$ is then obtained by the kriging approach, from all values $Z^{\text{NC}}(t'_1), \ldots, Z^{\text{NC}}(t'_p)$. The resulting kriging error $Z^{\text{NC}}(t) - \hat{Z}^{\text{NC}}(t)$ for each t is finally added to the trajectory $\hat{\mathbf{Z}} = (\hat{Z}(t_1), \ldots, \hat{Z}(t_N))$ obtained from the kriging based of the given values $Z(t'_1), \ldots, Z(t'_p)$:

$$Z^{\text{C}}(t) = \underbrace{\hat{Z}(t)}_{\substack{\text{Kriging} \\ \text{estimate}}} + \underbrace{Z^{\text{NC}}(t) - \hat{Z}^{\text{NC}}(t)}_{\substack{\text{Kriging} \\ \text{error}}} \ \forall t \tag{3.25}$$

We can directly observe that the trajectory $\mathbf{Z}^{\text{C}}$ goes through fixed values $Z(t'_1), \ldots, Z(t'_p)$, since the kriged trajectory $\hat{\mathbf{Z}}^{\text{NC}}$ goes through fixed values $Z^{\text{NC}}(t'_1), \ldots, Z^{\text{NC}}(t'_p)$:

$$Z^{\text{C}}(t'_i) = \hat{Z}(t'_i) + Z^{\text{NC}}(t'_i) - \hat{Z}^{\text{NC}}(t'_i) \tag{3.26}$$

$$= Z(t'_i) \ \forall t'_i \in t'_1, \ldots, t'_N \tag{3.27}$$

Moreover, it can be proved that $\mathbf{Z}^{\text{C}}$ respects both properties (3.21) and (3.22). The resulting simulation is then a sample from a Gaussian process with the required covariance structure C, and that is constrained to go through particular values $Z(t'_1), \ldots, Z(t'_p)$.

The algorithm that sums up this conditional simulation technique is the following:

The main computational time is spent in the Cholesky decomposition since this operation is $o(N^3)$. When N is large, this can become a problem. In the context where N is not known, or if a continuous output stream is desired (in order to produce a virtually infinite random sequence), an alternative algorithm can be used. Let us first remark that the Cholesky decomposition produces a matrix L which is lower triangular. This mean that the p-th output of the simulation depends on

Algorithm 3.1: Compute trajectory $\mathbf{Z}^{\mathrm{C}} = (Z^{\mathrm{C}}(t_1), \ldots, Z^{\mathrm{C}}(t_N))$

Require: Covariance structure C of the process
Require: $Z(t'_i)$ at $t'_i = t'_1, \ldots, t'_p$
From C compute the $N \times N$ covariance matrix $\Sigma_{(N)}$
$L_{(N)} = \text{Cholesky}(\Sigma_{(N)})$
Simulate $\mathbf{Z}^{\mathrm{NC}}$ using $L_{(N)}$ with equation (3.24)
Estimate trajectory $\hat{\mathbf{Z}}^{\mathrm{NC}}$ from $\Sigma_{(N)}$ and fixed values $Z^{\mathrm{NC}}(t'_i)$ following the kriging equation (3.14)
Estimate trajectory $\hat{\mathbf{Z}}$ from $\Sigma_{(N)}$ and fixed values $Z(t'_i)$ following the kriging equation (3.14)
return $Z^{\mathrm{C}}(t) = \hat{Z}(t) + Z^{\mathrm{NC}}(t) - \hat{Z}^{\mathrm{NC}}(t)\ \forall t = t_1, \ldots, t_N$

the last $p-1$ elements that were drawn from the standard Gaussian distribution. This p-th output can thus be computed provided that the p-th line of L and the past elements are known. However, it is noticeable that the Cholesky decomposition has a recursive formulation, that makes possible to compute the p-th line from the $p-1$ previous lines in the matrix. Also, since the covariance function is assumed to be neglectful after a given distance ρ (corresponding to the length-scale), we can reasonably assume that the influence of known values $Z(t'_i)$ is neglectful whenever $|t'_i - t| < \rho$. By restraining the computation of each element $Z^{\mathrm{C}}(t)$ of the output as a function of sufficiently near $Z(t'_i)$, and by updating iteratively the p-th line of the Cholesky decomposition, it is possible to design an algorithm that produces sequentially a correct output:

Algorithm 3.2: Compute trajectory $\mathbf{Z}^{\mathrm{C}}$ sequentially

Require: Covariance structure C of the process
Require: $Z(t'_i)$ at $t'_i = t'_1, \ldots, t'_p$
$\mathbf{y} \leftarrow \text{FIFO}(2\rho)$ {$\mathbf{y}$ has a FIFO structure of size 2ρ}
$\mathbf{Z}^{\mathrm{NC}} \leftarrow \text{FIFO}(2\rho)$ {and so $\mathbf{Z}^{\mathrm{NC}}$}
$t \leftarrow 1$
repeat
  $L^t_{(\rho)} = \text{updateCholesky}(L^{0:t-1}_{(\rho)})$
  $\mathbf{y} \leftarrow \text{push}(y_t \sim \mathcal{N}(0,1))$
  $\mathbf{Z}^{\mathrm{NC}} \leftarrow \text{push}(L^t_{(\rho)}\mathbf{y})$
  Estimate trajectory $\hat{\mathbf{Z}}^{\mathrm{NC}}$ from C and fixed values $Z^{\mathrm{NC}}(t'_i)$ (eq (3.14)), $\forall t'_i$ such that $|t'_i - t| < \rho$
  Estimate trajectory $\hat{\mathbf{Z}}$ from C and fixed values $Z(t'_i)$(eq (3.14)), $\forall t'_i$ such that $|t'_i - t| < \rho$
  return $Z^{\mathrm{C}}(t) = \hat{Z}(t) + Z^{\mathrm{NC}}(t) - \hat{Z}^{\mathrm{NC}}(t)$
  $t \leftarrow t + 1$
until needed

In this algorithm, `updateCholesky` allows to compute the t-th line $L^t_{(\rho)}$ of the Cholesky decomposition from all previous lines.

3.2.3. Application to character animation

We show two possibilities to exploit conditional stochastic simulation in the context of character animation. Other examples can be found in *[96]* . The first example shows how conditional simulation can be used to reconstruct missing or damaged parts of a motion; the second one presents possible applications in motion control.

3.2.3.1. Motion reconstruction

It is usual with traditional motion capture devices to encounter markers occlusions that alter the quality of the motion reconstruction. With markerless motion capture this problem is even more present as far as the complete pose estimation can fail for a more or less short period of time [113]. The objective is here to reconstruct the missing parts of the signal. Most of the classical approaches perform linear or spline interpolation between the known parts of the motion. In the case of large

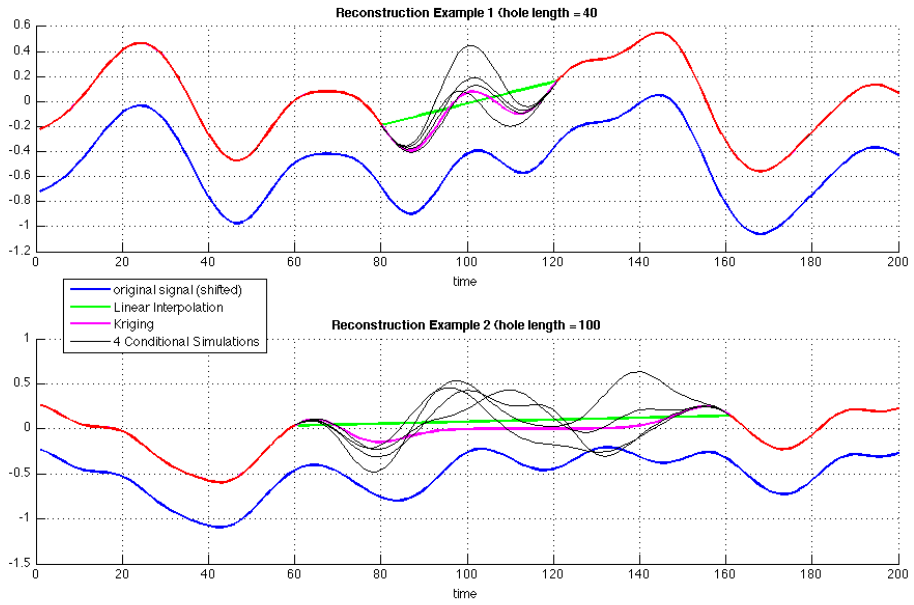


Figure 17.: **Hole filling using conditional stochastic simulation**: in this example the length-scale of the covariance function is around 10. When the size of the hole is 40, the simulation is very constrained and the variability is limited. Oppositely, when the hole is larger, our method provides different results with a greater variability, whereas the classical linear or kriged interpolate flatten the signal

holes, those types of interpolation behave badly as they tend to produce a continuous and smooth output which is generally different from the original motion dynamics. Our method first learns the covariance structure on the known parts of the motion and then simulates the unknown part of the motion conditioned to all known single frames.

Figure 17 presents an illustration of the reconstruction for two different hole lengths. One can see that for small holes, the variability between the different simulations proposed by our method is restrained, and that results are close to a simple kriging interpolation. For longer holes, the variability is bigger and results differ from the kriged solution. Far from observations, the kriging converges indeed toward a mean estimate, flattening the reconstructed part. On the other hand, each of the different trajectories simulated by the conditional approach is statistically coherent with the known part of the motion (which means here that the covariance structure of the whole reconstructed signal is the same than the one learned from the known part). Those proposed solutions might not correspond to the real motion, but can be used as credible, potential solutions.

3.2.3.2. Exemplary based motion control

We show here how conditional simulation can be efficiently used in the context of motion control. By motion control we mean that, given an exemplar motion, a new motion can be produced along with a set of kinematic constraints, and eventually timing information. Conditional simulation allows to derive an efficient, real-time motion synthesis process, which overview is depicted in Figure 18. Kinematic constraints, such as hands or feet positions are added to the system, along with timing information. The character pose is solved for by applying PGA-based Inverse Kinematics, which directly gives the corresponding coordinates in the PGA space. Then, a new motion is simulated over a time interval which is centered around the constraint time, and which length is twice the maximum among all estimated length-scales λ_i (which corresponds to the range of time dependance in the covariance model estimated for each PGA component). This interval contains indeed all poses that present significant time dependance with the new constraint and that have

then to be recomputed. This simulation is conducted conditioned to every other unchanged poses in the motion. This operation can eventually be processed sequentially.

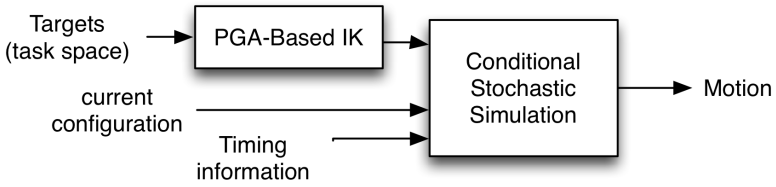


Figure 18.: **Using conditional simulation in the context of motion control**. A PGA-based Ik solver provides conditions directly in the PGA space. Along with the current configuration and timing information, a new motion can be generated

Figure 19 shows an example of this process. A baseball catch (motion 20 from subject 143 in CMU database) was used. A new catch pose is computed with PGA-based IK (figure 19.a). A new motion is then computed in its vicinity (the first PGA component is shown in figure 19.b). Two image strips showing rendering with a skinned character of both original and simulated sequences are shown (figure 19.cd).

Figure 19.: **Motion control**. This example handles a baseball catch motion. Figure (a) presents the original catch and a new catch generated by PGA-IK (applied on both arms). Figure (b) shows the first component of the PGA with its new simulated part. Notice the time interval over which the simulation has been performed. Figures (c) and (d) illustrate respectively the original motion and the synthesized motion on four frames.

3.3. Composition methods

We will now describe an animation system dedicated to the production of a comprehensible signed language sequences. This system was created in the context of the Signcom project *[62]* . The *SignCom* interaction system is divided into two parts: an off-line process of data storage and on-line data retrieval for real-time interaction. The originality of the work presented here originates in the methodology used for data storage and in the streaming method used to retrieve motion data. Our system provides fast and efficient motion retrieval during the animation process, taking into consideration the spatial and temporal aspects of signed language motion described above. The nature of the different types of information encoded in and by signs makes it necessary to store data in two different structures, namely a semantic database for textual annotations, and a raw database for motion capture data.

The process begins with a list of motion elements paired with timing information, retrieved from two different databases that contain semantic (annotation) and raw (motion capture) data (Section 3.3.1). Then our multichannel composition system builds a new motion expressed as a sequence of skeletal postures (Section 3.3). These postures contain information that encodes body and hand configurations as well as facial markers. Next, the facial markers are turned into a new geometric facial configuration by means of blendshapes and a learning method (Section 3.3.3); eye animation is also inferred from this skeletal posture (Section 3.3.4). Finally, the rendering engine computes the final avatar image.

3.3.1. Data Coding and Retrieval

As presented in the beginning of this document, the *SignCom* interaction system is divided into two parts: an off-line process of data storage and on-line data retrieval for real-time interaction. The originality of the work presented here originates in the methodology used for data storage and in the streaming method used to retrieve motion data. Our system provides fast and efficient motion retrieval during the animation process, taking into consideration the spatial and temporal aspects of signed language motion described above. The nature of the different types of information encoded in and by signs makes it necessary to store data in two different structures, namely a semantic database for textual annotations, and a raw database for motion capture data.

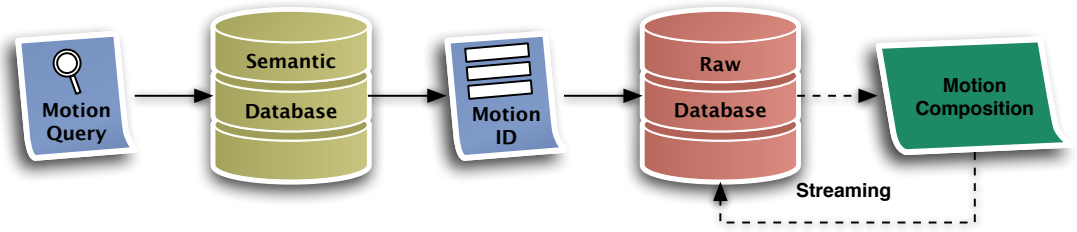


Figure 20.: Data retrieval and stream loading system. The semantic database, containing textual information from the annotation process, is queried first. The motion data corresponding to the obtained results are then streamed to the motion composition process.

As depicted in Figure 20, retrieving data from the databases is divided into two parts. The first part of the process consists of querying the semantic database, allowing us to extract data corresponding to a list of MotionIDs. In a nutshell, these MotionIDs represent the canonical index data structure of a motion element. Each contains the name of the sequence wherein the chunk occurs, time stamps relative to the beginning of this sequence, (noted as Frame In and Frame Out), and the involved body parts. This mapping between the annotation and motion data constitutes the semantic database (Figure 21), which is automatically constructed from an XML hierarchical description language provided by the annotation tool (ELAN in our case). We emphasize here the one-to-many nature of this mapping, where any one gloss from the textual annotation can be associated with several different realizations of the same gesture. As one example, the gloss COCKTAIL in Figure 21 corresponds to two MotionIDs, 1 and 4.

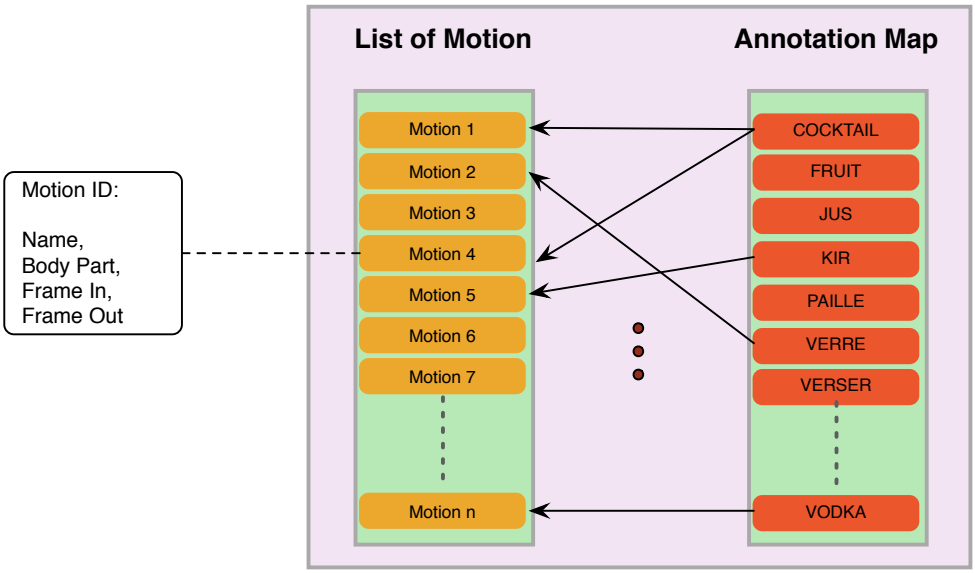


Figure 21.: The semantic database is a one-to-many mapping between annotated glosses and MotionIDs, which are canonical index data structures of motion elements.

In our application, retrieving data from the semantic database is achieved by specifying multiple-condition queries, the conditions of which can be keywords and/or body parts, and which return one or several MotionIDs. Secondarily, the query results are interpreted so that each MotionID leads to accessing the raw database and rendering the corresponding motion frames.

Raw motion database. Motions are traditionally stored on a hard drive in various format (.bvh, .fbx, .asf/amc, etc.). Interpreting these files amounts to building an internal representation of the motion in CPU memory. In our system, this internal representation contains an association of the hierarchical structure (commonly called a bindpose), and a list of relative transformations for each joint. The transformation for the root joint contains joint position and rotation (expressed in quaternions), while the transformation for the rest of the joints contains only a rotation. The time needed to read a motion file into this internal representation depends naturally on the complexity of the parser and the amount of geometrical computations, and is usually far from being negligible, preventing dynamic loads in our interactive application. Motion files are thus loaded and interpreted one time, and stored as a sequence of bits in our database, having written our own serialization process for this purpose.

Traditional databases function with a set of pair-valued data: one key (preferably unique) is associated to the useful data (in our case the motion). The simplest way to proceed is to associate for instance the whole motion file with a unique key, which can be chosen as the name of the original data file. The whole sequence is then handled by the database manager, and stored on the hard drive. This approach assumes that when retrieving the motion, all the data will be reconstructed in the CPU memory. In the context of a real-time animation controller, where small pieces of the motion are dynamically combined to achieve a desired goal, this approach is no longer efficient. We have designed our database therefore to handle a different data representation, allowing us to retrieve any part of a motion corresponding to a given annotation element (Figure 22).

Decomposing motions in the database is innovative because only a small portion of the motion (associated to a query result) is reconstructed in the memory. However, in traditional databases, data decomposition generally yields an increasing number of entries, which most of time increases the search time and the index size. Yet in our case we consider each motion to be a list of transfor-

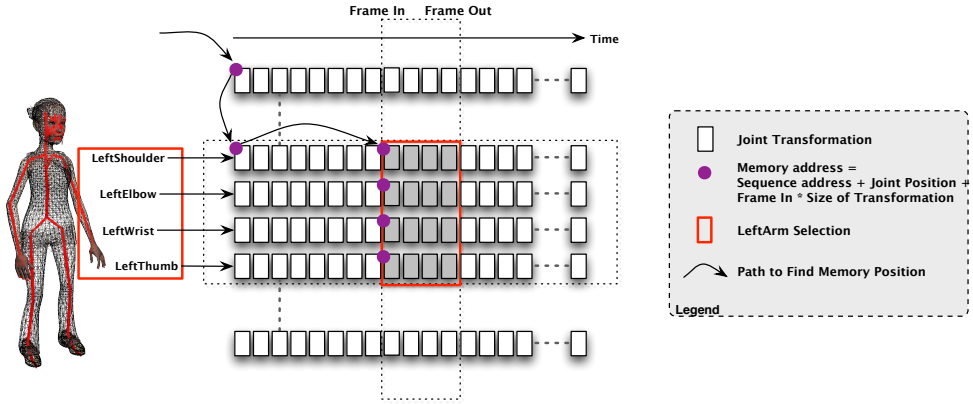


Figure 22.: Storage and data access in the raw database

mations with given sizes; therefore it is easy to find the memory address of a list of transformations as a linear combination of the sequence address, joint offset, and time stamps, as illustrated as a path in Figure 22.

To complete the access to the raw motion capture data, we have developed a streaming system which loads the motion to animate in a fragment-by-fragment manner during the animation process (a fragment being a small set of transformations), and with regards to the need of the motion composition system. This avoids costly access to large elements which could result in a drop in frame rate during the execution of the application, and gives the process a small memory footprint. Computationally, this allows the interactive nature of this animation system to move forward, since database search and data load time become negligible during animation.

3.3.2. Motion Composition

From our corpus of mocap data, our animation system computes a skeleton using a pre-defined morphology of joints and bindposes, which can be represented hierarchically as a tree of joints or articulations. Within the skeleton, we have identified sub-skeletons composed of potentially non-exclusive subsets of joints, including the upper body, lower body, arms, hands, head, etc. A controller associated to each sub-skeleton can set the system in motion using different techniques, i.e., motion playback, keyframe interpolation, inverse kinematics, etc.

The motion composition process can be divided into spatial and temporal composition processes. The spatial composition process uses motions computed for each controller's sub-skeleton, combining them in a priority scheme that depends on the desired animation; generally, the smaller sub-skeletons have a higher priority level, as shown in Figure 23. Temporal composition occurs for the set of controllers attached to the skeletal elements. Each controller has its own timing interval and a playback style (e.g., play once, repeat, reverse, etc.), and the blender process is responsible for blending the motions.

Figure 23 is a graphical representation of how we organize blenders and controllers during composition. Finally, we have developed a simple script language in order to easily specify different animation scenarios, containing controller and blender information associated with time stamps.

3.3.3. Facial Animation

Facial animation by blendshapes is a popular technique in the animation community, and we have chosen likewise. Following this method, the animation system blends several key facial con-

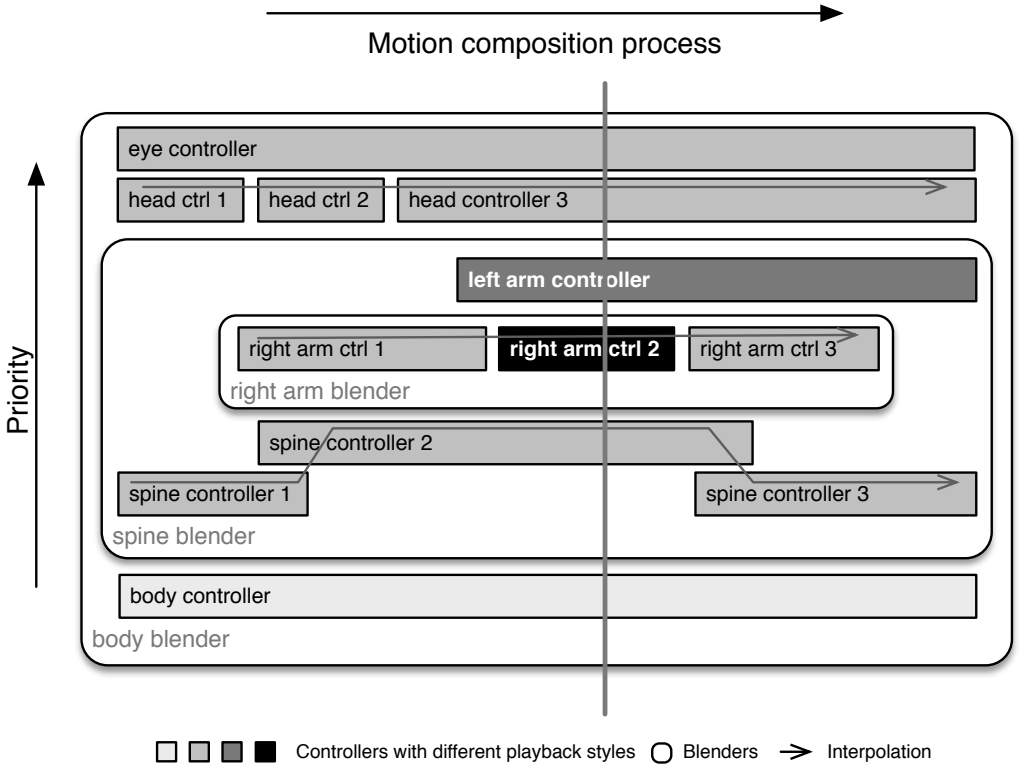


Figure 23.: Blenders are arranged hierarchically in the system and contain a series of controllers to animated different sections of the body. Skeletons are computed according to priority of controllers and, over time, the engine produces a stream of fluid motion.

figurations, manually designed by an animator, to produce appropriate facial animations. In order to choose the blending weights at each moment, the system uses the facial mocap data contained in the currently-processed skeleton, as described below.

Cross-mapping of facial mocap data and blendshape parameters. The process of cross-mapping mocap data and blendshapes parameters can be problematic for the animation process: it is often challenging to quantify the relation between facial mocap data and the animation parameters of a blendshape. Traditional approaches to solving this problem identify pairs of mocap data and blendshape parameters that are carefully selected and designed by the animator [60]. These pairs are then used in a learning process that determines the selection of corresponding blendshape parameters from new mocap data input values. More current methods usually rely on radial basis functions and kernel regression to achieve these steps [114, 115, 60, 116].

However, such methods have several drawbacks: a number of localized basis functions have to be chosen prior to the learning process, and the result is conditioned by the quality and density of input data. Thus, noisy input often yield bad estimates, this being known as the classical over-fitting problem.

In our work, both the body and facial data were recorded at the same time, and the positions of the facial markers in particular were observed to be quite noisy, resulting in marker inversions. For these reasons, we consider the problem as a probabilistic (Bayesian) inference problem and use a separate learning technique based on Gaussian Process Regression. In our approach, unknown sites correspond to new facial marker configurations (as produced by the previously described composition process), and the corresponding estimated value is a vector of blendshape weights. Since the dimensions of the learning data are rather large (123 for marker data and 50 for the

total amount of blendshapes in the geometric model we used), we rely on an online approximation method of the distribution that allows for a sparse representation of the posterior distribution [117]. As a preprocess, facial data is expressed in a common frame that varies minimally with respect to face deformations. The upper-nose point works well as a fixed point relative to which the positions of the other markers can be expressed. Secondly, both facial mocap data and blendshape parameters were reduced and centered before the learning process.

Figure 24 shows an illustration of the resulting blended faces along with the different markers used for capture.

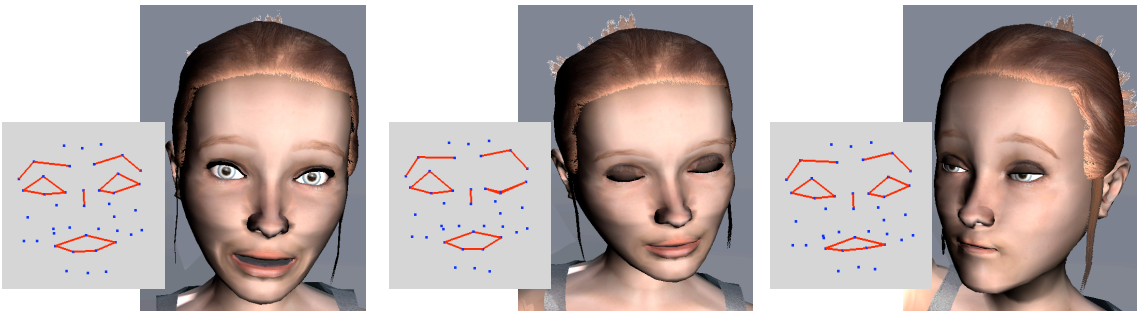


Figure 24.: Results of the facial animation system. Some examples of faces are shown, along with the corresponding markers position projected in 2D space.

3.3.4. Eye Animation

Our capture protocol was not able to capture the eye movements of the signer, even though it is well-known that the gaze is an important factor of non-verbal communication and is of assumed importance to signed languages. Recent approaches to model this problem rely on statistical models that try to capture the gaze-head coupling [118]. However, those methods only work for a limited range of situations and are not adapted to our production pipeline.

Alternatively we use a heuristic synthesis model that takes the neck's motion as produced by the composition process as input and generates eye gazes accordingly. First, from the angular velocities of the neck, visual targets are inferred by selecting time instants when the velocity passes below a given threshold for a given time period. Gazes are then generated according to those targets such that eye motions anticipate neck motion by a few milliseconds [119]. This anticipatory mechanism provides a baseline for eye motions, to which glances towards the interlocutor (camera) are added whenever the neck remains stable for a given period of time. This ad-hoc model thus integrates both physiological aspects (modeling of the vestibulo-ocular reflex) and communication elements (glances) by the signer. Figure 25 shows two examples of eye gazes generated by our approach.

Figure 25.: The two types of glances produced by our system (a) direct look to the interlocutor (b) anticipation of the neck rotation

3.4. Summary

In this Chapter the problem of motion control and synthesis was exposed. Three ways of producing new unobserved motions were presented, with different level of data-models coupling. The first one was a new inverse kinematic methods, which allows to integrate prior knowledge on the motion or likelihood of the poses in a Bayesian inference setting, but which can be used without any prior knowledge. The second one is based on conditional stochastic simulation, which firstly models the motion as a stochastic process, and then provide a way to sample new motion realizations under user specified constraints. The last presented method is far more close to the data, in the sense that it produces a new motion as a combination of existing motions. Here, the use of semantic, high-level information, to conduct this composition was exposed.

4

Perspectives and ongoing works

Regarding the different requirements exposed in the previous Section, several unresolved computer animation problems are presented here. Those problems are not particularly exclusive to the animation of virtual signers, and can address more widely general virtual character animation problems.

Evaluation and perceptive measures. The evaluation of the proposed animation methods is a crucial point to assess the quality of the produced animations. We have conducted some user studies *[120, 62]* , but we acknowledged that the validation of the animation protocols is far from sufficient. In the case of the virtual signer *[62]* , the user study has revealed a lot of defects in our signer, mostly because of its graphical appearance and because of unachieved finger contacts in hand poses. This establish the fundamental problem of how to decouple and evaluate separately the graphical content and the animation methodology [121]. Indeed, what makes a motion being a realistic motion still needs to be defined, but it is clearly a combination of visual appearance details and realistic temporal trajectories. New evaluation methodologies should be designed in this direction to handle properly this two aspects. Also, most of the studies on visual perception of motions have focused either on the naturalness (is a motion plausible [122] ? Is it physically realistic [123] ?) or the perception of diversity [124, 125]. SL have this advantage to make a semantic evaluation possible (how much of the discourse has been understood ?).

High frequency full body and facial motion capture. Signs are by nature very dexterous and quick gestures, that involve at the same time several modalities (arms, hands, body, gaze and facial expressions). Capturing accurately all these channels with an appropriate frequency (> 100 Mhz) actually pushes motion capture equipment to their very limits. It could be argued that splicing methods such as [126] would allow to capture independently the different modalities, and then combine them during a post process phase. However, the temporal synchronization issues raised by this method seem hard to alleviate. Moreover, asking the signer to perform alone the facial expressions corresponding to given sentences is also out of reach, since most of the facial mimics are generally done unconsciously. A parallel could be drawn with non-verbal communication: could we ask someone to perform accompanying gestures of an unspoken discourse ? Finally, new technologies such as surface capture [127], that captures simultaneously geometry and animation, are very attractive, but yet the resolution is not sufficient to capture the body and the face with an adequate precision, and only very few methods exist to manipulate this complex data in order to produce new animations.

We have recently started a new project (SIGN3D) in this direction together with an enterprise specialized in motion capture, which aims at capturing the motions of a signer with a high temporal and spatial resolution. We hope to measure improvements over the last version of our virtual signer in the acceptability of the system.

Expressivity filtering. As seen in the previous Section, the spatio-temporal variability of signs can be used as adjectives, or in a more general way, to inflect the nature of a sentence and enhance the global expressivity of the virtual signer. It has been shown [120] that temporal alignment methods [89] can be efficiently used to change the style and expressivity of a captured sentence. Nevertheless, big variations in style are can not only obtained by changing the timing of gestures, but most often by the change of spatial trajectories, and sometimes may inflect the entire sentence. Most of existing methods that build statistical models [30] of gestures may fail for this purpose, mostly because the style transfer is encoded by higher level linguistic rules, and because pure signal approaches are insufficient to model this variability. Hierarchical models encoding both a semantic and a signal level knowledge, would be desirable at this point.

Advanced motion retargeting. Most of the actual motion retargeting techniques focus on the adaptation of motion to changing the physical conditions of the motion [106] or more frequently kinematic constraints [128, 129, 130] through the use of inverse kinematic techniques. In the case of sign language the spatial relations between the fingers and the arms or the head are key elements for the comprehension of the discourse and should be preserved in the retargeting process. To this end, the recent work of Ho and colleagues [131] is really attractive, provided that the important relation between limbs could be preserved by their methods. Yet Its application to sign language synthesis remains to be explored. Whereas interaction with the floor or objects in the environment lead to hard constraints which lead to difficult optimization problems and procedures, constraints in sign language may be more diffuse or expressed qualitatively (e.g. "the thumb should touch the palm of the hand"). Algorithms dealing with such fuzzy or high level constraints could be extremely interesting, both numerically (more degrees of freedom while optimizing) and from a usability point of view. Finally, since arms motions are involved, a planing phase may also be required to avoid self collisions. Combined inverse kinematics and planing algorithms could be used [132], as well as more recent hybrid approaches [133]. Yet, real time algorithms for this class of problems remain to be found.

This is the subject of the ongoing phd thesis of thibaut LeNaour. In order to solve this problem, thibaut has started to explore new geometrical representations of the motion [134].

Multichannel combinations. As exposed in [44], the possibility of building new signed utterances by composing selectively pre-exisiting elements of a corpus data is possible. In this option, not only the spatial coherency should be preserved, but as well the channel's temporal synchronization:

- *spatial coherency.* Sign language allows to combine different gesture with different meanings at the same time, thus providing several information in a minimum of gestures. This combination differs from the classical blending approaches which mix motions together to produce new ones [61], as far as topological constraints should be preserved in the composition process. An example is given in Figure 26, where the same pose indicates at the same time that a dog is looking at (first sign) something while slobbering (sign 2). If both signs were to be recorded independently, a naive blending operation would fail because the hand would not anymore be located in front of the mouth. Moreover, as exposed in the previous Section, every spatialized gestures should be retargeted with respect to the current signing space. This brings us back to the problem of advanced motion retargeting, but also clearly reveals that the combination process should be driven by more abstract definition, possibly of linguistic nature.
- *temporal synchronization.* It is likely that the different motion elements have not the same duration. The consequent problem is twofold: *i)* a common timeline has to be found, eventually as the result of a combinatorial optimization, or driven by linguistic rules. Up to our knowledge though, no existing model of sign language describe such temporal rules or model the synchronization of the different channels *ii)* once a correct time plan has been devised, the temporal length of the motion chunks has to be adapted, while preserving the dynamic of

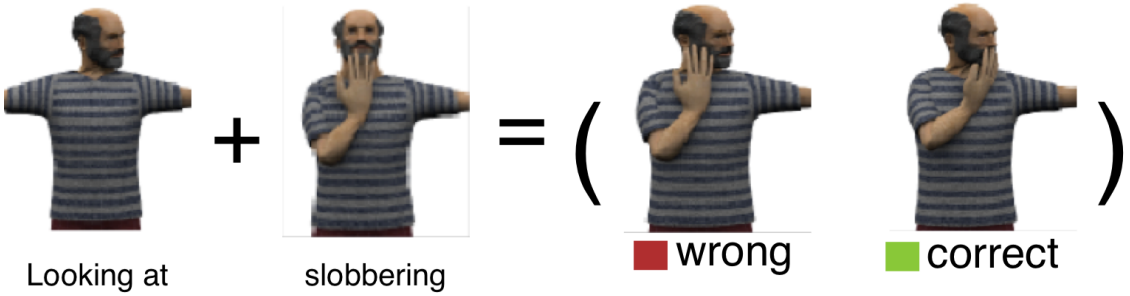


Figure 26.: Combination of two signs ("looking" and "slobbering")

the motions. To this end, time warping techniques can be used [89]. However, inter channels synchronizations may exist (for example between the hand and the arm motions [135]). Those synchronization schema can be extracted from analysis, but the proper way to introduce this empirical knowledge in the synthesis process has not been explored yet.

Part II.

Data-driven crowd analysis and synthesis

5

Preamble and application context

Contents

In this part of the document we switch to a different subject of interest: crowd phenomena. The study of crowd motions is an important field which gathers multi-disciplinary skills and which applications cover the production of visual effects, surveillance, urban environmental monitoring, structure and building simulation and architectural design, sociology and finally people behavioral analysis.

We recognize in this topic the same duality between analysis and synthesis schemes: most of the empirical studies on crowds tend to model the behavior of the crowd in order to explain or predict critical events, and as those models are being built upon observations on the real world, acquiring valuable information on those types of phenomena calls for dedicated and possibly automated procedures. Also, the video analysis of people in crowd is of straightforward importance, to understand single and social behaviors, to detect anomalies and suspicious events or objects in crowds scenes, to define first-aid and crisis support in areas where big events (stadium, sport exhibitions, concerts, large shows, political demonstrations..) are organized. This level of analysis is also greatly helped by a priori knowledges on what is observed (e.g. an abnormal situation will be defined as the opposite of a normal situation, but how to define normality ?). Hence, most of the recent analysis methods incorporate a "model" of the crowds.

Our work in this direction addresses both the analysis and the simulation issues, and as such belongs to the category of data-driven methods. We will start this chapter by giving a brief state-of-the-art of existing methods in crowd simulation and control (Section 5.1). Then we will give the focus on crowd analysis with vision techniques (Section 5.2). Section 5.3 will present more specifically data-driven approaches in the context of crowd.

5.1. Crowd simulation and control

Simulating a crowd of thousand of individuals is a challenging task. Most of the time, human crowds exhibit very subtle and specific patterns. The variability of crowd dynamics and behaviors is a consequence of the diversity of the persons inside it (age, sex, social and psychological attributes),

as well as the spatial configuration of obstacles and lanes. We first begin by giving the main approaches of crowd simulation and control, and their potential links with data-driven approaches. Again, our objective here is not to give a detailed state-of-the-art on the subject. Readers can refer to good reviews such as [136] or [137].

5.1.1. Crowd simulation

Simulating crowd of individuals has drawn a lot of attention over the past decades for the potential interests of computer graphics, but also for safety engineering or robotics applications. The different models are commonly divided into two categories: microscopic and macroscopic. **Microscopic approaches** tend to model members of the crowds as agents with specific behaviors. Sophisticated behaviour models seek autonomous agents endowed with goals and specific attributes [138, 139], but for a somehow limited number of individuals. Oppositely, their motions can be the result of simple laws, such as in the seminal work of Reynolds on flocking [140]. Designing and tuning these laws is now known as the steering problem, for which several solutions exist thanks to different strategies; for examples interacting particles under psychosocial forces [141], reproducing experimental observations [142], principle of least effort [143] or vision based strategies [144]. The most recent methods allow to simulate large scale crowds at interactive framerates with convincing emergent behaviors of the groups [145]. Conversely, **macroscopic models** generally consider the crowd as a whole and model its dynamic by means of continuum mechanics equations, allowing analogies with the domain of computational fluid dynamics [146, 147, 148, 149]. This type of modelling works well with dense crowds where the weight of individual decisions is somehow weakened, but fails to describe realistic interpersonal collision avoidance behaviors or heterogeneous crowds with individuals exhibiting distinct goals or motivations.

5.1.2. Simulation Control

Controlling a crowd to achieve a given effect is a rather difficult task, mostly because the only control parameters are those of the simulation model, which are generally not designed for it. Ulicny and colleagues [150] are the first to describe an interactive tool to design crowd scenes in an intuitive manner using a brush metaphor. With regards to the control of the pedestrian trajectories, existing solutions usually assume that individuals are driven by a given steering strategy combined with an ambient velocity field which is usually referred to as a flow or navigation field [151, 152, 153, 154]. In [152], Jin and colleagues define those fields as a combination of radial basis functions defined by the user. Park [153] defines control flows attached to special particles which motions can be keyframed during the simulation. Patil *et al.* define their navigation field with a sketch based interface or by extracting flow fields from videos, in a way similar to *[155]* . Other approaches consider the spatial relationships of the crowd members (coded as a graph) as an important feature to preserve, then use spectral interpolation methods [156] or mesh deformation techniques [157] to edit existing crowd animations.

5.2. Vision-based techniques for crowd phenomena

This section investigates the principal axes carried out by researchers to deal with crowd video analysis. A full review is out of the scope of this document and for that purpose, we refer the readers to a good review [158] and to associated references. As mentioned above, the inherent diversity and complexity of the behavior of a mass of people makes ambiguous the question of representing and modeling a crowd. In the two last decades, two main strategies have been carried out by the different authors:

1. representation at the *pedestrian* level: each individual is an entity/particle driven in a *Lagrangian* framework. The crowd representation results from the combination of a large

number of entities. In the following we denote as "Lagrangian" such approaches;

2. the crowd is modeled in a *continuous* framework related to some scalar/vectorial characteristic quantities (density or displacement field for instance). The governing equations are represented in an *Eulerian* context and the individual notion vanishes. We denote as "Eulerian" such techniques.

5.2.1. Crowds as a set of individuals

Within this class of method, the scene is modeled as a collection of pedestrians that interact with their environment (obstacles). In general, the associated analysis techniques rely on low-level vision (background subtraction, edge and object detection) that enables a human counting and eventually an action recognition. The former appearance models have been based on low level features like an edge map [159, 160, 161]. This latter is used afterward within a more advanced strategy, like a neural network or a probabilistic tracking approach [162], to segment, regroup and evaluate the number or individuals. In a step forward, some human detection approaches have been defined at an object level. The humans are first detected with an ad-hoc technique of head or body recognition. This relies on a model either based on the appearance of the humans (distribution of some functions based on the luminance) [163, 164, 165], on a 3D body model [165] or on the velocity of entities [166]. An additional post-processing step for classification (clustering, SVM, ...) enables to count and sometimes to characterize the nature of the motion.

Even if some of the mentioned technique exhibit very competitive results, when a large number of pedestrians are present in the crowd (> 50), most of the conventional tracking methods (like Kalman trackers) fail. In such cases, the degradation of the visual features related to single individuals disturbs the analysis. Moreover, the large induced state space yields computationally too expensive problems. In those situations, the Lagrangian approach fails and the analysis of the crowd sequence may amount to the analysis of a crowd flow that have global properties and may be treated as a whole. As an example, one can cite the recent work of Rodriguez and al. [167] who use the global motion of the crowd (the crowd behavior) to help the tracking.

5.2.2. Crowds as a continuous entity

The representation of crowd flows in a pure Eulerian approach has been studied in [146]. In this study, the author creates some links between dense crowds and fluid mechanics laws. Two flow regimes (e.g. high-density and low-density) have been proposed in a complete dynamical model that depends on some objective parameters (optimal orientation to reach the goal, pressure, velocity) and on some more subjective quantities like the crowd comfort. The experimental simulations have been successfully compared to some real scenes. In a context of crowd simulation, the authors in [168] present a real-time crowd model based on continuum dynamics. In *[169]* , Allain et al. have proposed a somehow simpler continuous dynamical model where pedestrians are assumed to reach an objective while interacting together in order to prevent from the formation of too dense areas. A disturbance potential has also been introduced to deal with more subjective interactions inside the flow. This dynamical model is then used as a prior information for analyzing crowded videos in an optimal control theory framework.

On the basis of this constatation that a crowd can be managed with continuous laws, several analysis techniques based on a continuum approach have been proposed [170, 171, 169, 172, 173, 174]. Related methodologies usually tend to solve the different problems of event detection or changes in the flow rate on the basis of the apparent motion (optical flow) estimated on the the whole image from the image luminance. The work of Ali and Shah [172] focuses on segmenting the crowd flow with regions of substantially different dynamics by examining the coherent structures in the flow. In [173], unsupervised feature clustering is used to define normal motion patterns, and Hidden Markov Models are used to detect particular situations. A similar approach has been

proposed in [170] where normal and abnormal behaviors are extract from the continuous optical-flow. This displacement indicator is also a prior descriptor to highlight circular and diverging flows in [174].

5.3. Data-driven crowd animation and simulation

Data-driven approaches aim at avoiding the defects of imperfect simulation models by describing directly the visible results instead of the underlying causes of the crowd's motion. The idea is to use a priori knowledges extracted from real situations, either to reproduce a global situation or local collision avoidance strategies. Several issues have to be treated:

- what kind of descriptors for crowd motions ? what should be the best mathematical tool to describe the dynamics of such a complex system with several parameters ?
- How to capture/estimate these descriptors from real situations (this point being closely related with the mathematical modeling of the crowd) ?
- How to apply/use them as an input for an interactive, controllable animation system ?
- Is crowd motion editing possible ? or finally, under which condition can we modify the original data to adapt to new situations ?

5.3.1. State-of-the-art of data-driven crowd simulation methods

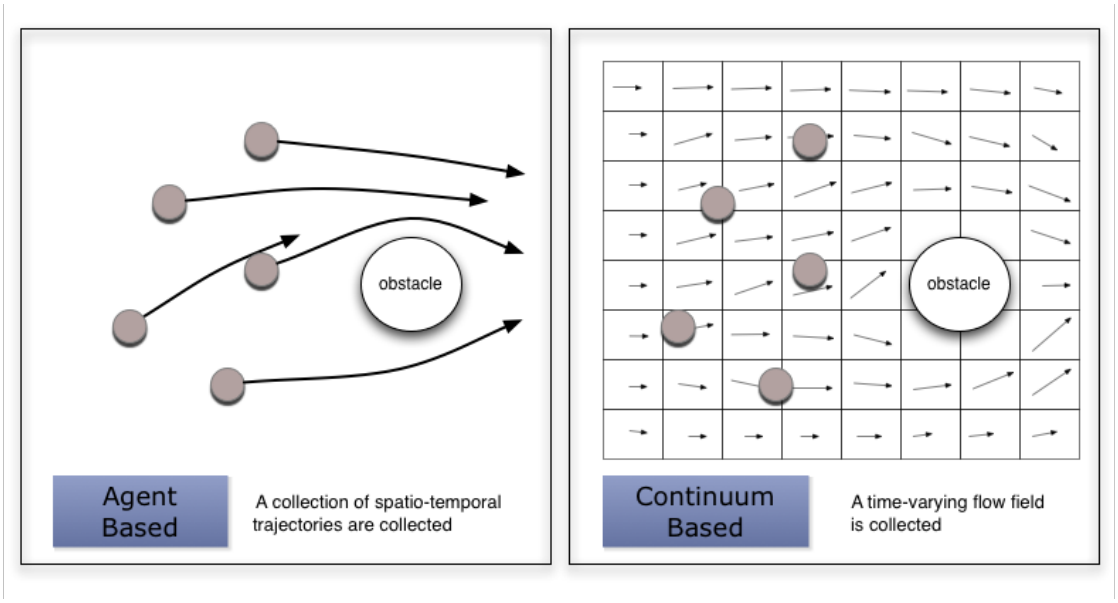


Figure 27.: **Differences between the two approaches of data driven crowd animation**.

Recently, several research groups have explored this approach of crowd animation [155, 142, 175, 176]. Paris and colleagues [142] use motion capture sets to capture the evolution of pedestrian along time. In [176], Lerner et al. use as a preprocess a manual tracking in the image space. Lee et al. [175] use a semi automatic method to label and track individuals in the video with a state-of-the-art kernel based approach. As stated by authors, this method is not robust for long duration, and requires several manual intervention along the sequence. We have proposed a fully automatic method in *[155]* , which however assumes that the considered crowd is dense. This approach will be explained in details in the next section. Concerning the synthesis part, Paris

et al. use the results of their capture to parameterize an ad-hoc collision avoidance algorithm. Lerner [176] and Lee [175] have a strongly similar approach to synthesize new situations from their example database: a local neighborhood representation is used as example query in the example database using an adapted approximate nearest neighbors search. The resulting examples are them combined to form the correct output response of the pedestrian. Those approaches successfully reproduce local avoidance schemes but do not reproduce a global crowd behavior.

We argue that it is possible to classify those approaches with a taxonomy similar to simulation models, *i.e.* microscopic and macroscopic. In the first category, which we will refer to as agent-based data driven approaches, the descriptor used to describe the crowd motion is the sum of all pedestrians spatio-temporal trajectories, thus encoding several local specific situations [175, 176, 142]. In the second category, which will be designed as continuum-based data driven approaches, the goal is to estimate an underlying flow which drives the crowd *[155]* . The figure 27 gives an illustration of these differences. A comparison between the features of the agent-based and continuum-based approaches have been reproduced for the sake of clarity in table 28 .

Most of the data-driven approaches use as input a video of a crowd situation since they stand for one of the most convenient and practical way to record a crowd event. Extracting important information for an animation controller constitutes in itself a problem. Those aspects will be discussed in the next chapter, as well as the design of data-driven controllers for crowd animation.

		Model-driven Approaches	Agent-Based approaches	Continuum-based approaches
Acquisition	type of crowds	sparse and dense	sparse	dense
	Data type	model parameters	database of microscopic situations	time-varying field of dense continuous quantities
	Size of data	neglectful	function of number of local situations in the video	function the time-varying field resolution
	nature	identification fully automatic	Tracking, semi-automatic, hand labelled	Dense estimation fully automatic
	processing time	several minutes	several hours/days	several minutes
Restitution	type	model integration	find approximate nearest neighbor situation in the database	control individuals w.r.t. the continuous quantities
	generate new local situations	yes	no	yes
	generate new global situations	yes	yes	no
	Major restriction	limited to the model expressivity	the results depends on the completeness of the database	The control loop can be tricky

Figure 28.: **Differences between the two approaches of data driven crowd animation**.

6

Data description of a crowd: acquisition and analysis

Contents

In this chapter, we consider the problem of estimating relevant parameters for crowd description from crowd videos. In this works, we first considered the velocity field observed in the image as a good descriptor. In order to extract it from crowd motions, we relied on dedicated fluid motion estimators (Section 6.1). Yet, it appeared that this information is insufficient to characterize properly the crowd. In a second time, we tried to estimate the density from the videos. Contrary to the velocity information, the density can not directly be observed in the image, so a different strategy need to be devised. We chose to rely on optimal control techniques that were able to use some assumptions on the crowd's dynamics to infer the hidden density parameter (Section 6.2). Finally, those questions lead us to formulate the problem of the technical validation of crowd analysis methods, which usually requires ground truth data that are particularly hard to obtain with crowd videos. To this end, we designed a synthetic dataset (Section 6.3) that allows to conduct several tests on crowd analysis methods.

6.1. Crowd motion estimation and post-processing

In this part, our goal is to estimate the crowd's velocity from a pair of images. Among the panel of existing approaches for estimating the apparent velocity field from a pair of images, the *optical-flow* techniques are known to be the most efficient ones [177]. Roughly, such techniques aim at minimizing an energy function composed of two terms: the *observation* and the *regularization* (or *smoothing*). The observation part is most of the time issued from the *optical flow constraint equation* (OFCE) and assumes that a given point keeps its intensity in the course of a displacement. Recovering the two components (u, v) of the velocity from this single relation leads to an ill-posed problem, especially in homogeneous areas. In such situations, an infinity of solutions are indeed

possible. This is the well-known "aperture problem" and it is common to manage it by using an additional smoothness constraint that penalizes the spatial variations of the velocity field. The energy function to minimize reads then:

$$\mathcal{H}(I,\boldsymbol{v}) = \iint_{\Omega} \Big\{ \underbrace{\left[\frac{\partial I(\boldsymbol{x},t)}{\partial t} + \nabla I(\boldsymbol{x},t)\cdot \boldsymbol{v}(\boldsymbol{x},t)\right]^2}_{\text{Observation : } dI/dt \approx 0} + \alpha \underbrace{\left[|\boldsymbol{\nabla} u(\boldsymbol{x},t)|^2 + |\boldsymbol{\nabla} v(\boldsymbol{x},t)|^2\right]}_{\text{Smoothing}} \Big\} d\boldsymbol{x} \tag{6.1}$$

where $\boldsymbol{v}(\boldsymbol{x},t) = (u,v)^T$ is the unknown velocity field at time t and location $\boldsymbol{x} = (x,y)$ in the image plane Ω, $I(\boldsymbol{x},t)$ is the image brightness and α a smoothing parameter to define. It can be shown, using the Euler-Lagrange conditions of optimality, that the standard smoothing term of the relation (6.1) is equivalent to promote solutions with a very low component of divergence and vorticity. This is not appropriate for crowd estimation since the apparent velocity field normally exhibits compact areas with high values of vorticity (when pedestrian get round an obstacle) and/or divergence (concentration at a given point). We then prefer to rely to a second-order div-curl regularization that preserves these quantities:

$$\mathcal{H}_{reg}(\boldsymbol{v}) = \iint_{\Omega} \Big(|\nabla \mathrm{div} \boldsymbol{v}(\boldsymbol{x})|^2 + |\nabla \mathrm{curl} \boldsymbol{v}(\boldsymbol{x})|^2 \Big) d\boldsymbol{x}. \tag{6.2}$$

Interested readers may referee to [155] to get precise descriptions on the optimization strategy and on associated numerical implementation issues.
Under the assumption that a very dense crowd behaves like fluids, a time series of dense displacements can then be obtained by minimizing the usual optical flow constraint equation (first term of relation (6.1)) associated with the div-curl smoothing in (6.2). In practice, as the area filled by a pedestrian in an image is more related to a *block* than a pixel, the information contained by all the instantaneous motion fields is redundant. Moreover, the pedestrian-free areas disturb the quality of the motion fields, even with the effect of the smoothing term. In such situations, from an image to an other, there is no warranty that the flow is consistent from a temporal point of view. This results in noisy time series. According to these remarks, it is then of primary interest to *i)* reduce this huge amount of data and *ii)* de-noise the time series. This is done in the Fourier space. The highest frequencies are removed to restore coherent motion fields. In our applications, only the 10% most energetic harmonics were conserved since we observed that they contain more than 90% of the information.

6.2. Density estimation with optimal control

6.2.1. Overview of the method

The coupling of crowd dynamics and real data exhibits very promising results and has opened a rich area of research. One of our paper *[178]* is a contribution in this direction. We argue that the apparent motion information is intrinsically insufficient to characterize the dynamics of the flow since the lack of motion in the image can be interpreted as a null density or a large congestion area where people are likely to be injured. We define a substantially complete crowd flow analysis as the extraction from the sequence of *i)* time-consistent motion fields and *ii)* an associated disturbance potential. The motion field is a rich dynamical descriptor of the flow which can be related to the velocity of flow. The disturbance potential accounts for several physical quantities such as the density or the pressure in the flow. This information is crucial to extract sensible and potentially dangerous areas. Although an important number of approaches, as the one presented above, are available to measure the apparent velocity field from images sequences in various situations, the estimation of the disturbance potential is a critical problem and is still an open domain of research. This component is indeed tricky to observe directly from images. It is nevertheless intuitive that

this potential influences the motion field: in a natural way, human beings tend to avoid over-concentrated or high-pressure areas, and their velocities are directly influenced by the surrounding person concentration.

We use recipes from optimal control theory [179] and variational assimilation [180], originally used in the context of meteorology, to define a new tool for the characterization of the crowd flow. Such techniques enable to estimate a (potentially high dimensional) system state driven with a dynamic model known up to some noise. A key advantage relies on the ability to measure unobserved parameters that control the dynamic model. As such, it is thoroughly adapted to the problem we are dealing with. The definition of a system based on variational assimilation especially requires *i)* a dynamic model related to the motion field and the disturbance and *ii)* an observation operator that links our data (images) to some components of system state (motion fields).

An overall schema is given in Figure 29. We take as input the original images and two user-defined information: the eventual position of obstacles and some predefined destination areas in the image. These two information are combined to compute a potential function that conveys information on the optimal directions of displacements for the crowd. From the input images are also derived some initializations for our algorithm as well as the observations (that mainly consist in the apparent motion between image pairs). These are used in the assimilation process, that tries to match, through an iterative process, the observations and the evolution of the dynamical process. As a result, a complete sequence of velocity and disturbance potential are computed.

We present some background on variational assimilation in the next section (6.2.2), while our model, along with implementation issues, is thoroughly described in the ending part of this section (6.2.3).

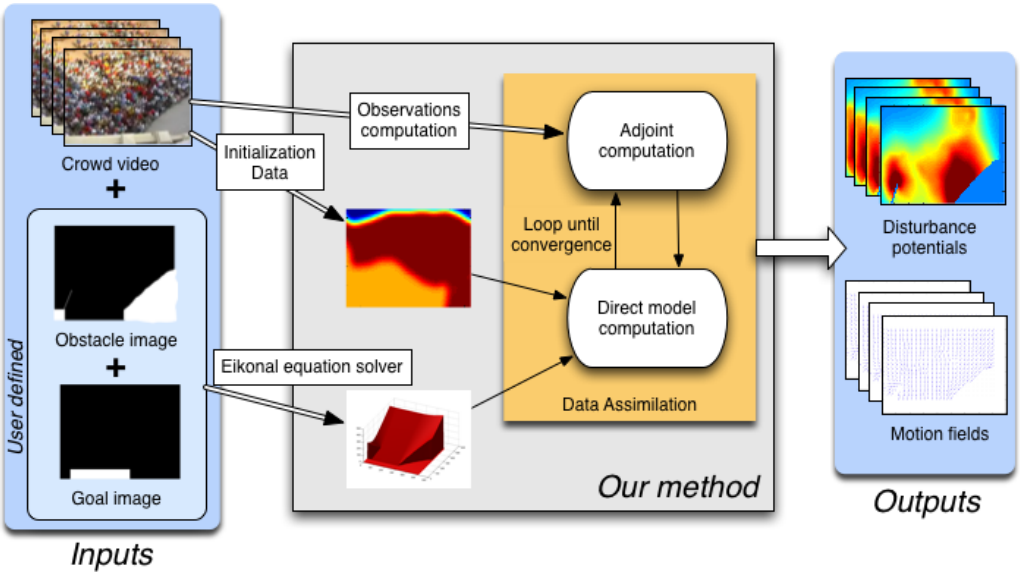


Figure 29.: **Method overview**.

6.2.2. Variational assimilation

Assuming that the crowd is coarsely driven by the dynamical model $\mathbb{M}$, an unknown additive *noise* variable $\boldsymbol{\epsilon}_{\mathbb{M}}$, relative to the deviation over the dynamics, is introduced in the definition of the model, which now writes:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \boldsymbol{\epsilon}_{\mathbb{M}}. \tag{6.3}$$

The initial condition may be given with uncertainties. In a similar way, the initial state value

may therefore depend on a *noise* term $\boldsymbol{\epsilon}_0$:

$$\mathbf{X}(t_0) = \mathbf{X}_0 + \boldsymbol{\epsilon}_0. \tag{6.4}$$

In our case, we can not directly observe the system state $\mathbf{X}$ in the image. Instead we are able to measure some *observations* $\mathbf{Y}$ in time. They are related to the system state by a (possibly non-linear) function $\mathbb{H}$, which is also known up to a given noise $\boldsymbol{\epsilon}_{\mathbb{H}}$. This yields the following system:

$$\mathbb{H}(\mathbf{X}(t)) = \mathbf{Y} + \boldsymbol{\epsilon}_{\mathbb{H}}. \tag{6.5}$$

Hence our estimation problem consists in finding the best system state $\mathbf{X}$ that satisfies the relations (7.15,7.16,7.17). This must be accomplished while minimizing the discrepancy between the dynamic model, the observation operator and the respect to the initial condition. This formalizes as the minimization of the following cost function $\mathcal{J}(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$:

$$\mathcal{J} = \frac{1}{2}\int_{t_0}^{t_f} \|\mathbf{Y} - \mathbb{H}(\mathbf{X})\|^2_{C_{\mathbb{H}}^{-1}} dt + \sum_{s=\{\mathbb{M},0\}} \frac{1}{2}\int_{t_0}^{t_f} \|\boldsymbol{\epsilon}_s\|^2_{C_s^{-1}} dt. \tag{6.6}$$

The norm $\|.\|_{C_s^{-1}}$ is the induced norm of the inner product $< C_s^{-1}., . >$ and C_s is an endomorphism defining the covariance matrix of parameters $\boldsymbol{\epsilon}_s$. In the rest of the document we denote $Q = C_{\mathbb{M}}$, $R = C_{\mathbb{H}}$ and $B = C_0$. These covariances are crucial for the assimilation of observations. Once the model and the observations are set, they remain the only parameters to be configured by the user. For example, setting an unchangeable crowd configuration at t_0 amounts to assign B^{-1} an infinite value $+\infty$.

Resolution of the system In order to estimate the system state $\mathbf{X}$ a common methodology consists canceling the gradient of this cost function. Unfortunately, the estimation of such gradient is in practice unfeasible for large system's state since it requires to compute perturbations along all the components of $\mathbf{X}$. A way to cope this difficulty, firstly proposed by Lions in [179], is to write *an adjoint formulation* of the problem. It can be shown that this yields the following algorithm:

1. Starting from $\tilde{\mathbf{X}}(t_0) = \mathbf{X}_0$, perform a *forward* integration: $\frac{\partial \tilde{\mathbf{X}}}{\partial t} + \mathbb{M}(\tilde{\mathbf{X}}) = 0$
2. $\tilde{\mathbf{X}}$ being available, **find the adjoint variables** $\boldsymbol{\lambda}(t)$ with the *backward* equation:

$$\frac{\partial \boldsymbol{\lambda}}{\partial t} + (\partial_{\mathbf{X}}\mathbb{M})^* \boldsymbol{\lambda} = (\partial_{\mathbf{X}}\mathbb{H})^* R^{-1} (\mathbf{Y} - \mathbb{H}(\mathbf{X})) \tag{6.7}$$

3. **Update the initial condition** : $d\mathbf{X}(t_0) = B\boldsymbol{\lambda}(t_0) + d\mathbf{X}(t_0)$;
4. $\boldsymbol{\lambda}$ being available, **find the state space** $d\mathbf{X}(t)$ from $d\mathbf{X}(t_0)$ with the *forward* integration

$$\frac{\partial d\mathbf{X}}{\partial t}(t) + (\partial_{\mathbf{X}}\mathbb{M})\, d\mathbf{X}(t) = Q\boldsymbol{\lambda}(t) \tag{6.8}$$

5. **Update** : $\tilde{\mathbf{X}} = \tilde{\mathbf{X}} + d\mathbf{X}$
6. **Loop** to step 2 until convergence

where $\partial_{\mathbf{X}}\mathbb{M}$ is the linear tangent simulation model of $\mathbb{M}$ and $(\partial_{\mathbf{X}}\mathbb{M})^*$ its adjoint [1]. In a similar way, we define the linear tangent observation model of $\mathbb{H}$ and deduce its adjoint $(\partial_{\mathbf{X}}\mathbb{H})^*$. Intuitively, the adjoints variables $\boldsymbol{\lambda}$ contain information about the discrepancy between the observations and the dynamic model. They are computed from a current solution $\tilde{\mathbf{X}}$ with the backward integration (6.7) that implicates both observations and dynamical operators. This deviation information between data/model is then used to refine the initial condition (step 3) and to recover the

[1] *i.e.* $< (\partial_{\mathbf{X}}\mathbb{M})\, x, y >=< x, (\partial_{\mathbf{X}}\mathbb{M})^* y >$

system state through an imperfect dynamic model where errors are $Q\boldsymbol{\lambda}$ (step 4). Note that if the dynamic is supposed to be perfect (like in many physical applications), the associated covariance Q is null and the algorithm only refines the initial condition.From the previous algorithm, a complete assimilation system is then defined with *i)* a dynamic model $\mathbb{M}$; *ii)* an observation operator $\mathbb{H}$; *iii)* an initial condition and *iv)* the covariance matrixes B, Q and R. The next section defines all these components for our problem.

6.2.3. Dynamic model, observations and covariance

Proposed dynamic model for crowd behavior The aim of this part is to design a simple dynamic model for crowds that will be used for the assimilation. The system's state $\mathbf{X}$ is composed of the two components of interest that are the velocity field $\boldsymbol{v} = (u, v)^T$ and of the disturbance potential of the crowd D ($\mathbf{X} = (u, v, D)^T = (\boldsymbol{v}, D)^T$). Let us define a model for the velocity evolution.

Velocity modeling In order to get a prior knowledge of the displacement of the crowd, we assume that all human share the same goal and that the topology (obstacles) of the analyzed scene is available. In a first place our methodology is thus restricted to image sequences exhibiting one main flow of pedestrians. Reasonably assuming that each pedestrian aims at minimizing their travel time to their objectives, the optimal direction at a given location can be modeled as the gradient of a potential function Φ defined over the whole domain D. This potential is the solution of the classical Eikonal equation which has among others been widely used in the context of path planing [181]. For a given scene, we then derive an optimal field $\boldsymbol{V} = (U, V)^T = \nabla\Phi$ of the pedestrians that corresponds to the theoretical normalized direction of a pedestrian without any constraint. If now the pedestrians evolve in a crowded environment, we assume that if their velocity differs from the optimal direction, this is due to a disturbance into the scene (density, pressure, ...). Therefore, we propose the following dynamical model:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \alpha\Big(\boldsymbol{V}(\boldsymbol{x}, t) \quad \underbrace{-\beta\nabla D(\boldsymbol{x}, t)}_{\text{disturbance repulsion}} \quad\Big) \tag{6.9}$$

where α and β are two constant coefficients that depend on the global speed of the scene.

Disturbance potential modeling As for the disturbance potential modeling, we simply assume that this scalar quantity is transported by the motion field and is also eventually diffused along time. This corresponds to a simple physical equation of transport of a scalar. It then obeys to a classical advection-diffusion relation:

$$\frac{\partial D(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla D(\boldsymbol{x}, t) = \delta\Delta D(\boldsymbol{x}, t). \tag{6.10}$$

where δ is a small diffusing parameter. Finally, the complete dynamical system of $\mathbf{X} = (\boldsymbol{v}, D)^T$ reads (with $(\bullet) = (\boldsymbol{x}, t)$):

$$\begin{bmatrix} \boldsymbol{v}(\bullet) \\ \frac{\partial D(\bullet)}{\partial t} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & \alpha\beta\nabla \\ 0 & \boldsymbol{v}(\bullet)\cdot\nabla - \beta\Delta \end{bmatrix}}_{\mathbb{M}(\mathbf{X})} \begin{bmatrix} \boldsymbol{v}(\bullet) \\ D(\bullet) \end{bmatrix} = \begin{bmatrix} \alpha\boldsymbol{V}(\bullet) \\ 0 \end{bmatrix} + \boldsymbol{\epsilon}_{\mathbf{m}} \tag{6.11}$$

To suppress the obstacle influence in the computation of the gradient ∇, we used non-symmetric finite-difference in their neighborhood. Concerning the the Laplacian operator Δ related to the diffusion in (6.10), we applied an anisotropic operator that do not diffuse into the obstacles.This dynamic model $\mathbb{M}$ is non-linear due to the advection term $\boldsymbol{v}(\bullet)\cdot\nabla$ that depends on the density. In practice, at a given iteration n, the velocity $\boldsymbol{v}$ used for the advection is the one obtained at iteration $n-1$ so that the operator is linear. The associated tangent linear $\left(\frac{\partial\mathbb{M}}{\partial\boldsymbol{\mathcal{X}}}\right)$ is then itself.

The analytical expression of the adjoint $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)^{\dagger}$ is more tricky to obtain but in our implementation, we use the fact that its discrete version is the transpose of the discrete version of $\left(\frac{\partial \mathbb{M}}{\partial \boldsymbol{\mathcal{X}}}\right)$ [182].

Let us now turn to the observations of the state variables.

Observations: velocity based on optical-flow As mentioned above, only the motion fields $\boldsymbol{v}$ can be accurately observed from the images, the disturbance potential being a tedious quantity to estimate. Starting from the well-known optical flow constraint equation (ofce), one can assume, to cope with the aperture problem, that the unknown optic flow vector at a location $\boldsymbol{x}$ is constant within some neighborhood of size n [183]. The motion field respects then:

$$K_n * \underbrace{\left(\frac{\partial I(\boldsymbol{x},t)}{\partial t} + \nabla I(\boldsymbol{x},t)\cdot \boldsymbol{v}(\boldsymbol{x},t)\right)}_{dI/dt} \approx 0, \tag{6.12}$$

where I stands for the luminance function and K_n is a Gaussian kernel of standard deviation n. From the previous relation, the observation system $\mathbf{Y}(\boldsymbol{x},t) = \mathbb{H}(\boldsymbol{x},t)\mathbf{X}(\boldsymbol{x},t) + \boldsymbol{\epsilon}_o$ can be defined with (noting $I_\bullet = \partial I/\partial \bullet$):

$$\mathbf{Y}(\boldsymbol{x},t) = K_n * I_t(\boldsymbol{x},t) \;\text{ and }\; \mathbb{H}(\boldsymbol{x},t) = \left[\; -K_n * I_x(\boldsymbol{x},t), \quad -K_n * I_y(\boldsymbol{x},t), \quad 0 \;\right]. \tag{6.13}$$

This observation operator involves only the motion field. This means that the correction on the disturbance potential will uniquely be achieved by relying on motion observations. From a computational point of view, this operator is linear. The associated tangent linear and adjoints are then derived in the same way than previously.

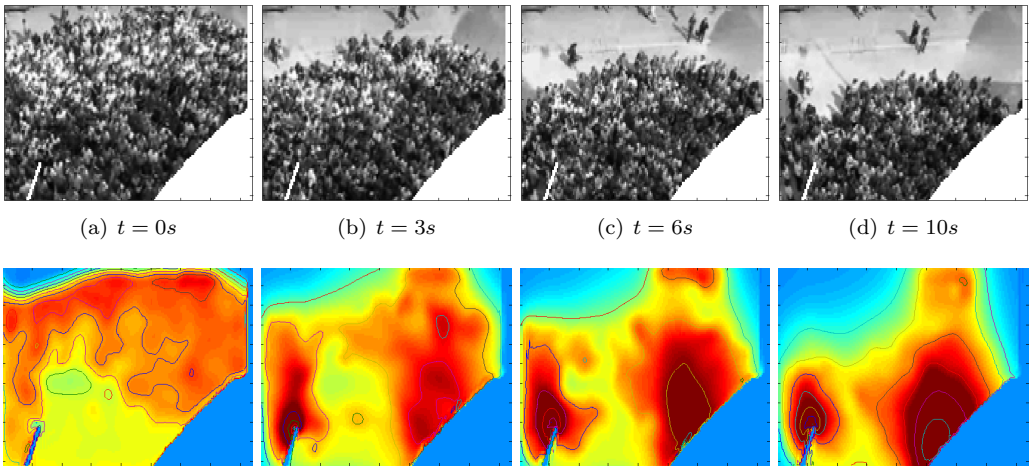
Covariances and initialisations For the initialization, we only need to get the disturbance potential since the corresponding initial velocity field is obtain from (6.9). The choice of this density depends on the scene to be analyzed. In our experiments, it was roughly set manually and filtered with a Gaussian kernel. Noting that the assimilation process refines this initialization, this latter can be only issued from a coarse and manual estimation.

The covariance matrix of the initial condition B and the covariance matrix of the dynamic model parameter Q have been fixed to constant diagonal matrices (no spatial prior on the validity of the model and the initial density are available). Concerning the observation covariance R, we used $R = R_{max} + (R_{min} - R_{max})(1 - \exp(-\|\nabla I\|/\sigma^2))$. This states that when the image brightness does not contain gradients, the usual ofce is not valid and the covariance is maximal. At the opposite, when high gradients appears, the ofce is confident and R is low.

6.2.4. Example on real crowd

We illustrate here the behavior of our algorithm on a real sequence showing a crowd entering a railway station in the Principality of Monaco (Figure 30). This example is interesting since a variety of phenomena are present: a continuous flow at the beginning followed by a compression of some peoples in the left part of the images. In addition, the limit of the door is a barrier that creates an opposite flux in the crowd flow. In this example, our method has detected two sensible areas where the disturbance potential is growing larger : the end of the barrier and the wall on the right of the image. This is very informative for safety engineers, since it allows to highlight potential risky zones. From an online surveillance system point of view, our method can detect critical disturbance elevations and thus would allow to trigger alarms. It is also possible to connect this information in some motion pattern detector such as presented in [184]. Those aspects have been left as perspectives. Let us remark here that the problem of validation is difficult since no ground truth is available. Nevertheless, from the state-of-the-art on crowd behavior, our estimations seem coherent.

Finally, since a quantitative evaluation is mandatory to understand the qualities and defects of our approach, we proposed the definition of a synthetic dataset that was made avilable to the



(a) $t = 0s$ (b) $t = 3s$ (c) $t = 6s$ (d) $t = 10s$

Figure 30.: (a) to (d) Images of the real sequence - (e) to (h) Estimated disturbance potential maps

community for the purpose of evaluating crowd analysis algorithm. We briefly present this dataset in the next Section.

6.3. Agoraset: a synthetic dataset for crowd analysis

As explained by Zhan and colleagues [185], conventional computer vision methods used in the context of tracking fail to analyze crowded situations. Principal reasons arise from several factors. Despite the inherent computational complexity of handling several individuals, the related pixel information is rather poor, and undergo multiple occlusions over time. Moreover, modeling the physical nature of a crowd is a strongly non-trivial task, as it implies inter-indivudal and environments interactions, with pedestrians exhibiting different goals or implied in social interactions.

Both the large applications field and the challenging vision problems have led to the development of an important number of new vision algorithms over the past decades. The related methods either seek to count or track individuals or to detect changes in the crowd flow or abnormal patterns. At the moment, the community is lacking of a common test bed and reference situations (with eventually associated ground truths). Though, it has been already shown in other context such as, for example, object recognition [186], optical flow [187] or articulated motion estimation [188] that datasets greatly stimulates the research field and allows for direct, objective comparisons between state-of-the-art algorithms. Our contribution is in that direction and aims at providing a variety of crowd situations along with their associated ground truths. To account for the different possible representations of crowd phenomena, these ground truths contain: the individual trajectories of each pedestrians in the crowd and the related continuous quantities such as density and dense velocity field.

No real crowd videos were included in the data set. The main reason is that obtaining ground truths for those videos is a very time-consuming task as it requires to label by hand the positions of each individuals in the scene. Moreover, experiments show that manual labeling is prone to errors and can differ between two persons. We instead rely on realistic image synthesis to achieve our goal. This idea is not new and has already been exploited in the context of surveillance of human activity [189, 190, 191]. Still, our data set constitutes the first of this kind devoted to crowd phenomena. Obtaining realistic synthetic crowd videos is in itself a challenge. Regarding crowd video analysis requirements, two major problems are to be taken into account: the visual quality of the images should reflect the diversity that can be observed in real footages and the dynamics of the crowd should be preserved. This last point include both details of single pedestrians gaits and

motions, as well as the overall continuum dynamics. In our work the rendering has been performed thanks to a commercially available renderer and a classical and well established simulation model is used.

6.3.1. Presentation of the dataset

In this first version of the proposed dataset, we have identified seven typical scenes where some crowd behaviors appears. They are schematized in the figure 31. Each scenario topology (also named "environment" or "scene"), was designed in accordance to situations often met in crowding issues. They correspond to an evolution on a flow of humans in a free environment (scene #1), in an environment with obstacles (#2 and #3), an evacuation through a door (#4), a dispersion (#5), a rotation (#6) (with an analogy of the famous crowd scene of the pilgrimage in Mekkah) or some crossing flows (#7) (this last case being related to the "unstructured crowd" of [167]).

From the scenes we have depicted in figure 31, several scenarios have been generated. For each environment, two different sequences that correspond to various values of desired velocity (soft and panic) have been generated. In each case, the pedestrian positions are randomly set in a starting area and are not submitted to any motion during $3\,s$. After this delay, the wished direction is included in the simulation model. Concerning the rendering, several videos are also available for a single event. They correspond to various camera parameters and lighting conditions. For now we propose for analysis two camera views : perfect sky, and sided view. And two lighting conditions : shading, and no shading. Thus allowing a variety of rendering realism for a same scenario.

Let us now turn to the production pipeline of the different video of the dataset.

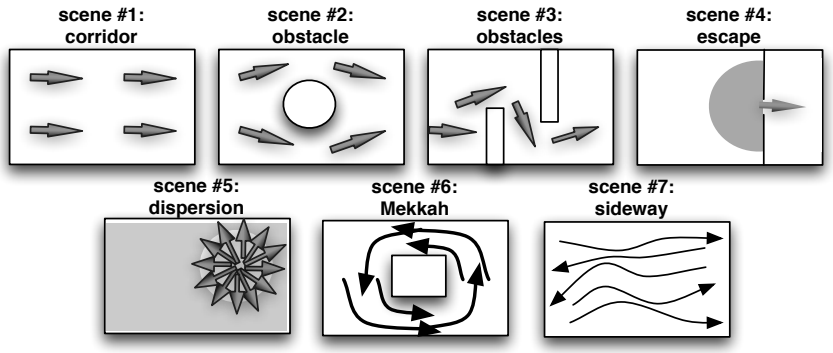


Figure 31.: **Scene typology**. The different scenes proposed in our database

6.3.2. Production pipeline

As a first step the crowd simulation model is presented, followed by a short description of the rendering process.

Crowd simulation model Because of its compacity and efficiency, we chose to use the model proposed by Helbing *et al.* [141] for crowd simulation. It considers pedestrians as Lagrangian particles carrying individual properties affecting their dynamic through different crowd-related forces. This model has been widely used for different purpose involving crowd dynamics as in [192, 193], or revisited as in [194].

As the latter authors, we also chose to slightly revisit the model in order to reach as much as possible visual realism. The original purpose of Helbing *et al.* model is indeed to match different macro data of the crowd like global evacuation time. The model's parameters are thus

not specifically calibrated to handle visual realism. The particles are too solid for having the visual impression of people being pressed in case of congestion. In this purpose we modify several parameters which are given in Table 1.

		Pedestrians interaction	Obstacles interaction	*Units*
Contact	k	6×10^3	6×10^3	$kg.s^{-2}$
	κ	12×10^3	12×10^3	$kg.m^{-1}.s^{-1}$
Avoidance	a	600	400	N
	b	0.3	0.3	m

Table 1.: **Modified parameters of the crowd model proposed by Helbing *et al.*** These values lead to more flexible behaviors of individuals.

The sequences to be simulated also need to provide enough variability to show realistic crowd behavior. In this purpose, the dynamic parameters are randomly picked for each pedestrian: in addition to the radius r being set in the interval $[0.2\,m, 0.3\,m]$, the mass m is taken as proportional to the latter with an average of $80\,kg$. The response time of pedestrians is in the interval $0.5\,s \pm 10\%$ while the desired nominal velocity is in $1.2\,m.s^{-1} \pm 10\%$. With these parameters, we assume the crowd is populated by different kinds of persons having little different intentions and capabilities, as in a real crowd flow. As in [195], pedestrians handle differently the social forces wether neighbors are in sight of pedestrians or not. We therefore add it a perception coefficient $\alpha = 1 - \left(\frac{\text{acos}(\mathbf{e}_{in}\mathbf{W}_i)}{\pi}\right)^{\gamma}$, with $\mathbf{e}_{in}$ being the unit vector from pedestrian i to the neighbor n, and $\mathbf{W}_i$ the desired direction to take. This coefficient also accounts for the anisotropic dynamic behavior of the crowd that we parameterize with $\gamma = 0.7$. In order to provide visual realism in panic situations, we also add a stumble term to the dynamic velocity equation, reading: $-\psi \mathbf{v}_i \mathbf{v}_i^2 g(\mu(r_i + r_n) - d_{in})$, where $\mathbf{v}_i$ is the velocity, d_{in} the distance between the jostling pedestrians, and $g(x)$ a function being zero if $x > 0$ and x otherwise. We set $\psi = 300\,kg.s.m^{-3}$ and $\mu = 1.2$. The pedestrians are also supposed to be reluctant to go backward from their goal. Their speed limitation when going this (wrong) way is then augmented by adding to the dynamic velocity equation the following term: $\nu m_i \frac{\mathbf{v}_i}{\tau_i} g(\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}\mathbf{W}_i)$, with τ_i being the characteristic reaction time and ν a reluctance coefficient set to 2.

Rendering process The output of the simulation model are then used as input for python scripts that automatically generates a 3D scene with human characters. A set of 26 characters were used (see Figure 32, left) in order to guarantee a sufficient local variability in shapes and colors. This number was set experimentally, but it turned out that a lower number of individuals raised notably the probability of having two or more of the same kind of geometric models near each other, with a possibility of disturbing the analysis. A short number of walking motions and idling gaits have been used for each individuals. At runtime, the best motion is chosen with respect to the pedestrian velocity. Here again, the inter-individual diversity is assured by different playback speeds, which prevents from several individuals having the same motion (also known as the clone effects).

For the rendering process we chose the Mental Ray renderer [196]. The mental ray physical sky model was used; it allows to have a natural and intuitive control over the illumination parameters. Most of the scenes were rendered with outdoor lighting conditions. A comparison between a real image extracted from a real video sequence and a rendering of a crowd scene with similar lighting conditions is shown in Figure 32. It illustrates the ability of our rendering pipeline to produce images that qualitatively looks like real ones.

We present in figures 33 some screenshots that rely on simulations #4-1. As for this environment, pedestrians aim at reaching the right part of the scene by crossing a door, under a "normal" pace (fig. 33). The first line of these figures correspond to a streak representation of the pedestrian trajectories whereas on the second line is depicted the rendered scene under a given camera position.

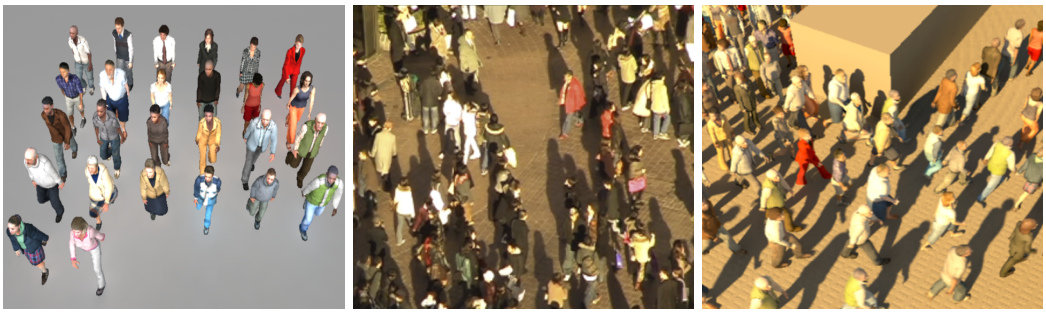


Figure 32.: **Crowd Rendering**. From left to right: the 26 different avatars used to produce the videos (their choice has been made to exhibit the maximum variability w.r.t. age, sex and cloths style); a real image from a video footage of Shibuya in Tokyo; a crowd rendering with similar day light conditions

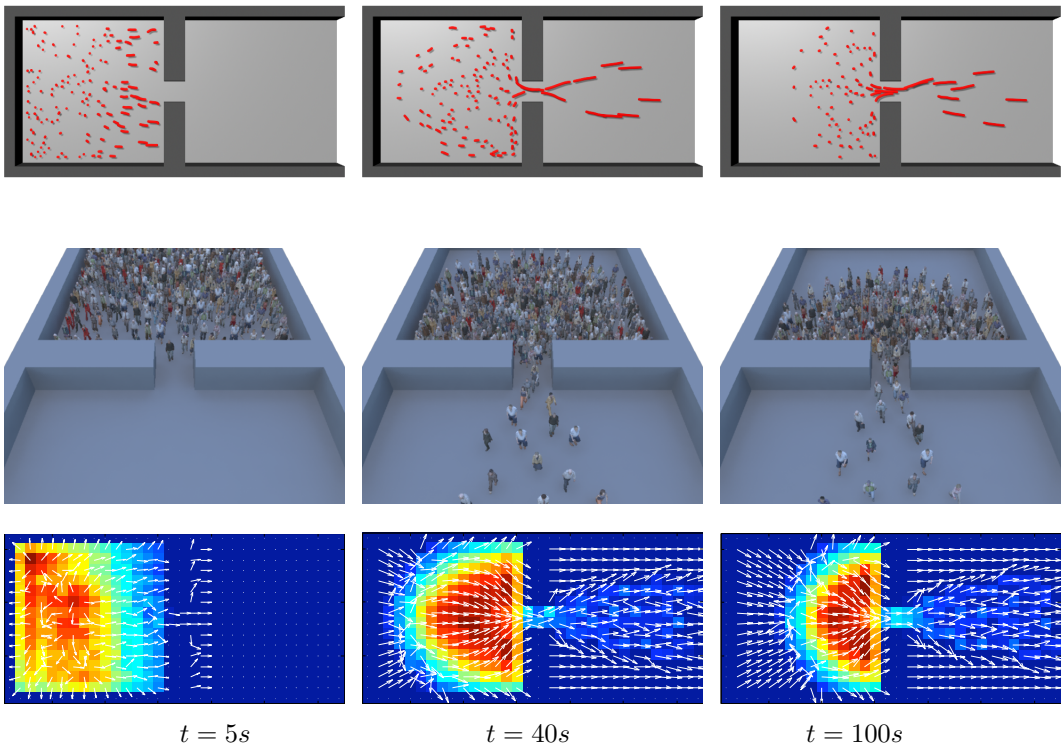


Figure 33.: Scenario #4-1 : Evacuation in a normal situation. First strip, Trajectories of one individual out of 3 for a 3 seconds duration. Second strip, screenshots of the corresponding video. Third strip, the related continuous parameters (density and velocity field).

From these figures, it is very interesting to observe that the rendering is very realistic. Moreover, the emerging phenomenon highlighted in [141] that correspond to an “arching” or ”clogging” effect near the exit appears clearly. This suggests that, in addition to a realistic rendering of the simulations, the crowd behaviors are consistent.

6.4. Summary

In this Chapter the problem of estimating quantities inherent to crowd flows was exposed. Our contributions lies in:

- using estimators from fluid flows analysis to estimate time series of dense velocity fields. Those estimators were chosen so as to preserve interesting characteristics of the crowd flows,
- developing a new density estimator based on some assumptions made over the crowd dynamics. The coupling with this model and the observed data in the image was performed thanks to the theory of variational assimilation.

Also, in order to test our methods, a database of synthetic situations was developed and proposed to the community.
In the next Section, the problem of incorporating those data in a crowd simulation will be considered.

7

Control methods for crowd simulation

Contents

In this Chapter, we examine how the data that was acquired by the different vision algorithms presented in the previous Chapter can be used in a synthesis loop. We recall here that most of the analysis was conducted so as to obtain some continuous information of the crowd motions. Section 7.1 will show that a simple control policy based on time series of velocity fields can give interesting results, but may lead to inconsistent trajectories because inter-pedestrians interactions are not properly taken into account. We thus present in Section 7.2 a new control strategy which helps in mixing these data (expressed as constraints) with an existing simulation model.

7.1. A simple data-driven animation of crowds

If one consider only the simple time series of motion fields computed thanks to algorithm described in the previous chapter, it is possible to consider this information as input data for a simple data-driven crowd animation system *[197]* . Given the position of a person in the virtual world, it is possible to get the corresponding position in the image frame along with a camera projection model. Parameters for this projection can be obtained exactly through camera calibration. We have considered as an approximation of this model a simple orthographic projection in the experiments presented in the result sections. This assumption holds whenever the camera is sufficiently far away from the scene. Once this projection has been defined, animating pedestrians which constitute the crowd amounts to solve the classical following differential equation (with $x(t)$ the position of a person in the image frame at time t) :

$$\frac{\partial \boldsymbol{x}}{\partial t} = \mathbf{v}(\boldsymbol{x}(t), t) \tag{7.1}$$

equipped with appropriate initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ which stands for the initial positions of the individual in the flow field. In our framework we have used the classical 4-th order Runge Kutta integration scheme, which allows to compute a new position $\boldsymbol{x}(t+1)$ given a fixed time step with an acceptable accuracy. This new position is then projected back in the virtual world frame.

7.1.1. Some results on real video sequences

We present the results obtained on two real sequences. Both data have been acquired with a simple video camera with an MPEG encoder. The motion estimation approach was applied without any particular process.

Strike sequence The first real sequence is a video representing a strike. All pedestrians are walking in the same direction. Two images of the sequence can be seen on Figure 34 (a-b). In Figure 34 (c-d), we present the synthetic crowd animation obtained superimposed on the estimated motion field. One can observe that the resulting crowd animation respect the initial yet simple pedestrian behaviors. The real scene can then be reproduced with accuracy while maintaining, despite the regularity of the flow, a particular diversity in the pedestrian trajectories.

Entrance sequence The second real sequence shows a crowd entering a stadium. This example is very worthwhile since a variety of phenomena are present: a continuous flow at the beginning followed by a compression of some peoples in the left part of the images. In addition, the limit of the door is an obstacle that creates two opposite fluxes and that generates a vortex in the motion fields. Four images of the sequence are displayed in Figures 34 (e–h).

The corresponding animations are represented in Figures 34 (i–l). Figures 34 (m–p) are focused on the region that exhibits opposite motions. One can see that this complex behavior has been correctly captured and re-synthesized. This is very stimulative regarding the possibilities of this approach to manage complex flows. The next step of the process will be to extract the different behaviors (compression, rotation for instance) and to synthesize them independently.

7.1.2. Discussion

Our technique has been applied with success to reproduce the observed scenes. Nevertheless, we observed that the quality of the generated animation is directly linked to the initial density of the crowd members (manually set up). In this sense, it is the role of the animator to design an initial crowded situation that is similar to the video conditions. In this sense, the velocity in itself fails to characterize completely the crowd dynamics, and it would be of prime interest to add the density information as a complement to describe the dynamics. It is then extremely important to be able to estimate simultaneously the motion field *and the associated density* to create animations from minimal inputs and manual adjustments.

The density estimation of the pedestrians from an image sequence is however a very difficult problem which has not yet found a solution in the computer vision community. Noting that the density is intrinsically related the the velocity and that this latter quantity can be estimated from images, our contribution consists in estimating the density using optimal control tools. Following a dynamic model, this quantity acts as a control parameter to explain the temporal variations of the velocity field.

Also, this simple analysis/synthesis scheme may not keep people away from colliding between each other. This point could be solved by mixing the *a priori* information acquired by motion estimation as an input parameter of a classical dynamic system. This implies that one has to mix in some ways quantities that can be both related to a continuum (velocity fields, density) and individuals (steering behaviors for collision avoidance). In the next Section, we propose a framework which helps in handling this dual nature of information.

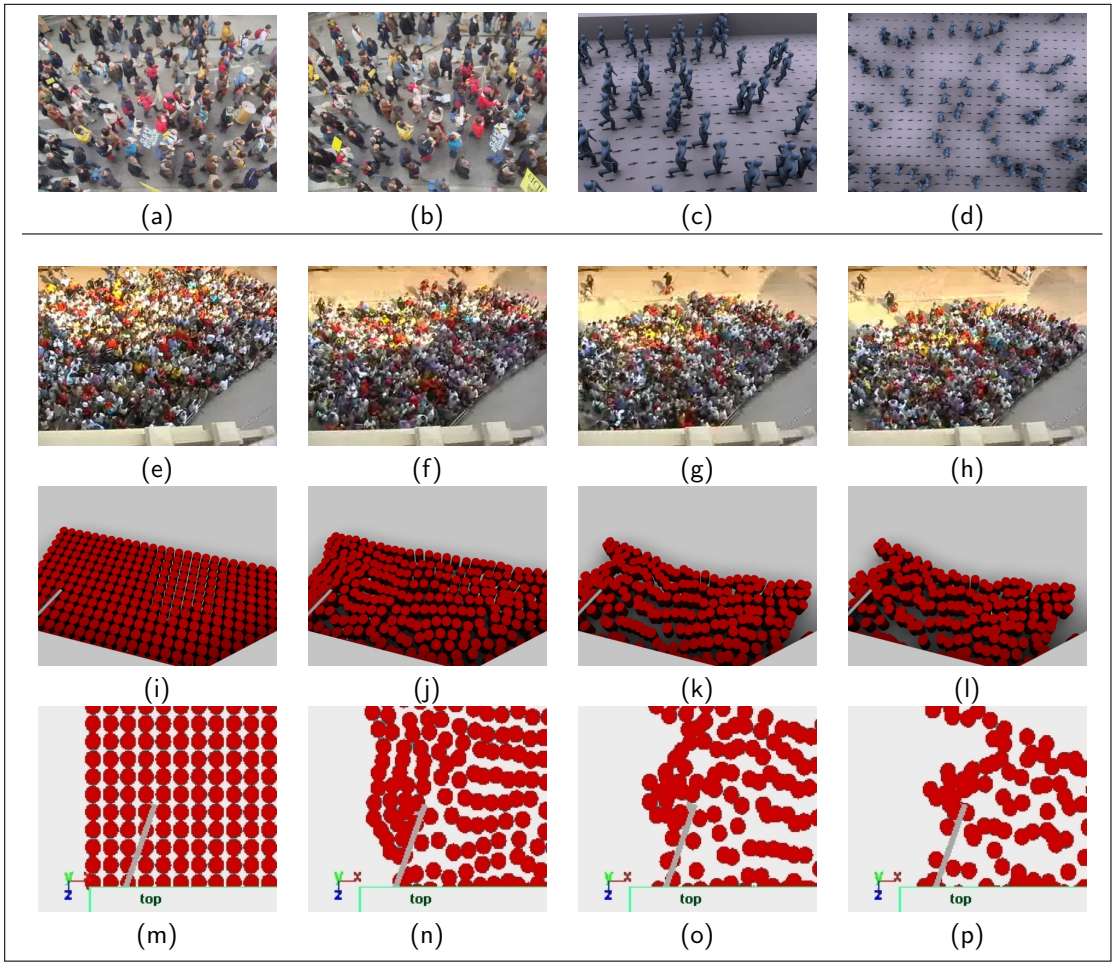


Figure 34.: **Experimental results on real data:** (a,b): Sequence #2: strike video taken from above; (c,d) images of the resulting animation; (e,f,g,h) Sequence #3: video of a crowded entrance; (i,j,k,l) images of the resulting animation; (m,n,o,p) close-up on a remarkable zone where two opposite fluxes of people are juxtaposed

7.2. Crowd Control

In this Section, we present a new crowd control methodology, which can be seen as an editing process, in the sense that the trajectories produced by the simulation model are deformed to achieve users constraints while minimizing the discrepancy with the simulation model's dynamics. The types of constraints can be twofold: *i)* per-individual constraints, meaning that the user can specify its own properties related to pedestrians (like positions, velocities or even shape-related information) or *ii)* macroscopic constraints, such as respecting a given velocity field or higher order dynamical information, like the divergence or rotational components of a velocity field. Our optimization process uses recipes from optimal control of variational models by formulating the problem with the adjoint theory [179]. It is virtually adaptable to any kind of simulation model provided that it can be analytically described as a variational system. This is usually the case with crowd dynamics model, but not anymore if one considers cognitive modeling of pedestrian steering behaviors. Also, the quality of the produced animation strongly depends on both the realism of the controlled simulation model and on the nature of the constraint imposed by the animator. In that sense our method augments the latent qualities of a given simulation model but is not meant to produce systematically more realistic simulations than advanced techniques.

Our crowd control system requires three major ingredients: *i)* a simulation model, which describes how the crowd is moving in a deterministic fashion, *ii)* constraints provided by the animator and *iii)* the control process which combines the two previous information. We note the strong relations with the formulation of the estimation problem of the previous Chapter.

Simulation. We refer to the simulation model as a unified model $\mathbb{M}$ driving the state of pedestrians $\mathbf{X}$, which evolution is governed by the following partial differential equation:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = 0. \tag{7.2}$$

However, a crowd can exhibit different scales of dynamic. One can chose to control only large scales, potentially governed by continuum equations, and leading to express the model in the Eulerian space. But people also interact to each other, at least to avoid collision, involving a Lagrangian expression of the model. The more convenient way to gather these dynamics into a unified model is to express them in the Lagrangian space (i.e. $\mathbf{X}$ is related to pedestrians quantities). This choice also requires to express the large scale model, generally continuous, in the Lagrangian domain. The projection of the continuum dynamic on the Lagrangian space can be done by using differentiable kernel functions (as in [149]), such that common operators (gradient, laplacian, etc.) applied on continuum quantities can be expressed in the Lagrangian model. As well as Lagrangian interactions, this amounts to simply consider weighted relations between pedestrians. These relations can also be used to express Eulerian quantities (i.e. continuous, or related to the environment) with respect to Lagrangian data over an Eulerian grid. A good example is provided by the density, which can be computed as the convolution of a Gaussian kernel and Dirac centered on the pedestrians positions.

We have developed for this purpose a complete graph formalism which description can be found in the following reference *[198]* .

Control. The control aims at automatically defining the best control parameters of the model $\mathbb{M}$ in order to reach a specific configuration of the states $\mathbf{X}(t)$. A "sensor" is used to compare the model output to a reference signal which stands for the desired constraints. This error is amplified by a controller before being operated by the system, giving a corrected output. In our case, the controller corresponds to an *assimilation* process where the sensor is an observation operator $\mathbb{H}$. We note $\mathbf{Y}$ the reference signal.

Unlike lot of engineering control applications, the control loop is not expressed in the frequency domain, and $\mathbf{X}$ can not be determined by one-pass analytical means. The reason is the specificity of the system which contains time integration of the model. To deal with this specificity, we use optimal control theory recipes as explained in the next Section.We will start by giving the dynamical crowd simulation model used throughout the experiments.

7.2.1. Simulation Model

The evolution of pedestrians is governed by physiological capabilities, psychological behaviors, group strategies and goal achievement. Assuming that this evolution is also governed by Newton's law of motion, we obtain for each pedestrian p_i the system :

$$\frac{\partial y_i}{\partial t} = u_i, \tag{7.3}$$

$$m_i \frac{\partial u_i}{\partial t} = \mathbb{F}_i, \tag{7.4}$$

with y_i being the position of the pedestrian, u_i its velocity and m_i its mass randomly taken in the range [60 kg, 80 kg]. For clarity's sake, we will use a matrix notation stating, as an example, $\mathbf{y} = [y_i]$ the vector of the whole positions of the pedestrians, and $\underline{\mathbf{y}}$ being the diagonal matrix

builded by the same vector. The overall force applied to pedestrians, $\mathbb{F}$, gathers all constraints applied to them. We propose to use the model proposed by [141] by decomposing this force into four major components:

$$\mathbb{F} = \underbrace{\mathbb{F}_{\text{will}}}_{\text{source}} + \underbrace{\mathbb{F}_{\text{fatigue}}}_{\text{friction}} + \underbrace{\mathbb{F}_{\text{sociological}}}_{\text{interactions}} + \mathbb{F}_{\text{obstacle}} \tag{7.5}$$

In the first time, people want to reach some position with a given amount of determination, and will release power according to this amount. The direction to the goal will be expressed as unit vector $\mathbf{W}$, and the level of determination as $\boldsymbol{\alpha}$ (valued $140\,N$ in experiments), leading to the source force: $\mathbb{F}_{\text{will},i} = \alpha_i W_i$, and in crowd space:

$$\mathbb{F}_{\text{will}} = \underline{\boldsymbol{\alpha}}\mathbf{W}. \tag{7.6}$$

But a pedestrian will be slow down by his physiological capabilities since moving is power consuming. This can be expressed as $\mathbb{F}_{\text{fatigue},i} = -k_i u_i$, and in crowd space:

$$\mathbb{F}_{\text{fatigue}} = -\underline{\mathbf{k}}\mathbf{u}, \tag{7.7}$$

where $\mathbf{k}$ are the the friction coefficients of pedestrians (valued $140\,N$ in experiments).

Sociological interactions Pedestrians repulse each other according to a sociological force $\mathbb{F}_{\text{sociological}}$. This force is directed for every pedestrian i to its neighbor j by the unit vector e_{ij}. The intensity of this force decreases with the distance between the pedestrians i and j using an inverse exponential function. The sociological force therefore reads for every pedestrians i and j:

$$f_{ij}^{\mathbf{y}} = -ae^{-\frac{\|y_j - y_i\| - (r_i + r_j)}{b}} e_{ij}, \tag{7.8}$$

where a and b are two coefficients related respectively to the force intensity and to the cutback distance separating high repulsions from the low ones. In the following experiments we set $a = 1000\,N$ and $b = 0.08\,m$. The quantity r stands for the modeled radius of the pedestrians which values are randomly taken in the range $[0.25\,m, 0.35\,m]$. Introducing the adjacency matrix of pedestrians $\boldsymbol{A}$ weighted by f, it is possible to express the sociological force in the crowd space, as proposed by [157], writing:

$$\mathbb{F}_{\text{sociological}} = \boldsymbol{A}_{f^{\mathbf{y}}}\mathbf{1}. \tag{7.9}$$

The connectivity of $\boldsymbol{A}$ is set such that pedestrians distant of more than $2\,m$ are not connected.

In order to present the control results using a simple crowd model, collision body forces are neglected. Besides, the steady conditions used in this method do not require such forces as compared to panic or rush situations.

Obstacle force The obstacles repulsion is directed oppositely toward the closest wall point $\mathbf{x}_{\text{obs}}$, and its intensity is given by a function of the distance from the *i-th* pedestrian position to this point. Letting $e_{\text{obs},i}$ be the unit vector of the force, we obtain for each pedestrian i the obstacle force:

$$w_{\text{obs},i}^{\mathbf{y}} = -ae^{-\frac{\|\mathbf{x}_{\text{obs}} - y_i\| - r_i}{b}} e_{\text{obs},i}, \tag{7.10}$$

leading to the expression: $\mathbb{F}_{\text{obstacle}} = \mathbf{w}_{\text{obs}}^{\mathbf{y}}$.

We now dispose of a complete differentiable dynamical model stating the evolution of $\mathbf{X} = [\mathbf{y}, \mathbf{u}]^T$, by the model:

$$\mathbb{M}(\mathbf{X}) = \begin{bmatrix} -\mathbf{u} \\ -\underline{\mathbf{m}}^{-1}\mathbb{F}(\mathbf{y}, \mathbf{u}) \end{bmatrix} \tag{7.11}$$

Starting from a given initial configuration of pedestrians $\mathbf{X}_0$, we obtain $\mathbf{X}(t)$ with $t \in [t_0, t_f]$.

Derivation The control process requires the linear tangent of the presented model and its adjoint, which reads:

$$(\partial_{\mathbf{X}}\mathbb{M})^* = \begin{bmatrix} 0 & \partial_{\mathbf{y}}\mathbb{F}^T \\ 1 & \partial_{\mathbf{u}}\mathbb{F}^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -\underline{\mathbf{m}}^{-1} \end{bmatrix}. \tag{7.12}$$

In this purpose, the derivation by the positions $\mathbf{y}$ and the velocities $\mathbf{u}$ reads:

$$\partial_{\mathbf{y}}\mathbb{F}^T = (\underline{\boldsymbol{\alpha}}\underline{J_W})^T + \boldsymbol{L}_{f^{\partial \mathbf{y}}} + (\underline{\mathbf{w}_{\text{obs}}^{\partial \mathbf{y}}})^T, \tag{7.13}$$

$$\partial_{\mathbf{u}}\mathbb{F}^T = -\underline{\mathbf{k}}, \tag{7.14}$$

with J_W being the spatial jacobian matrix of the path W_i and $\boldsymbol{L}$ the Laplacian operator associated to $\boldsymbol{A}$ and symmetric in our case.

7.2.2. Control policy

The crowd phenomenon being by nature complex, the dynamical model $\mathbb{M}$ is usually non-linear and can be seen as an approximation of a real pedestrian behavior. The evolution of $\mathbf{X}(t)$ can be directly inferred by integrating $\mathbb{M}$ over time. The assimilation process is in charge of modifying this solution by adding some constraints (or observations) at given times. As a result, the control is defined as a tradeoff between what can be expected from the model and what is actually given by the observations. In fact, the degree of freedom allowing to modify the evolution of the particles predicted by $\mathbb{M}$ is defined as a degree of confidence in the model and in the observations, which can mathematically be represented with a control variable related to the deviation over the initial dynamics. In the assimilation process, those quantities are related to the covariance matrices. This is explained thoroughly in the next section.

7.2.2.1. Problem statement

Assuming that the crowd is coarsely driven by the dynamical model $\mathbb{M}$, an unknown additive *control* variable $\boldsymbol{\epsilon}_{\mathbb{M}}$, relative to the deviation over the dynamics, is introduced in the definition of the model, which now writes:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \boldsymbol{\epsilon}_{\mathbb{M}}. \tag{7.15}$$

In some applications, the initial condition may also be unsure, meaning it can be modified. In a similar way, the initial state value may therefore depend on a *control* variable $\boldsymbol{\epsilon}_0$:

$$\mathbf{X}(t_0) = \mathbf{X}_0 + \boldsymbol{\epsilon}_0. \tag{7.16}$$

The differences between the simulation and the control lies in the *constraints* imposed by the user: at some specific key times $t_i, i = \{1, ..., C\}$ (C being the number of user constraints), we expect the solution to be closed some user constraints $\mathbf{Y}_i$. They are linked to the system state $\mathbf{X}$ through an observation operator $\mathbb{H}$ yielding the system

$$\mathbb{H}(\mathbf{X}(t)) = \mathbf{Y} + \boldsymbol{\epsilon}_{\mathbb{H}} \tag{7.17}$$

with

$$\begin{aligned} &\mathbf{Y} = \mathbf{Y}_i \text{ and } \boldsymbol{\epsilon}_{\mathbb{H}} = \boldsymbol{\epsilon}_{\mathbb{H}i} \text{ if } t = t_i \ ; \\ &\mathbf{Y} = 0 \text{ and } \boldsymbol{\epsilon}_{\mathbb{H}i} = 0 \text{ otherwise.} \end{aligned} \tag{7.18}$$

The values $\boldsymbol{\epsilon}_{\mathbb{H}i}$ are the errors associated to the constraints $\mathbf{Y}_i$ and depend on the precision expected by the user. As will be shown in the following, they are in practice defined through their covariance matrices. Therefore, the control problem consists in extracting $\mathbf{X}$ that satisfies the relations (7.15,7.16,7.17). This is the *control* issue: how to find a control of lower energy on variable $\boldsymbol{\epsilon}_{\mathbb{M}}$

and $\boldsymbol{\epsilon}_0$ that leads to the lowest discrepancy between the constraints $\mathbf{Y}$ and the state variable $\mathbf{X}$. This discrepancy is measured with the difference $\mathbf{Y} - \mathbb{H}(\mathbf{X})$ and the problem can be expressed as a minimization with respect to the control variables $(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$ of the cost function $\mathcal{J}(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$ defined as:

$$\mathcal{J} = \frac{1}{2}\int_{t_0}^{t_f} \|\mathbf{Y} - \mathbb{H}(\mathbf{X})\|^2_{C_{\mathbb{H}}^{-1}} dt + \sum_{s=\{\mathbb{M},0\}} \frac{1}{2}\int_{t_0}^{t_f} \|\boldsymbol{\epsilon}_s\|^2_{C_s^{-1}} dt. \tag{7.19}$$

The norm $\|.\|_{C_s^{-1}}$ is the induced norm of the inner product $< C_s^{-1}.,. >$ and C_s is an endomorphism defining the covariance matrix of parameters $\boldsymbol{\epsilon}_s$. Just like in the previous Chapter, we denote $Q = C_{\mathbb{M}}$, $R = C_{\mathbb{H}}$ and $B = C_0$. These covariances are crucial for the assimilation of observations. Once the model and the observations are set, they remain the only parameters to be configured by the user. For example, setting an unchangeable crowd configuration at t_0 amounts to assign B^{-1} an infinite value $+\infty$.

7.2.2.2. Resolution of the system

We present here a way to estimate the control variables $(\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_{\mathbb{M}})$ required in equations (7.15,7.16) to generate the controlled states $\mathbf{X}$. Minimizing the cost function in (7.19) requires to cancel its derivatives $\delta\mathcal{J}(\delta\boldsymbol{\epsilon}_0, \delta\boldsymbol{\epsilon}_{\mathbb{M}})$ with respect to the control variables $(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$. In practice, due to the size of the system'state, this estimation of is done using an *adjoint formulation*. The user has first to define the linearisation and the associated adjoint operators of involved models (observation and simulation), as well as the associated error covariances matrices related to the confidence in the simulation model, the initial conditions and the observations. Once done, it can be shown that the minimization in (7.19) can performed through some forward/backward integrations. More details about adjoint techniques can be found in [179] and applications in computer graphics in [199, 200]. To sum up, the following steps are required for the user:

PREREQUISITES : from a given simulation model $\mathbb{M}$, an initial condition $\mathbf{X}_0$ and user constraints $\mathbf{Y} = \mathbb{H}(\mathbf{X})$

1. Define associated error covariance matrices Q, B, R related to the expected precision on the model, initial conditions and the constraints.
2. Derive the linear tangent simulation model $\partial_{\mathbf{X}}\mathbb{M} \in \mathbb{R}^{4N\times 4N}$ of $\mathbb{M}$ and deduce its adjoint $(\partial_{\mathbf{X}}\mathbb{M})^*$ such as $< (\partial_{\mathbf{X}}\mathbb{M})\, x, y >=< x, (\partial_{\mathbf{X}}\mathbb{M})^*\, y >$
3. In a similar way, derive the linear tangent observation model of $\mathbb{H}$ and deduce its adjoint $(\partial_{\mathbf{X}}\mathbb{H})^*$

Then, using the adjoint technique, it can be shown that the global editing process leads to the following incremental method:

EDITING PROCESS 1: BASIC INCREMENTAL TECHNIQUE USING THE ADJOINT METHOD.

Initialization: set an initial iteration $k = 0$

1. Set control variables to zeros
$$\boldsymbol{\epsilon}_{\mathbb{M}}^{k}(t) = 0 \ ; \ \boldsymbol{\epsilon}_{0}^{k} = 0 \ ; \ \mathbf{X}^{k}(t_0) = \mathbf{X}_0$$
2. Obtain a first simulation of $\mathbf{X}^k(t)$ by integrating
$$\mathbf{X}^{k}(t_0) = \mathbf{X}_0 + \boldsymbol{\epsilon}_{0}^{k} \ ; \ \frac{\partial \mathbf{X}^k}{\partial t} + \mathbb{M}(\mathbf{X}^k) = \boldsymbol{\epsilon}_{\mathbb{M}}^{k}$$

Incremental loop:

3 From a current state $\mathbf{X}^k$, perform a backward estimation of $\boldsymbol{\lambda}(t)$ in an adjoint model that takes into account the observations:
$$\boldsymbol{\lambda}(t_f) = 0; \frac{\partial \boldsymbol{\lambda}}{\partial t} + (\partial_{\mathbf{X}^k}\mathbb{M})^* \, \boldsymbol{\lambda} = (\partial_{\mathbf{X}^k}\mathbb{H})^* \, R^{-1} \left(\mathbf{Y} - \mathbb{H}(\mathbf{X}^k)\right)$$

4 From the adjoint variable $\boldsymbol{\lambda}(t_0)$, we compute the incremental *control variable* and update the initial condition:
$$d\boldsymbol{\epsilon}_{0}^{k} = B\boldsymbol{\lambda}(t_0) \ ; \ d\mathbf{X}^{k}(t_0) = d\boldsymbol{\epsilon}_{0}^{k}$$

5 From the adjoint variables $\boldsymbol{\lambda}(t)$ and the updated initial condition, we compute the *control variable* and estimate all
states $d\mathbf{X}^k(t)$ using :
$$d\boldsymbol{\epsilon}_{\mathbb{M}}^{k}(t) = Q\boldsymbol{\lambda}(t) \ ; \ \frac{\partial d\mathbf{X}^k}{\partial t} + \partial_{\mathbf{X}^k}\mathbb{M} d\mathbf{X}^k = d\boldsymbol{\epsilon}_{\mathbb{M}}^{k}$$

6 Update the system state and the control variables :
$$\mathbf{X}^{k+1} = \mathbf{X}^k + d\mathbf{X}^k$$
$$\boldsymbol{\epsilon}_{0}^{k+1} = \boldsymbol{\epsilon}_{0}^{k} + d\boldsymbol{\epsilon}_{0}^{k} \ ; \ \boldsymbol{\epsilon}_{\mathbb{M}}^{k+1}(t) = \boldsymbol{\epsilon}_{\mathbb{M}}^{k}(t) + d\boldsymbol{\epsilon}_{\mathbb{M}}^{k}(t)$$

7 Set $k = k + 1$ and loop to 3) until convergence

Finally we obtain the control variables $\boldsymbol{\epsilon}_0 = \boldsymbol{\epsilon}_0^k$ and $\boldsymbol{\epsilon}_{\mathbb{M}}(t) = \boldsymbol{\epsilon}_{\mathbb{M}}^k(t)$

This algorithm enables to estimate, in a set of backward/forward integration steps, the overall control variables $(\boldsymbol{\epsilon}_0, \boldsymbol{\epsilon}_{\mathbb{M}})$ required in equations (7.15,7.16) for the estimation of the states $\mathbf{X}(t)$. The adjoint variables $\boldsymbol{\lambda}$ are indicators about the discrepancy between the corrected simulation and the constraints. To start the control a first assumption of $\mathbf{X}(t)$ is needed. It is usually obtained by the direct integration of $\mathbb{M}(\mathbf{X})$, and according to an initial condition $\mathbf{X}_0$ (step 2 of the previous process).

The above technique is in fact simply a gradient descent where the adjoint variables have been introduced. Such methodology is in general very efficient, provided that the initialization is not too far from the final solution. For the manipulation of continuous data associated to Eulerian models, there exists efficient techniques to ensure a good initialization, as for example multi-resolution schemes. However here, because of several specific Lagrangian strategies as for example collision avoidance (particles/particles and particles/obstacles), the temporal integration of a Lagrangian system can generate very different solutions for two closed initial conditions. As a consequence, depending on the user constraints, the initial integration of the system state in step 2 of EDITING PROCESS 1 is likely to be far from the expected solution, yielding some numerical difficulties: algorithm locked in a loop, time-consuming and error-prone techniques. This kind of issue is not new and is in general solved using Monte-Carlo techniques [201, 202] or using a selection on a set of pre-computed trajectories [203]. However in our application, both strategies are ineffective since the system state is too big and would require a number of particle/pre-computed trajectories too large regarding to the actual computational capabilities. To face this issue we suggest, after

several iteration steps $\mathcal{N}$ fixed by the user, to *re-initialize the system* by a new simulation in the complete model, this time using the current solution of the system state and control variables as initial conditions. This enables to redefine a more consistent initial trajectory yielding a robustness with respect to local minima. This new procedure leads to the modified editing process:

> EDITING PROCESS 2: AUGMENTED INCREMENTAL TECHNIQUE USING THE ADJOINT METHOD.
>
> Fix an iteration step $\mathcal{N}$
> Initialization: set an initial iteration $k = 0$
>
> 1. Set control variables to zeros
> $$\boldsymbol{\epsilon}^k_{\mathbb{M}}(t) = 0 \; ; \; \boldsymbol{\epsilon}^k_0 = 0.$$
>
> Incremental loop: while the convergence is not reached
>
> 2 From $\mathbf{X}_0$, obtain a simulation in a perfect model $\mathbb{M}$ (step 2 of EDITING PROCESS 1) using current values of $\boldsymbol{\epsilon}^k_{\mathbb{M}}(t)$ and $\boldsymbol{\epsilon}^k_0$
>
> 3 Perform $\mathcal{N}$ times steps (3–7) of EDITING PROCESS 1
>
> 4 Perform a re-simulation of the initial condition by looping to step 2
>
> Finally we obtain the control variables $\boldsymbol{\epsilon}_0 = \boldsymbol{\epsilon}^k_0$ and $\boldsymbol{\epsilon}_{\mathbb{M}}(t) = \boldsymbol{\epsilon}^k_{\mathbb{M}}(t)$

7.2.3. Results

We present here some results obtained with this control strategy respectively applied with per-pedestrian constraints and continuum-related constraints. We note that the observation operators and the associated derivation can be found in *[4]* .

7.2.3.1. Per-pedestrians constraints

Individual position/velocity constraints

The first experiment aims at testing the procedure by controlling the positions y and the velocities u of only two pedestrians in the crowd, yielding a simple observation operator $\mathbb{H}$ which is the identity. From a starting group of 38 members, two people are given a rendez-vous constraint at a given time $t = 20\,s$ and in a specific position (illustrated in Figure 35). Hence, the editing process has to find a path for both individuals through the group to fulfill the constraint. As the optimization is performed globally for all individuals, it is interesting to observe that other pedestrians help them in finding a solution, as can be seen in the images of Figure 35 and in the accompanying video.

Of course, it is important to mention that such a scenario using only a simulation model would be very awkward to obtain, and would require for both pedestrians to wait for each other at the meeting point. Let us now turn to the second experiment on individual constraints, which involves two groups of interacting people with the same kind of constraints.

Letter constraints

The second experiment considers two groups of people, each one evolving in opposite directions, in which we aim at making each group to form, after crossing each others, the letters "SG". Here, the operator is again the identity since we impose the positions $\mathbf{y}$ of the pedestrians.

As can be observed in figure 36(a), the simulation without any control creates two homogeneous flows without any specific pattern. In this experiment, the major difficulty comes from the melting of the trajectories that is indeed difficult to correctly control since the simulation presents some chaotic attributes from two closed initial conditions. This issue, already mentioned in section 7.2.2.2, is in practice faced using the augmented procedure of EDITING PROCESS 2.

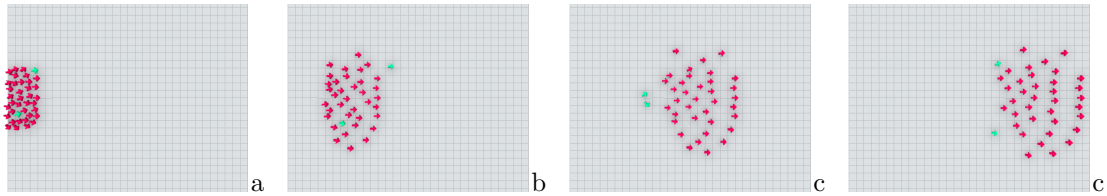


Figure 35.: **Rendez-vous experiment** In these illustrations, pedestrians are represented by arrows to clearly distinguish the two involved pedestrians . The meeting point is located behind the group, and should be reached at $t = 20s$. (a) $t = 0s$ (b) $t = 10s$ (c) $t = 20s$ (d) $t = 30s$

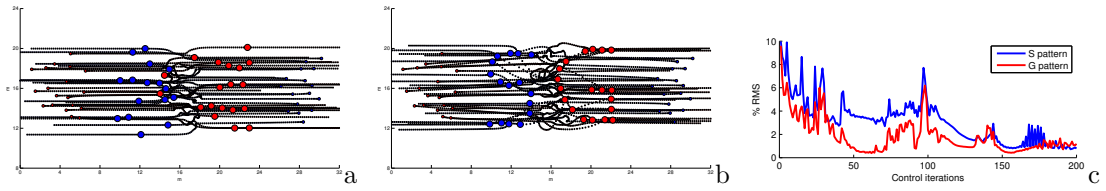


Figure 36.: **Letters experiment** Two groups of pedestrians, in blue (initially in the right hand side of the scene) and red (initially in the left hand side), are evolving in opposite directions. Figure (a) represents the trajectories at regular time steps obtained after the simulation of the model. In figure (b), we present the trajectories after 200 interations of the control process. The two pattern S and G clearly appear. Figure (c) depicts the associated RMS between real and desired positions.

The complete experimentation as well as comparisons with a direct simulation model aiming at forming the required configuration are visible in the accompanying video. In figure 36(a), we illustrate the trajectories of the two groups without any control whereas figure 36(b) presents trajectories after 200 iterations of the editing process, highlighting the benefit of the control process. As shown in the videos, the remaining trajectories are consistent and yield a more natural evolution than using a direct model to reach this goal without control. To assess some quantitative values, the figure 36(c) shows the evolution of the Root Mean Square (RMS) errors along the iterations between required and actual positions for the two groups of individuals. It is first interesting to observe that globally the RMS decrease along iterations, illustrating the benefit of the proposed editing process. The large variations observed (as around iteration 100) are clearly due to the chaotic behavior of lagrangian simulation models that are likely to generate different scenario for two closed configurations. Despite this variations, it is nevertheless satisfactory to observe that the global RMS decreases along iterations, finally yielding a consistent solution.

Comparing to other solution handling the shape control problem, such as the interpolation of Laplacian coordinates as proposed by Takahashi and colleagues [157], our method produces trajectories which match as much as possible the dynamics induced by the simulation model, which is not the case in most geometric approaches.

7.2.3.2. Continuum constraints

Motion transfer from a video Here, we follow the idea proposed in [155, 154], which consists in capturing a velocity field from a video, and then use it as a constraint to modify the global crowd motion. We refer to this idea as *video-based motion transfer* for crowd. The experiment is illustrated in Figure 41. From an abstract video of moving shapes (courtesy of BBC motion video), a dense velocity field is extracted with a Lucas-Kanade filter [204]. This time varying velocity field

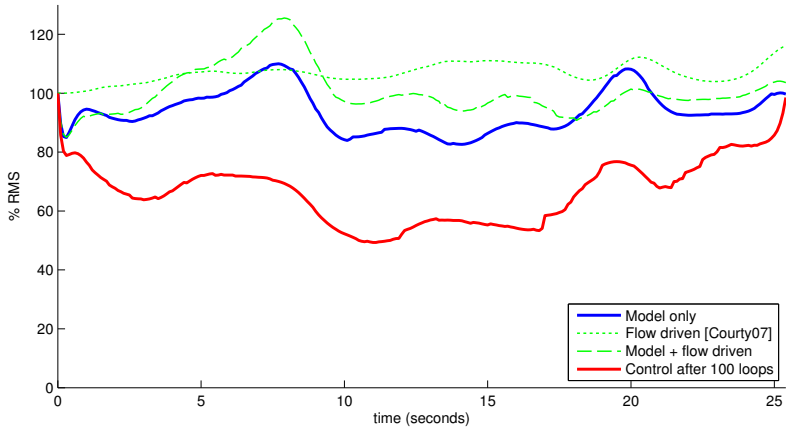


Figure 37.: **Comparisons between methods for the video-based motion transfer**, Evolution of the normalized RMS along the animation

then serves as a time varying constraint in the proposed control process. Extracts of the results are presented in Figure 41 and can be seen in the accompanying video.

For this example, we measure the quality of our approach by estimating how much of the motion information given by the flow constraint is retained in the final animation. This is accomplished by computing the spatial normalized RMS between the desired velocity fields and the observed ones at each time step. Results are presented in Figure 37. The initial normalized RMS of the direct output of the simulation model and the controlled version are presented. For comparisons sake, the smae value was computed for the simple advection proposed in [155], and for a direct simulation of the model where the desired velocity field $\mathbf{U}$ was simply added to the velocity term u_i. This way of integrating the motion field in the evolution equation is classical for most flow field based approaches (see for instance [152] or [154]). From all the four cases, our method achieves the best RMS performance. Let us note that this RMS is usually above 50% since it is computed above the entire Eulerian domain (a 32×16 grid), and that it is not covered with pedestrians. Here, and contrary to [155], it is important to note that the produced pedestrian trajectories match to a certain extent the original simulation model (for instance, no collisions between individuals). This was not the case in [155], since individual motions were only obtained by advecting the individuals along the velocity field, which of course does not prevent individuals to collide.

Flow regulation with vorticity control In this experiment four groups of people are trying to reach the opposite exits in a cross-shaped corridor (see Figure 38 for a schematic of the scene). This kind of situation is very frequently described in other works as an interesting configuration for the examination of emerging behaviors, see for instance Ref. [147] or [144]. Our aim is to show that it is possible to change the magnitude of the bottleneck by imposing a global constraint (*i.e.* at the environment level) on the observed vorticity, which is closely related to the whirling of individuals in the crowd flow. This example is particularly tricky, since it involves a lot of pedestrian interactions in a confined space. We suggest to face this issue by imposing to the global flow a rotation in the crossing area through a vorticity operator $\mathbb{H}$.

The vorticity describes rotating structures in a flow and is obtained by the Lagrangian cross-product differentiation of the Lagrangian velocities .This operator is thus highly non-linear. It's expression and linearization is given in [205]. Hence, we impose a vorticity constraint for a duration of one second bewteen the 16th and the 17th second of the animation (Figure 39.b). The original observed vorticity (Figure 39.a) in the output of the simulation model shows quasi-random positive and negative values, which simply translates the fact that pedestrians are bumping into each other, and that each pedestrians tries to find out its own path toward the exit. One can observe the vorticty pattern obtained at the end of the control process (Figure 39.c) matches much better the

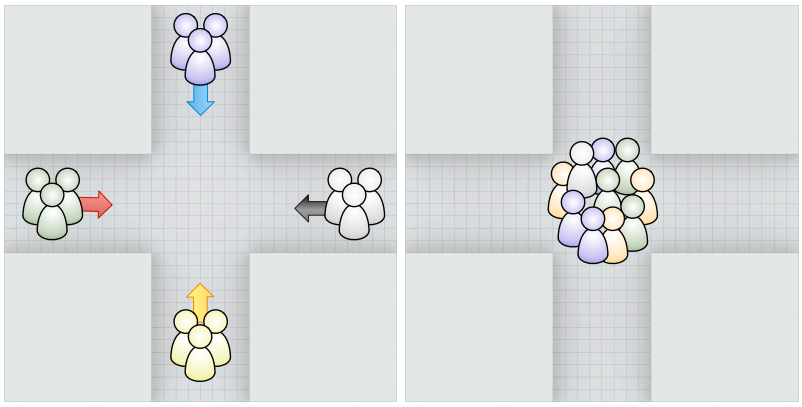


Figure 38.: In this example, four groups of 38 people are trying to reach the opposite side, creating a congestion in the crossing zone.

constraint. The resulting paths can be seen in the accompanying videos and in Figure 40. It is

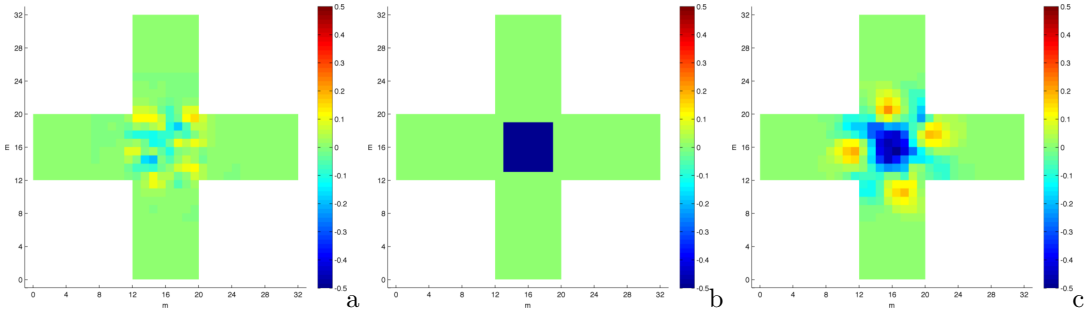


Figure 39.: Observed vorticity in the scene at time $t = 12.5$s. The grid dimension is 32×32. (a) Observed density in the direct output of the simulation model (b) user constraint (homogeneous positive vorticity in the crossing zone) (c) output of the control procedure

noticeable that at the beginning of the sequence, each group has chosen a side at that at the instant of the constraint a whirling pattern has emerged as a result of the control process.

7.3. Summary

This Section was concerned with the definition of data-driven crowd simulation methods. We have proposed:

- a pure data-driven method, which uses time series of dense velocity fields to animate pedestrians. We have shown that though convincing, the produced results fail in reproducing consistent individual trajectories. It seems tedious to do without an underlying model which would guarantee the consistency of the produced trajectories ;
- a new control paradigm which purpose is to mix a simulation model and some high level constraints. Typically, those constraints can be obtained by crowd video analysis. In this work, the coupling between the model and the constraints was performed thanks to optimal control.

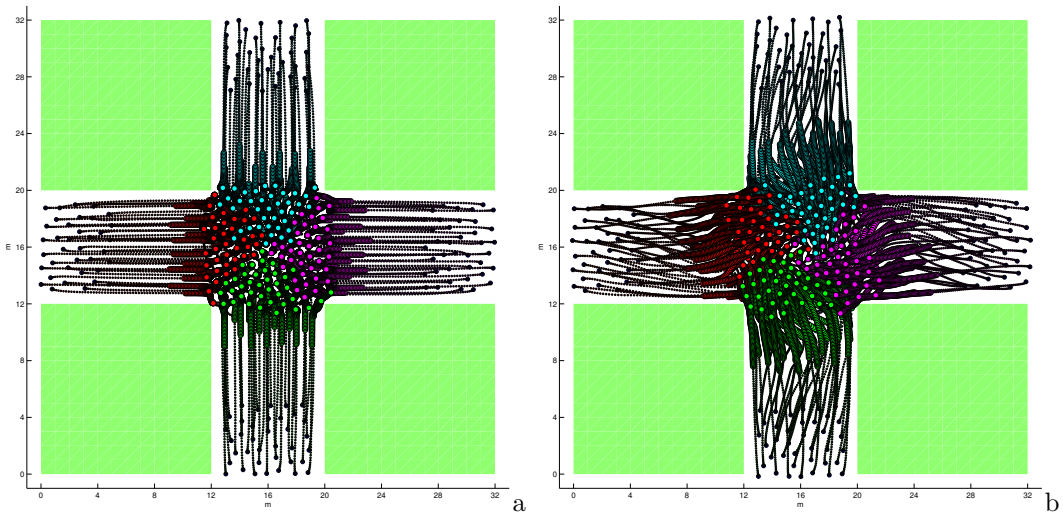


Figure 40.: Trajectories of pedestrians for a duration of 20 seconds. The position of the individuals between the $t = 10\,s$ and $t = 15\,s$ are highlighted: (a) direct output of the simulation model (b) output of the assimilation procedure.

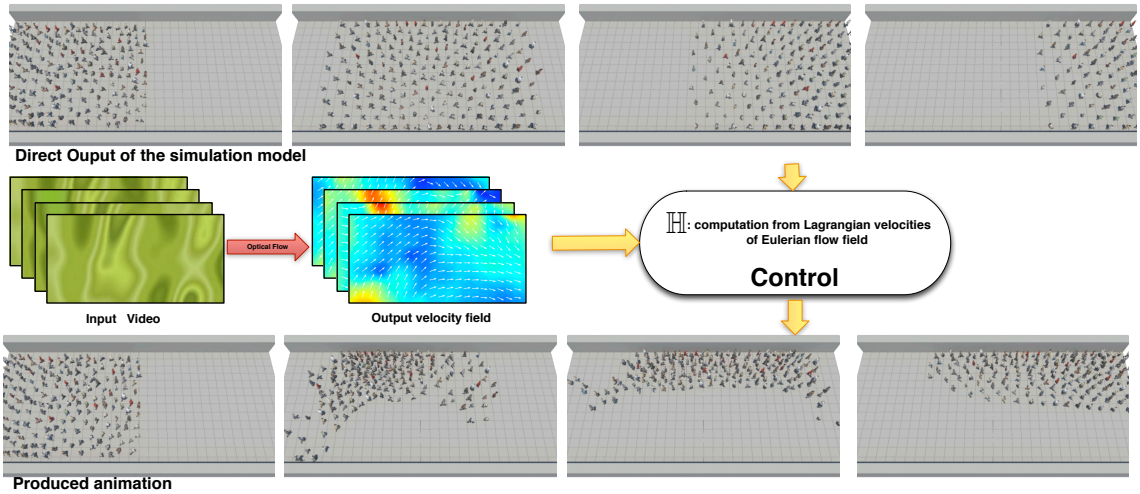


Figure 41.: **Video-based motion transfer** In this experiment the aim is to transfer the motion estimated from a given video to the crowd. A time varying motion field is first estimated thanks to an optical flow estimator. Then, this flow is used as constraints in our control procedure. The observation operator $\mathbb{H}$ relates here the pedestrians (Lagrangian) velocities to the corresponding continuum (Eulerian) flow field.

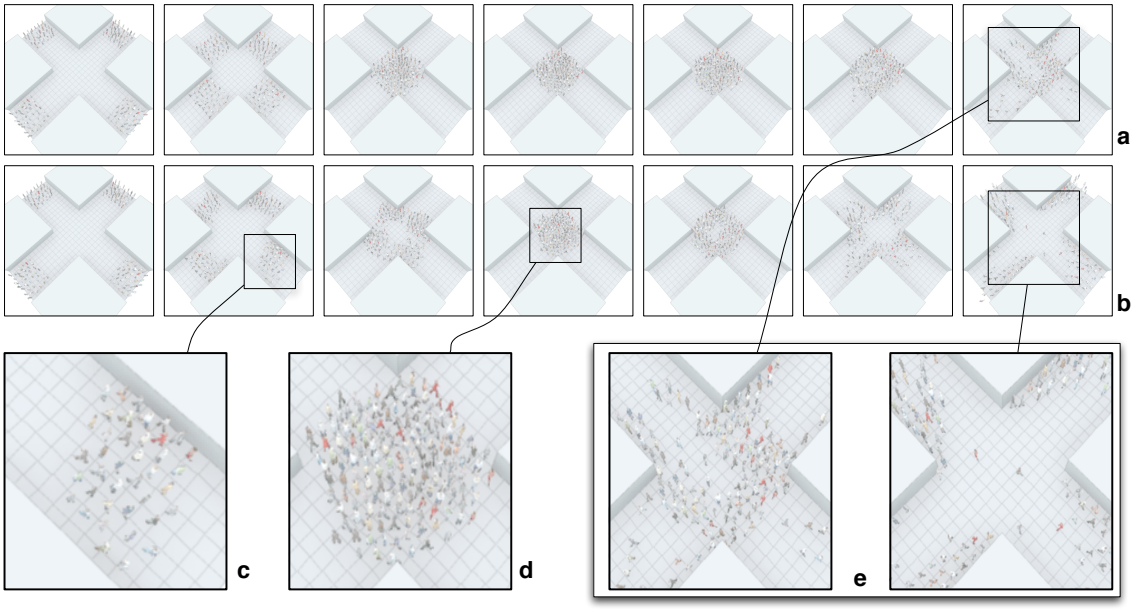


Figure 42.: **Vorticity control experiment.** Comparison of seven rendered frames captured at the same instants of the (a) direct output of the simulation model (b) controlled simulation after 25 iterations of the control procedure. In focus (c) one can see that the group anticipates the crossing zone by adopting an asymmetrical shape (d) a whirling pattern emerges from the control, and finally (e) illustrates the differences of evacuation time between the two simulations.

8

Perspectives and ongoing works

In this Part of the document, the problem of data-driven crowd simulation and analysis has been presented. Our primary goal was to be able to extract valuable information from real footages of crowds, and use this information in a simulation process. We have mainly focused on medium to high density crowds, which prevents individual tracking and pedestrian-based approaches to work. It was then natural to consider hybrid methods involving continuum dynamics and individual steering strategies. From a methodological point of view, we found that an efficient way to mix observations with a priori over the crowd dynamics was to use the variational assimilation framework. In this setting, it was possible to guide the analysis of crowd videos and extract quantities that can not be observed directly in the image. This framework also allowed to formulate the crowd control problem which amounts to edit a simulation a posteriori with constraints that can be related to continuous values (such as density or higher order quantities) or individual quantities. However, several issues were raised by those studies and could be the subject of future works.

Control of highly dynamic system: toward a multi scale approach The control of highly dynamic system is a very tedious task. In the case of a crowd, and because of the potential barriers due to the non-penetration constraints between pedestrians, the control generally boils down to the minimization of a highly non-convex energy function. In our previous works *[198]* , we considered the use of adjoint operators that allow merely a gradient descent approach, combined with an annealing-like procedure. Though the results are quite interesting, the convergence properties of the algorithm remains unclear, as well as the guarantees to find an acceptable solutions. It could be interesting to test Monte-Carlo like strategies or stochastic gradient that could more easily cope with the pedestrian to pedestrian interactions.

Stochastic simulation and analysis models . For now, the simulation model is known to be imperfect and as such incorporates an error term, which can be seen as a control term, *i.e.* a degree of variability that can be used to add constraints to the system. However, it could be very interesting to consider this term more generally as an expression of the non deterministic nature of the crowd. Most of the crowd simulation models are deterministic, and for a given situation a fixed initialization state would produce the same output. However, it is very unlikely to observe the same configurations in reality for similar situations (e.g. entrance of a railway station at a rush hour). How should the level of variability between the simulation model and what can be observed should be quantified ? How informative is the output of a simulation which will never be observed in real life ? It can be interesting to see the simulation output as a realization of a stochastic process. One should seek therefore to express the random nature directly in the expression of the model. The control problem could also be cast in such a formulation, where only the expected value of the simulation could be controlled, and not the fine interactions between individuals.

Learning from synthetic data. This part has highlighted the possible virtuous circle in the analysis/synthesis loop. Another possible instance of this principle lies in the use of synthetic gen-

erated sequence to produce learning database. Hence, new applications that were not conceivable in the past are made possible thanks to the quality of the produced images. A good example can be given through the pedestrian segmentation problem. It is indeed very difficult to get in large quantities ground truth of occlusions it was almost impossible to use machine learning techniques for people-in-crowd segmentation systems. By providing realistic body-to-body occlusions of dense crowds in videos with associated ground truth segmentation mask, it is now possible to learn local occlusions and motion patterns and transfer the associated learned segmentation to previously unseen data. Such holistic approaches to video analysis have been successfully applied in the past for people-in-crowd tracking [167]. It might now be possible to use it for other crowd analysis problems such as segmentation or fine crowd density estimation. We have recently started two applications of this principle, in the form of a density estimator and segmentation algorithms for medium to high density crowd videos. In Figure 43, an excerpt of a synthetic database of occlusion masks is shown. This database has been created with sequences from the Agoraset dataset, but with a dome of camera that allowed to render 64 views of the segmentation synthetic ground truth. All the masks from the entire sequence are then gathered, and used in a learning strategy.

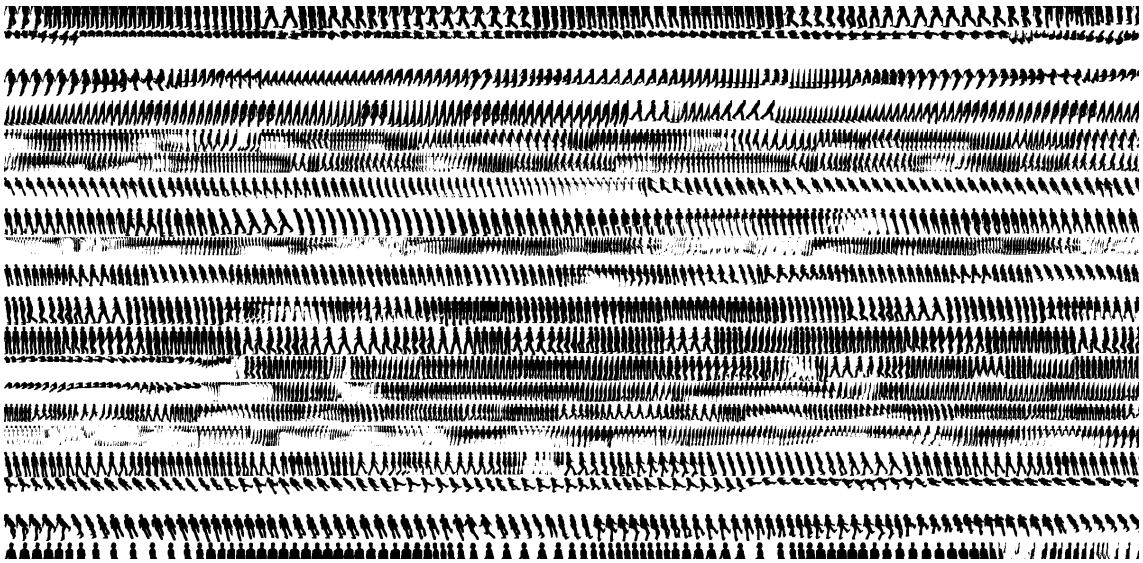


Figure 43.: **Learning occlusion masks from a synthetic ground truth. Excerpt from the** 886963 **occlusion masks extracted from the corridor scene of Figure 42**.

Part III.

Final conclusion and perspectives

Final conclusion and perspectives

Conclusion In the presented works we have proposed solutions to some inherent problems of data-driven animation. Two main application frameworks were considered: human character animation, with a special care to communication gestures and sign language, and crowd analysis/simulation. We have observed that acquiring quality data may require to have a prior knowledge of the observed phenomenon, and that this prior knowledge can also necessitate data to build an associated consistent model. Though the majority of these works are concentrated around the topic of animating from data, we have also demonstrated in the case of crowds how to guide data acquisition from existing models.
The main challenges were concerned with the **data "abstraction"**, *i.e.* our possibilities to build generic knowledge from some examples, and **data "integration"**, *i.e.* our capacity to use this knowledge in a control loop where a user can specify new behaviors at a task level. From a methodological point of view, we have examined statistical methods that have been adapted to the particular types of data that were considered, and also variational techniques that helped in handling physical models expressed by partial derivative equations.
Also, as most of our matter of interest involves a human component, we have been confronted with the difficulties of handling data that bury a strong semantic component. This semantic level, which is not accessible directly from the low level information, has nevertheless to be preserved. This somehow constitutes some definitive limitations of pure signal processing techniques applied to human motions, and it seems that this semantic knowledge, when available, should be added at a certain point in the processing loop.

General perspectives We present here some possible future directions of research for the considered problematic:

- **Increase in the quality and quantity of data**. The democratization of the acquisition systems, which also tend to become less costly, and the qualitative augmentation of the performances of the estimation algorithm bring inevitably to an explosion of the quantity of available data. While data-driven techniques were restricted to "controlled" acquisition conditions (using a motion capture room with high speed cameras is restricted to researchers, video game or visual effects companies), it seems possible to benefit from a lot of new captors (that can capture alternative quantities such acceleration or muscle electric activities for example) and use the web dissemination power and sharing capabilities to build motion models. In this sense, we can foresee the potential interests of the computer animation community to the context of "Big Data" and the associated problems: capture, curation, storage, search, sharing, analysis and visualization. While it is tempting to see in this trend the advent of the data era, which lowers the importance of the models and the theory because it is already "contained" into the data, I believe that coupling this data to an existing knowledge, expressed in the form models, is almost mandatory. It remains that a special care should be given to algorithms that are able to process a very large amount of data. On the methodological side, new methods have emerged from the machine learning theory, such as random projections, matrix factorization or sparsity enforcing algorithms. Their developments and applications to data driven animation methods seem very appealing ;
- **The duality signal/semantic**. While considering human activities and motions the problem of semantic arises. The associated data have a strong meaning, and small changes in the corresponding signal can lead to big differences in the inherent semantic. While we have only considered algorithms which measure distances, similarities or correlations in the signal space, only a few concern have been given to measure performances in terms of semantic content. In computer animation, the only existing approaches tend to measure a posteriori if the meaning of an action or motion has been preserved by user studies. It could be interesting to also consider directly in the control loop metrics related to the semantic content, as

for example it has been done in image processing with perception metrics. One can envisage several levels between the signal and semantic levels, from motor coordination to syntactic structures in the discourse. The same principles are not limited to human character animation. In the case of a crowd, analyzing from the data the general motivations of the group and its behavior could help in producing better data-driven techniques by incorporating a level of control based on group motivations and social laws.

Context of my future researches As for me, the barycenter of my research interests has drifted from computer animation/graphics to machine learning and computer vision problems. Therefore I'm in the process of addressing new applications, or at least different from pure computer animation. In this direction, I am participating to the creation of a new Irisa team "Obelix" (french acronym for **Ob**servation of **E**nvironment with Comp**L**e**X** **I**magery), which will be centered on the understanding of environmental systems through observations. Here again, large amount of data are available from local probes, sensor networks or hyper-spectral remote sensing images. Those data incorporate a time dimension, are by essence multidimensional and can also be related to physical models. I see in these problematics a lot of common features with my past researches, but applied to new interesting challenges with major societal components.

Bibliographic references

Références

[1] Nicolas Courty. *Animation référencée vision : de la tâche au comportement.* PhD thesis, INSA de Rennes, 2002.

[2] Alexis Héloir. *Agent virtuel signeur - Aide à la communication des personnes sourdes.* PhD thesis, University of Bretagne Sud, 2008.

[3] Charly Awad. *Indexation et Interrogations de Bases de Données de Mouvements pour l'Animation d'Humanoides Virtuels.* PhD thesis, University of Bretagne Sud, 2010.

[4] Pierre Allain. *Analyse et synthèse de mouvements de foule par contrôle optimal.* PhD thesis, University of Bretagne Sud, 2012.

[5] N.I. Badler, C.B. Phillips, and B.L. Webber. *Simulating humans: computer graphics animation and control.* Oxford University Press, USA, 1993.

[6] W. A. Wolovich and H. Elliot. A computational technique for inverse kinematics. In *Proc. of 23rd IEEE Conf. on Decision and Control*, pages 1359–1363, 1984.

[7] C. Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Frasier University, September 1993.

[8] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 13(4):313–336, 1994.

[9] R. Boulic, R. Mas, and D. Thalmann. A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics*, 20(5), 1996.

[10] K. Yamane and Y. Nakamura. Natural motion animation through constraining and deconstraining at will. *IEEE Tra. on Visualization and Computer Graphics*, 09(3):352–360, 2003.

[11] P. Baerlocher and R. Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6):402–417, 2004.

[12] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4), 2005.

[13] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, 1998.

[14] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. *ACM Trans. on Graphics*, 24(3):1062–1070, 2005.

[15] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, September 1981.

[16] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Trans. on Graphics*, 21(3):473–482, 2002.

[17] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21(3):491–500, 2002.

[18] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 559–568, New York, NY, USA, 2004. ACM.

[19] J. Chai and J.K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. *ACM Trans. on Graphics*, 26(3):686–696, July 2007.

[20] Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *Siggraph 2002, Computer Graphics Proceedings*, pages 465–472, 2002.

[21] S. Carvalho, R. Boulic, and D. Thalmann. Interactive Low-Dimensional Human Motion Synthesis by Combining Motion Models and PIK. *Computer Animation & Virtual Worlds*, 18(3), 2007.

[22] J. M. Wang, D. J. Fleet, and A Hertzmann. A. gaussian process dynamical models for human motion. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, pages 283–298, 2008.

[23] L. Ikemoto, O. Arikan, and D. Forsyth. Generalizing motion edits with gaussian processes. *ACM Trans. Graph.*, 28(1):1–12, 2009.

[24] M. Lau, Z. Bar-Joseph, and J. Kuffner. Modeling spatial and temporal variation in motion data. *ACM Trans. Graph. (Siggraph Asia)*, 28(5), 2009.

[25] K. Pullen and C. Bregler. Animating by multi-level sampling. In *Proc. of Computer Animation*, pages 36–42, 2000.

[26] M. Brand and A. Hertzmann. Style machines. In *Siggraph 2000, Computer Graphics Proceedings*, pages 183–192, 2000.

[27] J. Chai and J. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. on*

Graphics, 24(3):686–696, 2005.

[28] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Trans. on Graphics*, 23(3):522–531, August 2004.

[29] R. Urtasun, D. J.Fleet, and P. Fua. Gaussian process dynamical models for 3d people tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2006.

[30] J. M. Wang, D. J. Fleet, and A. Hertzmann. A. multifactor gaussian process models for style-content separation. In *Proc. of int. conf. on Machine Learning (ICML)*, June 2007.

[31] Okan Arikan, David A. Forsyth, and James F. O'Brien. Motion synthesis from annotations. *ACM Trans. Graph.*, 22(3):402–408, 2003.

[32] Shih-Pin Chao, Chih-Yi Chiu, Shi-Nine Yang, and Tsang-Gang Lin. Tai chi synthesizer: a motion synthesis framework based on key-postures and motion instructions: Research articles. *Comput. Animat. Virtual Worlds*, 15(3-4):259–268, 2004.

[33] K. Forbes and E. Fiume. An efficient search algorithm for motion data using weighted pca. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 67–76, 2005.

[34] Eamonn Keogh, Themistoklis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. Indexing large human-motion databases. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 780–791. VLDB Endowment, 2004.

[35] Suddha Basu, Shrinath Shanbhag, and Sharat Chandran. Search and transitioning for motion captured sequences. In *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 220–223, 2005.

[36] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24(3):677–685, 2005.

[37] I. Marshall and E. Safar. Grammar development for sign language avatar-based synthesis. In *In Proc. of the 3rd Int. Conf. on Universal Access in Human-Computer Interaction (UAHCI 2005)*, 2005.

[38] Y.H. Chiu, C.H. Wu, H.Y. Su, and C.J. Cheng. Joint optimization of word alignment and epenthesis generation for chinese to taiwanese sign synthesis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(1):28–39, January 2007.

[39] J. R. Kennaway, J. R. W. Glauert, and I. Zwitserlood. Providing signed content on the internet by synthesized animation. *ACM Trans. Comput.-Hum. Interact.*, 14(3):15, 2007.

[40] S. Fotinea, E. Efthimiou, G. Caridakis, and K. Karpouzis. A knowledge-based sign synthesis architecture. *Universal Access in the Information Society*, 6(4):405–418, 2008.

[41] M. Huenerfauth, L. Zhao, E. Gu, and J. Allbeck. Evaluation of american sign language generation by native asl signers. *ACM Trans. Access. Comput.*, 1(1):1–27, 2008.

[42] S. Gibet, T. Lebourque, and P.F. Marteau. High level specification and animation of communicative gestures. *Journal of Visual Languages and Computing*, 12:657–687, 2001.

[43] M. Filhol, A. Braffort, and L. Bolot. Signing avatar: Say hello to elsi!,. In *Proc. of Gesture Workshop 2007*, LNCS, Lisbon, Portugal, June 2007.

[44] C. Awad, N. Courty, K. Duarte, T. Le Naour, and S. Gibet. A combined semantic and motion capture database for real-time sign language synthesis. In *Proc of IVA*, volume 5773 of *LNAI*, pages 432–38. Springer-Verlag, Berlin, Heidelberg, 2009.

[45] A. Kendon. *Tools, Language and Cognition*, chapter Human gesture, pages 43–62. Cambridge University Press, 1993.

[46] D. McNeill. *Hand and Mind - What Gestures Reveal about Thought*. The University of Chicago Press, Chicago, IL, 1992.

[47] S. Kita, I. van Gijn, and H. van der Hulst. Movement phase in signs and co-speech gestures, and their transcriptions by human coders. In *Proc. of the Int. Gesture Workshop*, volume 1371 of *LNCS*, pages 23–35. Springer-Verlag, London, 1997.

[48] M. Kipp, M. Neff, K. Kipp, and I. Albrecht. Toward natural gesture synthesis: Evaluating gesture units in a data-driven approach. In *Intelligent Virtual Agents (IVA'07)*, pages 15–28, 2007.

[49] William C. Stokoe. *Semiotics and Human Sign Language*. Walter de Gruyter Inc., 1972.

[50] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, and J. Henning. *Hamburg Notation System for Sign Languages - An Introductory Guide*. University of Hamburg Press, 1989.

[51] Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth F. Churchill. *Embodied Conversational Agents*. The MIT Press, 2000.

[52] R. Elliott, J. Glauert, V. Jennings, and J. Kennaway. An overview of the sigml notation and sigml signing

software system. In *Workshop on the Representation and Processing of Signed Languages, 4th Int'l Conf. on Language Resources and Evaluation*, 2004.

[53] A. Kranstedt, S. Kopp, and I. Wachsmuth. MURML: A Multimodal Utterance Representation Markup Language for Conversational Agents. In *Proceedings of the AAMAS02 Workshop on Embodied Conversational Agents - let's specify and evaluate them*, Bologna, Italy, July 2002.

[54] Han Noot and Zsófia Ruttkay. Variations in gesturing and speech by gestyle. *Int. J. Hum.-Comput. Stud.*, 62(2):211–229, 2005.

[55] B. Hartmann, M. Mancini, and C. Pelachaud. Implementing expressive gesture synthesis for embodied conversational agents. *Gesture in Human-Computer Interaction and Simulation*, 3881:188–199, 2006.

[56] H. Vilhalmsson, N. Cantelmo, J. Cassell, N.E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A.N. Marshall, C. Pelachaud, Z. Ruttkay, K. Thorisson, H. van Welbergen, and R.J. van der Werf. The behavior markup language: Recent developments and challenges. In *IVA 2007*, 2007.

[57] D. Tolani, A. Goswami, and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, 62(5):353–388, 2000.

[58] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *Journal Computer Animation and Virtual Worlds*, 15(1):39–52, 2004.

[59] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics*, 27(1):233–51, March 2008.

[60] Z. Deng, P.i-Y. Chiang, P. Fox, and U. Newmann. Animating blendshape faces by cross-mapping motion capture data. In *Proc. of the 2006 symp. on Interactive 3D graphics and games*, pages 43–48, Redwood City, California, March 2006.

[61] O. Arikan, D. Forsyth, and J. O'Brien. Motion synthesis from annotations. *ACM Trans. on Graphics*, 22(3):402–408, July 2003.

[62] Sylvie Gibet, Nicolas Courty, Kyle Duarte, and Thibaut Le Naour. The signcom system for data-driven animation of interactive virtual signers: Methodology and evaluation. *ACM Transactions on Interactive Intelligent Systems*, 1(1):6:1–6:23, October 2011.

[63] G.. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-from-silhouette with stereo. In *CVPR*, pages 375–382, 2003.

[64] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *Int. Journal of Computer Vision*, 61(2):185–205, 2005.

[65] R. Raskar, H. Nii, B. DeDecker, Y. Hashimoto, J. Summet, D. Moore, Y. Zhao, J. Westhues, P. Dietz, M. Inami, S. Nayar, J. Barnwell, M. Noland, P. Bekaert, V. Branzoi, and E. Burns. Prakash: Lighting-aware motion capture using photosensing markers and multiplexed illumination. *ACM Trans. Graph.*, 26(3), August 2007. session : Performance Capture.

[66] K. Shoemake. Quaternions and 4×4 matrices. In James Arvo, editor, *Graphics Gems II*, pages 351–354. Academic Press, 1991.

[67] J. Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA 2004)*. IEEE, May 2004.

[68] T. Tangkuampien and D. Suter. Human motion de-noising via greedy kernel principal component analysis filtering. In *ICPR*, pages 457–460, 2006.

[69] A. Héloir, N. Courty, S. Gibet, and F. Multon. Temporal alignment of communicative gesture sequences. *Computer Animation and Virtual Worlds*, 17:347–357, July 2006.

[70] Nicolas Courty. Bilateral Human Motion Filtering. In *Proc. of the 16th European Signal Processing Conference (EUSIPCO 2008) Proc. of the 16th European Signal Processing Conference (EUSIPCO 2008)*, pages 1–5, Lausanne Suisse, 2008.

[71] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998.

[72] E. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM Trans. Graph*, 24(3):845–852, 2005.

[73] C. Liu, W. Freeman, R. Szeliski, and S. Bing Kang. Noise estimation from a single image. In *CVPR*, pages 901–908. IEEE Computer Society, 2006.

[74] B. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. In *Proc. of Siggraph*, pages 433–442, 2001.

[75] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph*, 21(3):257–266, 2002.

[76] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph*, 23(3):664–672, 2004.

[77] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Trans. Graph*, 25(3):637–645, 2006.

[78] H. Winnemöller, S. Olsen, and B. Gooch. Real-time video abstraction. *ACM Trans. Graph*, 25(3):1221–1226, 2006.

[79] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, volume 3951 of *Lecture Notes in Computer Science*, pages 211–224, 2006.

[80] W. Wong, A. Chung, and S. Yu. Trilateral filtering for biomedical images. In *ISBI*, pages 820–823. IEEE, 2004.

[81] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, 2002.

[82] A. Buades, B. Coll, and J.-M. Morel. Neighborhood filters and PDE's. *Numerische Mathematik*, 105(1):1–34, 2006.

[83] T. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph*, 22(3), July 2003.

[84] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Trans. Graph*, 22(3):950–953, July 2003.

[85] S. Paris, H. Briceño, and F.-X. Sillion. Capture of hair geometry from multiple images. *ACM Trans. Graph*, 23(3):712–719, 2004.

[86] J. Lee and S. Y. Shin. General construction of time-domain filters for orientation data. *IEEE Trans. on Visualization and Computer Graphics*, 8(2):119–128, 2002.

[87] J. Johnstone and J. Williams. Rational control of orientation for animation. In *Graphics Interface '95*, pages 179–186, May 1995.

[88] Y. C. Fang, C. C. Hsieh, M. J. Kim, J. J. Chang, and T. C. Woo. Real time motion fairing with unit quaternions. *Computer-Aided Design*, 30(3):191–198, 1998.

[89] Alexis Héloir, Nicolas Courty, Sylvie Gibet, and Franck Multon. Temporal alignment of communicative gesture sequences. *Computer Animation and Virtual Worlds (selected best papers from CASA'06)*, 17:347–357, 2006.

[90] M. P. Johnson. *Exploiting quaternions to support expressive interactive character motion.* PhD thesis, Massachusetts Institute of Technology, 2003.

[91] T. Fletcher, C. Lu, S. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Trans. Med. Imaging*, 23(8):995–1005, 2004.

[92] Salem Said, Nicolas Courty, Nicolas Lebihan, and Stephen J. Sangwine. Exact Principal Geodesic Analysis for Data on SO(3). In *Proceedings of the 15th European Signal Processing Conference, EUSIPCO-2007 15th European Signal Processing Conference, EUSIPCO-2007*, pages 1700–1705, Poznan Pologne, 2007. EURASIP. Département Images et Signal.

[93] S. Sommer, F. Lauze, and M. Nielsen. The differential of the exponential map, jacobi fields and exact principal geodesic analysis. *CoRR*, abs/1008.1902, 2010.

[94] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In *ECCV 2010*, volume 6316 of *LNCS*, pages 43–56. 2010.

[95] Maxime Tournier, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Reveret. Motion Compression using Principal Geodesics Analysis. *Computer Graphics Forum (Proceedings of Eurographics 2009)*, 2009.

[96] Nicolas Courty and Anne Cuzol. Conditional Stochastic Simulation for Character Animation. *Computer Animation and Virtual Worlds (best selected papers from CASA 2010)*, pages 1–10, 2010.

[97] Nicolas Courty, Thomas Burger, and Pierre-François Marteau. Geodesic Analysis on the Gaussian RKHS hypersphere. In *proceedings of ECML-PKDD 2012*, LNCS, Royaume-Uni, September 2012.

[98] L. Ljung. *System Identification - Theory For the User*. PTR Prentice Hall, Upper Saddle River, N.J., 1999.

[99] K.P. Körding and D.M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427:244–247, 2004.

[100] S. Gibet and P.F. Marteau. A self-organised model for the control, planning and learning of nonlinear multivariable systems using a sensori- feedback. *Journal of Applied Intelligence*, 4:337–349, 1994.

[101] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measures and Control*, 108:163–171, September 1986.

[102] A. Maciejewski. Dealing with the ill-conditioned equations of motion for articulated figures. *IEEE Computer Graphics and Applications*, 10(3):63–71, May 1990.

[103] N. Courty, E. Marchand, and B. Arnaldi. Through-the-eyes control of a virtual humanoÔd. In *Proc. of Computer Animation 2001*, pages 74–83, Seoul, South Korea, November 2001.

[104] B. Le Callennec and R. Boulic. Interactive motion deformation with prioritized constraints. *Graphical Models*,

68(2):175–193, 2006.

[105] E.A. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control*, 2000.

[106] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. *ACM Tra. On Graphics*, 24(1):98–117, 2005.

[107] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particles filters for online nonlinear / non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[108] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag, 2001.

[109] Nicolas Courty and Elise Arnaud. Sequential monte carlo inverse kinematics. Research Report 6426, INRIA, December 2007.

[110] M.L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics, 1999.

[111] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[112] C. Lantuéjoul. *Geostatistical Simulation*. Springer, 2002.

[113] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[114] Yong Cao, Wen C. Tien, Petros Faloutsos, and Frédéric Pighin. Expressive speech-driven facial animation. *ACM Transactions on Graphics*, 24(4):1283–302, October 2005.

[115] Zhigang Deng, Ulrich Newmann, J. P. Lewis, Tae-Yong Kim, Murtaza Bulut, and Shrikanth Narayanan. Expressive facial animation synthesis by learning speech coarticulation and expression spaces. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1523–34, November 2006.

[116] Xuecheng Liu, Tianlu Mao, Shihong Xia, Yong Yu, and Zhaoqi Wang. Facial animation by optimized blendshapes from motion capture data. *Computer Animation and Virtual Worlds*, 19(3-4):235–45, September 2008.

[117] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–68, March 2002.

[118] Xiaohan Ma and Zhigang Deng. Natural eye motion synthesis by modeling gaze-head coupling. In *2009 IEEE Virtual Reality Conference*, pages 143–50, Lafayette, Louisiana, USA, March 2009.

[119] Tateo Warabi. The reaction time of eye-head coordination in man. *Neuroscience Letters*, 6(1):47–51, October 1977.

[120] Alexis Héloir, Michael Kipp, Sylvie Gibet, and Nicolas Courty. Evaluating data-driven style transformation for gesturing embodied agents. In *Intelligent Virtual Agent (IVA 2008) Intelligent Virtual Agent (IVA 2008)*, volume 5208, pages 215–222, Tokyo Japon, 2008.

[121] R. McDonnell, M. Breidt, and H. Bülthoff. Render me real?: investigating the effect of render style on the perception of animated virtual humans. pages 91–99, 2012.

[122] M. Vicovaro, L. Hoyet, L. Burigana, and C. O'Sullivan. Evaluating the plausibility of edited throwing animations. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 175–182, 2012.

[123] L. Hoyet, F. Multon, A. Lecuyer, and T. Komura. Perception based real-time dynamic adaptation of human motions. In *Motion in Games*, volume 6459 of *LNCS*, pages 266–277. 2010.

[124] R. McDonnell, M. Larkin, S. Dobbyn, S. Collins, and C. O'Sullivan. Clone attack! perception of crowd variety. *ACM Trans. on Graphics*, 27(3):1–8, 2008.

[125] R. McDonnell, M. Larkin, B. Hernandez, I. Rudomín, and C. O'Sullivan. Eye-catching crowds: saliency based selective variation. 2009.

[126] A. Majkowska, V. B. Zordan, and P. Faloutsos. Automatic splicing for hand and body animations. In *SCA '06: Proc. of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 309–316, 2006.

[127] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[128] K. j. Choi and H. s. Ko. On-line motion retargetting. *Journal of Visualization and Computer Animation*, 11:223–235, 2000.

[129] R. Kulpa, F. Multon, and B. Arnaldi. Morphology-independent representation of motions for interactive human-like animation. *Comput. Graph. Forum*, 24(3):343–352, 2005.

[130] C. Hecker, B. Raabe, R. Enslow, J. DeWeese, J. Maynard, and K. van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. on Graphics*, 27(3):1–11, 2008.

[131] E. Ho, T. Komura, and C.-L. Tai. Spatial relationship preserving character motion adaptation. *ACM Trans. on Graphics*, 29(4):1–8, 2010.

[132] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour. An integrated approach to inverse kinematics and path planning for redundant manipulators. In *ICRA 2006: Int. Conf. on Robotic and Automation*, pages 1874–1879, 2006.

[133] L. Zhang, M. C. Lin, D. Manocha, and J. Pan. A hybrid approach for simulating human motion in constrained environments. *Computer Animation and Virtual Worlds*, 21(3–4):137–149, 2010.

[134] Thibaut Le Naour, Nicolas Courty, and Sylvie Gibet. Cinématique guidée par les distances. *Revue Electronique Francophone d'Informatique Graphique (REFIG)*, 6(1), June 2012.

[135] A. Héloir and S. Gibet. A qualitative and quantitative characterisation of style in sign language gestures. In *Gesture in Human-Computer Interaction and Simulation, GW 2007, Lecture Notes in Artificial Intelligence, LNAI*, Lisboa, Portugal, 2009. Springer Verlag.

[136] D. Thalmann and S.R. Musse. *Crowd Simulation*. Springer-Verlag London Limited, 2007.

[137] N. Pelechano, J. Allbeck, and N. Badler. *Virtual Crowds: Methods, Simulation, and Control*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, 2008.

[138] S. Raupp Musse and D. Thalmann. Hierarchical model for real time simulation of virtual human crowds. In *IEEE Trans. on Vis and Comp. Graph.*, volume 7(2), pages 152–164. IEEE Comp. Society, 2001.

[139] M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. *Comput. Graph. Forum*, 23(3):519–528, 2004.

[140] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Comp. Graph.:* Proc. of SIGGRAPH '87, 21(4):25–34, July 1987.

[141] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(1):487–490, 2000.

[142] S. Paris, J. Pettre, and S. Donikian. Pedestrian steering for crowd simulation: A predictive approach. *Computer Graphics Forum (Proc. of Eurographics)*, 26(3), 2007.

[143] S. J. Guy, J. Chhugani, S. Curtis, P. Dubey, M. C. Lin, and D. Manocha. Pledestrians: A least-effort approach to crowd simulation. In *Symposium on Computer Animation, SCA'10*, Madrid, Spain, August 2010. ACM SIGGRAPH/Eurographics.

[144] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2010)*, 29:123:1–123:9, July 2010.

[145] N. Pelechano and N. Badler. Modeling crowd and trained leader behavior during building evacuation. *IEEE Comput. Graph. Appl.*, 26(6):80–86, November 2006.

[146] R. L. Hughes. A continuum theory of pedestrian motion. *Transportation Res. B*, 36(6):507–535, June 2002.

[147] A. Treuille, S. Cooper, and Z. Popovic. Continuum crowds. *ACM Tran. on Graph., special issue, Proc. ACM SIGGRAPH 2006*, 25(3):1160–1168, 2006.

[148] L. Pimenta, N. Michael, R. Mesquita, G. Pereira, and V. Kumar. Control of swarms based on hydrodynamic models. In *International Conference on Robotics and Automation, ICRA'08*, pages 1948–1953. IEEE, 2008.

[149] R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH 2009)*, 28:122:1–122:8, December 2009.

[150] B. Ulicny, P. Ciechomski, and D. Thalmann. Crowdbrush: interactive authoring of real-time crowd scenes. In *Symposium on Computer Animation'04*, SCA '04, pages 243–252, 2004.

[151] S. Chenney. Flow tiles. In *Eurographics/ACM SIGGRAPH Symp. on Comp. Anim. (SCA'04)*, pages 233–242, Grenoble, France, August 2004.

[152] X. Jin, J. Xu, C. Wang, S. Huang, and J. Zhang. Interactive control of large-crowd navigation in virtual environments using vector fields. *IEEE Computer Graphics and Applications*, 28:37–46, 2008.

[153] M. Park. Guiding flows for controlling crowds. *The Visual Computer*, 26:1383–1391, 2010.

[154] S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha. Directing crowd simulations using navigation fields. *IEEE Transactions on Visualization and Computer Graphics*, 17:244 – 254, February 2011.

[155] N. Courty and T. Corpetti. Crowd motion capture. *Computer Animation and Virtual Worlds*, 18(4–5):361–370, 2007.

[156] T. Kwon, K. Hoon Lee, J. Lee, and S. Takahashi. Group motion editing. *ACM TOG., Proc. ACM SIGGRAPH 2008*, 27(3), 2008.

[157] S. Takahashi, K. Yoshida, T. Kwon, K. H. Lee, J. Lee, and S. Y. Shin. Spectral-based group formation control. *Computer Graphics Forum (Proc. Eurographics 2009)*, 28(2):639–648, 2009.

[158] B. Zhan, D.N. Monekosso, P. Remagnino, S.A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Mach.*

Vision Appl., 19(5-6):345–357, 2008.

[159] S.-Y. Cho, T.W.S. Chow, and C.-T. Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Tra. on SMC*, 29(4):535 –541, aug 1999.

[160] Carlo S. Regazzoni and Alessandra Tesei. Distributed data fusion for real-time crowding estimation. *Signal Processing*, 53(1):47 – 63, 1996.

[161] D.B. Yang, H.H. González-Ba nos, and L.J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 122, Washington, DC, USA, 2003. IEEE Computer Society.

[162] Evangelos Kalogerakis, Olga Vesselova, James Hays, Alexei Efros, and Aaron Hertzmann. You'll never walk alone: modeling social behavior for multi-target tracking. In *ICCV*, Kyoto, Japan, 2009.

[163] Ahmed Elgammal and Larry S. Davis. Probabilistic framework for segmenting people under occlusion. In *In Proc. of IEEE 8th International Conference on Computer Vision*, pages 145–152, 2001.

[164] S.-F. Lin, J.-Y. Chen, and H.-X. Chai. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Tra. on SMC*, 31(6):645–654, 2001.

[165] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *CVPR (2)*, pages 459–466, 2003.

[166] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, pages 705–711, New York, June 2006.

[167] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1–8, Kyoto, Japan, October 2009.

[168] A. Treuille, S. Cooper, and Z. Popovic. Continuum crowds. *ACM TOG., Proc. ACM SIGGRAPH 2006*, 25(3):1160–1168, 2006.

[169] P. Allain, N. Courty, and T. Corpetti. Crowd flow characterization with optimal control theory. In *ACCV*, Xi'an, China, 2009.

[170] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, pages 935–942, Los Alamitos, CA, USA, 2009.

[171] N. Courty and T. Corpetti. Crowd motion capture. *Computer Animation and Virtual Worlds*, 18(4–5):361–370, 2007.

[172] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *CVPR*, pages 1–6, Minneapolis, Minnesota, June 2007.

[173] E.L. Andrade, S. Blunsden, and R.B. Fisher. Modelling crowd scenes for event detection. In *ICPR*, pages 175–178, Washington, DC, USA, 2006.

[174] B. A. Boghossian and S. A. Velastin. Motion-based machine vision techniques for the management of large crowds. In *ICECS*, volume 2, pages 961–964 vol.2, August 2002.

[175] K. Lee, M. Choi, Q. Hong, and J. Lee. Group behavior from video: a data-driven approach to crowd simulation. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation, SCA'07*, pages 109–118, San Diego, California, August 2007.

[176] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum (Proc. of Eurographics)*, 26(3), 2007.

[177] B. Horn and B. Schunck. Determining optical flow. *Art. Intell.*, 17:185–203, 1981.

[178] Pierre Allain, Nicolas Courty, and Thomas Corpetti. Crowd Flow Characterization with Optimal Control Theory. In *Ninth Asian Conference on Computer Vision (ACCV 2009) Ninth Asian Conference on Computer Vision (ACCV 2009)*, pages 279–290, Xi'an Chine, 2009.

[179] J.-L. Lions. *Optimal control of systems governed by PDEs*. Springer-Verlag, 1971.

[180] F.X. Le-Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus*, pages 97–110, 1986.

[181] R. Kimmel and J Sethian. Optimal algorithm for shape from shading and path planning. *J. of Math. Ima. and Vis.*, 14(3):237–244, 2001.

[182] O. Talagrand. *Variational assimilation. Adjoint equations*. Kluwer Academic Publishers, 2002.

[183] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *In Proc. Seventh Int. Joint Conf. on Art. Intell.*, pages 674–679, Vancouver, Canada,, 1981.

[184] E. Andrade, S. Blunsden, and R. Fisher. Modelling crowd scenes for event detection. In *Int. Conf. on Patt. Recognitiion, ICPR 2006*, pages 175–178, 2006.

[185] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Mach. Vis. Appl.*, 19(5-6):345–357, 2008.

[186] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006.

[187] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, pages 1–8, Rio de Janeiro, Brazil, October 2007.

[188] L. Sigal, A. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 87(1):4–27, March 2010.

[189] F. Qureshi and D. Terzopoulos. Surveillance in virtual reality: System design and multi-camera control. In *CVPR*, Minneapolis, Minnesota, USA, June 2007.

[190] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *CVPR*, Minneapolis, Minnesota, USA, June 2007.

[191] S.R. Musse, M. Paravisi, R. Rodrigues, J.C.S. Jacques Jr, and C.R. Jung. Using synthetic ground truth data to evaluate computer vision techniques. In *Proc. of int. workshop on PETS*, pages 25–32, Rio de Janeiro, Brazil, October 2007.

[192] Soraia R. Musse, Cláudio R. Jung, Julio C. S. Jacques, and Adriana Braun. Using computer vision to simulate the motion of virtual agents. *Computer Animation and Virtual Worlds*, 18(2):83–93, 2007.

[193] M. Moussaid, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings of the Royal Society B: Biological Sciences*, 276(1668):2755–2762, 2009.

[194] Taras I. Lakoba, D. J. Kaup, and Neal M. Finkelstein. Modifications of the helbing-molnár-farkas-vicsek social force model for pedestrian evolution. *Simulation*, 81(5):339–352, May 2005.

[195] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *PHYSICAL REVIEW E*, 51:4282, 1995.

[196] T. Driemeyer. *Rendering with mental ray.* Springer, New York, 2001.

[197] Nicolas Courty and Thomas Corpetti. Crowd Motion Capture. *Computer Animation and Virtual Worlds (selected best papers from CASA 2007)*, 18(4–5):361–370, 2007.

[198] Pierre Allain, Nicolas Courty, and Thomas. Corpetti. Particle swarm control. Research Report 1997, IRISA, 2012.

[199] A. McNamara, A. Treuille, Z. Popović, and J. Stam. Fluid control using the adjoint method. *ACM Transactions on Graphics (Proc. SIGGRAPH 2004)*, 23:449–456, July 2004.

[200] C. Wojtan, P. Mucha, and G. Turk. Keyframe control of complex particle systems using the adjoint method. In *Symposium on Computer Animation'06*, pages 15–23, 2006.

[201] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1805 –1819, nov. 2005.

[202] K. Smith, D. Gatica-Perez, and J.-M. Odobez. Using particles to track varying numbers of interacting people. In *Int Conf on Comp. Vis. and Pat. Recogn., CVPR*, pages 962–969. CVPR, 2005.

[203] C. Twigg and D. James. Many-worlds browsing for control of multibody dynamics. *ACM Trans. Graph.*, 26, July 2007.

[204] B.D. Lucas. *Generalized image matching by the method of differences.* PhD thesis, School of Comp. Science, Carnegie– Mellon University, Pittsburgh, PA., 1984.

[205] AnonymousAuthors. Particle swarm control. *submitted to Swarm Intelligence*, 2012.

[206] C. Cortes and V. Vapnik. Support vector machine. *Machine Learning*, 20(3):273–297, 1995.

[207] B. Schölkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. *Artificial Neural Networks - ICANN'97*, pages 583–588, 1997.

[208] B. Schölkopf and A.J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* the MIT Press, 2002.

[209] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163, 2005.

[210] S. Said, N. Courty, N. LeBihan, and S. J. Sangwine. Exact principal geodesic analysis for data on so(3). In *Proceedings of EUSIPCO 2007*, Poznan, Poland, 2007.

[211] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, 30(5):509–541, 1977.

[212] W.S. Kendall. Convexity and the hemisphere. *Journal of the London Mathematical Society*, 2(3):567, 1991.

[213] S. Mika, B. Schölkopf, A.J. Smola, K.R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, pages 536–542. MIT Press, 1999.

[214] J. Kwok and I. Tsang. The pre-image problem in kernel methods. *IEEE Trans. on Neural Networks*, 15(6):1517–1525, 2004.

[215] D. Huang, Y. Tian, and F. De la Torre. Local isomorphism to solve the pre-image problem in kernel methods. In *CVPR'11*, pages 2761–2768, 2011.

A

Elements of Notations

In this document, the following choices have been adopted for mathematical notations:

- scalar and multivariate values:
  - α, λ: scalar values in lower case,
  - $\mathbf{u}$: vector values in bold lower case,
  - $\mathbf{x}$: denotes a generic multivariate data
  - $\hat{x}$ generally denotes an estimated value of variable x,
  - $\mathbf{q}$: denotes generally a rotation in $\mathbb{R}^3$ expressed as a quaternion
- Matrix values:
  - $\mathbf{M}$: matrix values in bold upper case,
  - $\mathbf{I}_n$: Identity matrix of rank n,
  - Σ: covariance matrix,
- spaces, group:
  - $\mathbb{R}^n$: n-dimensional Euclidian space,
  - $\mathbb{H}$: Hamiltonian quaternion space,
  - S^3: Unit hypersphere of dimension 3,
  - $SO(3)$: Special orthogonal group of dimension 3
  - $\mathfrak{so}(3)^n$: associated Lie algebra
- Riemannian geometry:
  - $\mathcal{M}$ is a Riemannian manifold,
  - $\log_{\mathbf{x}}(.)$ is a mapping from $\mathcal{M}$ to the tangential vector space $\mathcal{T}_{\mathbf{x}}$ in $\mathbf{x}$,
  - $\exp(.)$ is a mapping from $\mathcal{T}_{\mathbf{x}}$ to $\mathcal{M}$,
- Dynamical models and PDE:
  - $\mathbb{M}$ is a dynamical model,
  - $\mathbb{H}$ is an observation/constraint operator,
  - $\boldsymbol{\epsilon}$ is a noise/control term,
  - $\partial_{\mathbf{X}}\mathbb{M}$ tangential linear operator of $\mathbb{M}$ in $\mathbf{X}$ and
  - $(\partial_{\mathbf{X}}\mathbb{M})^*$ the associated adjoint operator
- operators and operations:
  - $||.||_{\Sigma^{-1}}$ is the induced norm of the inner product $< \Sigma^{-1}., . >$.
  - $\mathbf{M}^+$: pseudo-inverse of $\mathbf{M}$,

B

Application of Geodesic analysis in the context of Machine Learning

Most of the well known methods using the kernel trick [206, 207] postulate that since the data are embedded in a Kernel Reproducing Hilbert Space (RKHS) with high dimensionality, non-linear data description is likely to become linear. As such, most of the classical linear methods can be applied with benefits. However, in the RKHS associated to numerous kernels (including the Gaussian kernel, on which this work is focused), all vectors have a unitary norm: the dataset lies on a hypersphere [208]. Hence, should this particular geometry be explicitly exploited by using non linear statistical tools in the RKHS? This work is a step in this direction. We notably show on two different applications (classification and clustering) that this idea can yield enhanced results over some real world datasets. The key idea is to consider a geodesic distance on the hypersphere rather than the Euclidean one to perform the data analysis. The geodesic distance corresponds to the total length of the shortest path over the hypersphere between two points, and it can be computed readily using trigonometric operators (Figure 44). Interestingly enough, this leads us to the definition of a new kernel: It appears that the geodesic distances in the original RKHS are equivalent to the Euclidean distances in a new RKHS. Thus, when data are embedded in this latter, it is indeed really justified to use linear methods. Our construction can be related to the work of Lafferty and Lebanon [209], who define a family of kernels based on diffusion operators over a Riemannian manifold. In our case, the geometric structure of the manifold is directly used to give a closed-form kernel expression instead of using a Fischer information metric.

In the next Section B.1, we set notations, and we provide background materials on geodesic distances and Riemannian manifolds. In Section B.2 we adapt the classical tools of geodesic analysis to the Gaussian RKHS: To overcome the main drawback of kernelized space (the coordinates of the vectors are unknown), we find a transformation of the Gram matrix induced by the Gaussian kernel which takes into account geodesic distances.

B.1. Geodesic analysis on the hypersphere

This section introduces the basis of a geodesic analysis on the hypersphere in the RKHS induced by the Gaussian Kernel. After stating the problem, basic facts about Riemannian geometry are presented and the notion of geodesic analysis is introduced.

B.1.1. Problem Statement

Let $X = \{x_1, \ldots, x_p\}_{(x_i \in \mathbb{R}^n)}$ be a set of p separated training samples described with n variables, and living in a space isomorphic to $\mathbb{R}^n$ and referred to as the *input space*. It is endowed with the Euclidean inner product denoted $\langle ., . \rangle_{\mathbb{R}^n}$ in the following. Let $k(., .)$ be a symmetric form measuring the similarity among pairs of X, also called *kernel*. Let $\mathcal{H}$ be the associated RKHS, or

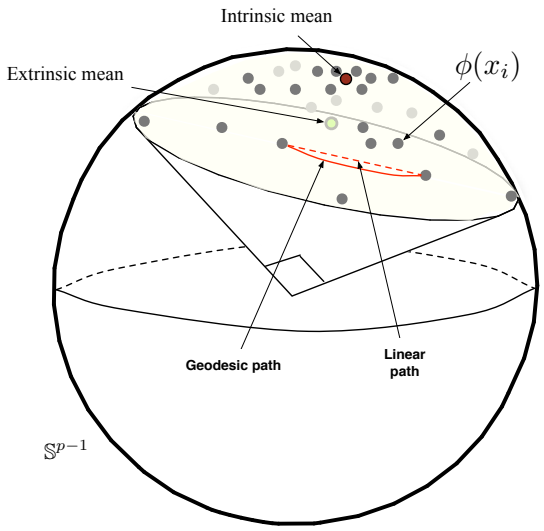


Figure 44.: Whatever the distribution of X, $\phi(X)$ lies within sphere quadrant. We propose to consider geodesic distance between elements of $\phi(X)$ rather than the Euclidean one. The Karcher (intrinsic) mean of $\phi(X)$ is represented as a red point, whereas the extrinsic mean is depicted in green. Note the latter is inside the hypersphere, whereas the Karcher mean lies on it.

feature space, also equipped with a dedicated inner product noted $\langle ., . \rangle_{\mathcal{H}}$, such that for any pair $(x_i, x_j) \in X^2$, we have:

$$\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} = k(x_i, x_j) \tag{B.1}$$

where $\phi(.)$ is an implicit mapping from $\mathbb{R}^n$ onto $\mathcal{H}$. We use the shorthand notation $\phi(X)$ for the set $\{\phi(x_1), \ldots, \phi(x_p)\}_{(\phi(x_i) \in \mathcal{H})}$. $\mathbf{K}$ is the Gram matrix of $\phi(X)$, and as such $\mathbf{K}_{ij} = k(x_i, x_j)$. We use the generic notation x for any vector of $\mathbb{R}^n$. Similarly, any vector of $\mathcal{H}$ is noted $\phi(x)$ (if its pre-image is assumed to be x) or simply y (if there is no assumption on its pre-image).

A kernel of particular interest in this work is the Gaussian kernel, defined as:

$$k(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{B.2}$$

with the variance parameter $\sigma^2 \in \mathbb{R}_+^*$. Remark that: (1) the norm of any $\phi(x_i) \in \mathcal{H}$ is the unity, *i.e.* $\langle \phi(x_i), \phi(x_i) \rangle_{\mathcal{H}} = 1$, (2) the Gaussian RKHS is of infinite dimension. As a consequence, whatever X, $\phi(X)$ spans a subspace of dimension exactly p, and as such $\phi(X)$ lies on the unit hypersphere $\mathbb{S}^{p-1} \subset \mathcal{H}$. Moreover, as the inner product of two unit vectors corresponds to the cosine of their angle, and as $\forall (x_i, x_j)$, $k(x_i, x_j) \in [0, 1]$, whatever X, $\phi(X)$ lies in a restriction $\mathcal{R}$ of $\mathbb{S}^{p-1}$ which is embedded in a sphere quadrant (its maximum angle is smaller than or equal to $\pi/2$, such as illustrated on Figure 44). Naturally, as $k(x_i, x_j)$ varies according to the value of the σ parameter, the surface of $\mathcal{R}$ varies accordingly: When σ increases, $k(x_i, x_j)$ increases, (*i.e.* the cosine between x_i and x_j increases), and thus the surface of $\mathcal{R}$ decreases. Conversely, when $\sigma \to 0$, $\mathcal{R}$ tends to a sphere quadrant.

B.1.2. Analysis on Riemannian manifolds

A Riemannian manifold $\mathcal{M}$ in a vector space $\mathcal{V}$ with inner product $\langle ., . \rangle_{\mathcal{V}}$ is a real differentiable manifold such that the tangent space $\mathcal{T}_{x^*}$ associated to each vector x^* is endowed with an inner

product $\langle .,. \rangle_{\mathcal{T}_{x^*}}$. In this work, $\langle .,. \rangle_{\mathcal{T}_{x^*}}$ reduces to $\langle .,. \rangle_{\mathcal{V}}$ on $\mathcal{T}_{x^*}$, so for simplicity we assimilate $\langle .,. \rangle_{\mathcal{T}_{x^*}}$ to $\langle .,. \rangle_{\mathcal{V}}$.

Classically data analysis is performed in $\mathcal{V} = \mathbb{R}^n$ and not in $\mathcal{M}$, as in the former it is rather natural to formalize the intuitive geometric notions (distance, mean, variance, direction, etc.) which are necessary to characterize the dataset. On the other hand, the statistical analysis of a dataset within $\mathcal{M}$ requires the non-trivial generalization of these notions to the setting of Riemmanian geometry. One of the first statistical analysis tool designed for Riemannian manifold is the Principal Geodesic Analysis (or PGA), the goal of which is to find a set of directions, called *geodesic directions* or *principal geodesics*, that best encode the statistical variability of the data. PGA was first introduced by Fletcher et al. [91], and received since then numerous addenda [210, 93], which are beyond the scope of this work. Here, we only focus on the tools of Riemannian geometry which are involved in the definition of PGA. The crucial observation of Fletcher is that a first order approximation of the distances among the samples of the dataset can be obtained if one projects the dataset in $\mathcal{T}_\mu$, the tangent space at μ, the Karcher mean of the dataset. We recall that the *Karcher mean* [211] $\mu \in \mathcal{M}$ differs from the traditional mean $\overline{x} \in \mathcal{V}$ (also called the *extrinsic mean*): It is the point of $\mathcal{M}$ which minimizes the sum of squared geodesic distances to every input data. As such, it constitutes an *intrinsic mean* (see Figure 44 for an illustration). We have:

$$\mu = \arg\min_{x \in \mathcal{M}} \sum_{i=1}^{p} d_{geod}(x_i, x)^2. \tag{B.3}$$

This approximation of the geodesic distances in $\mathcal{M}$ by the Euclidean distances in $\mathcal{T}_\mu$ seems particularly appealing, and it has been shown [94] that for a sphere the induced error is rather low. However, as this manifold lies in $\mathcal{V} = \mathcal{H}$ (instead of $\mathbb{R}^n$), the tractability of this approximation addresses several questions: First, how to define geodesic distances on the manifold embedding $\phi(X)$, and compute the associated Karcher mean μ of $\phi(X)$? Second, how to characterize $\mathcal{T}_\mu$ and project $\phi(X)$ onto $\mathcal{T}_\mu$? These two questions are addressed in two dedicated subsections of the next section.

B.2. Data Analysis over the hypersphere in the Gaussian RKHS

Let us consider the unit hypersphere $\mathbb{S}^{p-1} \in \mathcal{H}$, the surface of which is the Riemannian manifold which embeds $\phi(X)$.

B.2.1. Geodesic distance and Karcher mean

The Riemannian distance (or the geodesic distance) between $\phi(x_i)$ and $\phi(x_j)$ on $\mathbb{S}^{p-1}$ corresponds to the length of the portion of the great circle embedding $\phi(x_i)$ and $\phi(x_j)$. It is simply given by:

$$d_{geod}(\phi(x_i), \phi(x_j)) = \arccos(\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}). \tag{B.4}$$

Then Equation (B.3) reads:

$$\mu \quad = \quad \arg\min_{y \in \mathcal{H}} \sum_{i=1}^{p} \arccos(\langle \phi(x_i), y \rangle_{\mathcal{H}})^2. \tag{B.5}$$

The Karcher mean of X exists and is uniquely defined as long as X belongs to a Riemannian ball of radius $\pi/4$ [211, 212] which is the case since two points can be at maximum distant from $\pi/2$. Usually, non-linear optimization methods can be used to compute this mean. However, finding the coordinates for μ is impossible, since we do not have access to the coordinates of $\phi(X)$. Instead, we turn on the search of the pre-image $\tilde{x} \in \mathbb{R}^n$ of $\mu \in \mathcal{H}$ (such that $\mu = \phi(\tilde{x})$). It is the solution

of the following (non-linear) minimization problem:

$$\tilde{x} = \arg\min_{x\in\mathbb{R}^n} \sum_{i=1}^{p} \arccos(\langle\phi(x_i),\phi(x)\rangle_{\mathcal{H}})^2, \tag{B.6}$$

$$= \arg\min_{x\in\mathbb{R}^n} \sum_{i=1}^{p} \arccos(k(x_i,x))^2. \tag{B.7}$$

To operate this minimization, let us consider

$$\begin{aligned} f:\mathbb{R}^n &\to \mathbb{R} \\ x &\mapsto \sum_{i=1}^{p} \arccos(k(x_i,x))^2 \end{aligned}$$

and compute its gradient:

$$\nabla f(x) = \sum_{i=1}^{p} \frac{\partial}{\partial x} \arccos(k(x_i,x))^2, \tag{B.8}$$

$$= \frac{2}{\sigma^2} \sum_{i=1}^{p} \frac{\arccos(k(x_i,x))k(x_i,x)}{\sqrt{1-k(x_i,\ x)^2}}(x_i - x).$$

Setting this derivative to zero leads to a fixed point algorithm similar to the seminal work on pre-image computation proposed by Mika et al. [213]. This algorithm amounts to refining in several iterations a solution $\tilde{x}^t$ such that:

$$\tilde{x}^{t+1} = \frac{\sum_i \alpha_t(i)x_i}{\sum_i \alpha_t(i)} \text{ with } \alpha_t(i) = \frac{\arccos(k(x_i,x))k(x_i,\tilde{x}^t)}{\sqrt{1-k(x_i,\tilde{x}^t)^2}} \tag{B.9}$$

However, as stated in [213], this approach is prone to find local minima and its output is strongly dependent on the choice of the initial guess. Therefore, we propose a simple greedy algorithm (Alg. B.1), which simply consists in repeating p times the previous optimization by setting the initial guess as the different inputs x_i (this latter is then omitted in the sum of equation B.9). The estimation of the Karcher's mean pre-image is achieved using Algorithm B.1 with an $\mathcal{O}(k.n^2)$ complexity, where k is the number of iteration and n the number of samples. In practice k is small, namely less than 10 for the tested datasets when an RBF kernel is used. However, a possible drawback of this approach is that it only provides an approximation for the Karcher mean, since the true one may not have an exact pre-image in the input space. Thus, it may be interesting to consider other approaches to find the pre-image of the Karcher mean, e.g. distance based [214] or local isomorphism [215]. Nevertheless, their direct application is impossible since the Karcher mean is only defined through a minimization procedure without a closed-form solution. Fig. 45 illustrates the result of Alg. B.1 to compute the pre-image of the Karcher mean on two toy datasets (points randomly sampled over a square and a spiral in 2 dimensions).

B.2.2. Projection on the tangent space

In the particular case of hypersperical manifolds, the mapping of any point onto a tangent space (this mapping is usually referred to as the *logarithmic map*), and the reverse mapping (the *exponential map*) are easy to define: The logarithmic map at location μ which projects any point $\phi(x_i) \in \mathcal{R} \subset \mathbb{S}^{p-1}$ onto $\mathcal{T}_\mu$ has the following form:

$$\begin{aligned} \text{Log}_\mu : \mathcal{R}\setminus\mu &\to \mathcal{T}_\mu \\ y &\mapsto \frac{\theta}{\sin(\theta)}(y - \cos(\theta)\cdot\mu) \end{aligned} \tag{B.10}$$

Algorithm B.1: Pre-image of the Karcher mean on the sphere in the RKHS

$\epsilon \leftarrow$ small value, $\tilde{x} \leftarrow \text{mean}(X)$
for $i = 1$ **to** p **do**
  $x_i^{t=0} \leftarrow x_i$
  repeat
    update $\tilde{x}_i^{t+1}$ using equation B.9 with $\tilde{x}_i^t$
  until $||\tilde{x}_i^{t+1} - \tilde{x}_i^t||^2 < \epsilon$
  if $f(\tilde{x}_i^{t+1}) < f(\tilde{x})$ **then**
    $\tilde{x} \leftarrow \tilde{x}_i^{t+1}$
  end if
end for
Output $\tilde{x}$

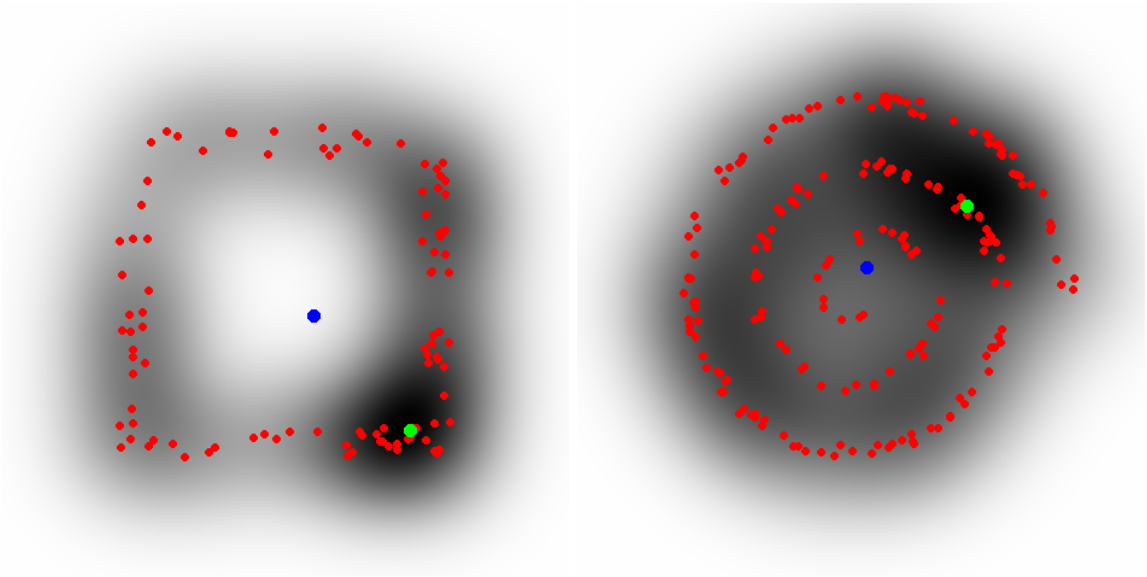


Figure 45.: Illustration of Karcher mean on two datasets: The dataset is represented by red points. The blue point is the data mean in input space, The green point is the pre-image of the Karcher mean after mapping onto the RKHS (the grayscale represents the function f values as described in Equation B.8).

where θ is the angle between μ and y *i.e.* $\theta = \arccos(\langle \mu, y \rangle_{\mathcal{H}})$. When $\theta = 0$, it is natural to consider that $y = \mu$. Conversely, the exponential map[1], which projects a vector y of $\mathcal{T}_\mu$ onto $\mathbb{S}^{p-1}$, is defined as:

$$\begin{aligned} \text{Exp}_\mu : \mathcal{T}_\mu &\rightarrow \mathbb{S}^{p-1} \\ y &\mapsto \frac{\sin(\theta)}{\theta} \cdot y + \cos(\theta) \cdot \mu \end{aligned} \tag{B.11}$$

where θ is given by $\theta = \arccos\left(\frac{\langle y, \mu \rangle}{||y||}\right) = ||y||$.

When using the kernel notation, and for $\phi(x_i) \neq \mu$ Equation B.10 reads:

$$\text{Log}_{\phi(\tilde{x})}(\phi(x_i)) = \frac{\arccos(k(x_i, \tilde{x}))}{\sqrt{1 - k(x_i, \tilde{x})^2}} (\phi(x_i) - k(x_i, \tilde{x})\phi(\tilde{x})). \tag{B.12}$$

[1]It is important to note that points on $\mathcal{R}$ are presented as vectors from the center of the hypersphere, while points on $\mathcal{T}_\mu$ are presented as vectors from μ.

So far, the exact computation of this projection cannot be conducted, as ϕ remains unknown. However, it is possible to derive the Gram matrix of $\mathrm{Log}_{\phi(\tilde{x})}(\phi(X))$:

$$\begin{aligned}\mathbf{K}^{\tilde{x}}_{ij} &= \langle \mathrm{Log}_{\phi(\tilde{x})}(\phi(x_i)), \mathrm{Log}_{\phi(\tilde{x})}(\phi(x_j))\rangle_{\mathcal{H}}, \\ &= \frac{\arccos(k(x_i,\tilde{x}))\arccos(k(x_j,\tilde{x}))}{\sqrt{1-k(x_i,\tilde{x})^2}\sqrt{1-k(x_j,\tilde{x})^2}}\cdot \\ &\quad (\phi(x_i)-k(x_i,\tilde{x})\phi(\tilde{x}))^T(\phi(x_j)-k(x_j,\tilde{x})\phi(\tilde{x})). \end{aligned} \tag{B.13}$$

With some simple calculations we finally obtain a simple form for the entries of $\mathbf{K}^{\tilde{x}}$:

$$\mathbf{K}^{\tilde{x}}_{ij} = \frac{\arccos(k(x_i,\tilde{x}))\arccos(k(x_j,\tilde{x}))}{\sqrt{1-k(x_i,\tilde{x})^2}\sqrt{1-k(x_j,\tilde{x})^2}}\cdot(k(x_i,x_j)-k(x_i,\tilde{x})k(x_j,\tilde{x})). \tag{B.14}$$

Finally, it is possible to consider the geodesic distances in $\mathcal{H}$, by simply replacing the Gram matrix $\mathbf{K}$ associated to the kernel $k(.,.)$ by another Gram matrix $\mathbf{K}^{\tilde{x}}$. it is also possible to interpret $\mathbf{K}^{\tilde{x}}$ directly as the Gram matrix derived from a new kernel $k^{\tilde{x}}$ []. This geometrical interpretation of the approach has been tested in a context of clustering and classification. We refer the reader to [] for experimental results associated to the technique.

Résumé

Dans le domaine infographie, l'animation par ordinateur désigne notre capacité à faire se mouvoir, via un ensemble de programmes ou d'algorithmes, des représentations géométriques virtuelles d'objets le plus souvent réels. Si les jeux vidéos ou le champ des effets spéciaux sont les premiers domaines d'application, le développement des technologies de l'information dans notre vie de tous les jours a ouvert la voie à un grand nombre d'autres utilisations. L'animation à proprement parler nécessite de spécifier le comportement dans l'espace et le temps de l'objet considéré. Il existe un grand nombre de méthodes permettant de réaliser cette spécification. Parmi celles-ci, la catégorie des méthodes dites "basées données" permettent d'obtenir un réalisme important en s'appuyant sur des exemples capturés du monde réel. A ces méthodes sont associés trois problèmes fondamentaux :

- **un problème d'acquisition** : quels sont les bons descripteurs du mouvement ? Comment, et via quel médium, les capter du monde réel ?
- **un problème de généralisation** : comment généraliser l'information obtenue à partir de quelques exemples ? Comment en caractériser leur portée ?
- **un problème de contrôle**: par quel biais introduire cette information dans une boucle d'animation ? Peut on la modifier pour l'adapter à des cadres différents de ceux de la captation ?

Pris ensemble, ces problèmes permettent d'établir des schémas d'Analyse/Synthèse, où l'information peut être capturée grâce à une formalisation (un modèle) numérique du phénomène considéré, puis utilisée à son tour dans la définition même du modèle. Ce document présente des travaux de recherche dans cette direction menés par moi et mon équipe depuis 2004. Deux domaines d'application "phare" sont considérés : l'animation de personnages virtuels, et plus particulièrement des personnages doués de la faculté de communiquer en langue des signes, et la simulation de foules de gens guidée par les données. Dans les deux cas est montré comment les données peuvent interagir avec les modèles numériques. Nous tentons même de mettre en avant l'émergence de cercles vertueux où l'on s'aperçoit que de meilleures données entrainent de meilleurs modèles, qui à leur tour autorisent la captation de nouvelles données et ainsi de suite. Ainsi cette thématique ouvre un large panel de problèmes évoluant de l'apprentissage à la simulation numérique.

Abstract

In the computer graphics domain the terms "Computer Animation" refer to the possibilities of animating with a set of programs and algorithms some virtual and mostly geometrical representations of objects. If computer games and production of visual effects in the cinematography industry are the most obvious applications, the development of 3D technologies in our day to day life through more powerful computing architectures and/or mobile devices has drastically augmented the number of possible use of those technologies. Animating objects requires to define the behavior of a virtual description of an object through space and time. Among the different possibilities, **Data-driven** methods usually provide a good way to obtain realism with only a few downfalls; the idea is to be able to use some a priori knowledge about the motion (e.g. kinematic trajectories) obtained from the observation of the corresponding phenomenon in the real world. However, three problems need to be addressed:

- **the acquisition problem**: what are the good descriptors for a motion ? Is it possible to capture them directly from the real world ?
- **the generalization problem**: how can one generalize the information contained in the observed examples ? How to characterize the extent of this generalization ?
- **the control problem**: how to use this knowledge into an animation method ? Is it possible to control in some ways the final animation ?

Altogether, those problems set up the basis of Analysis/Synthesis schemes that try to couple data and models. This document presents some contributions in these directions that were elaborated and developed during my previous years of research since 2004. Two main domains of applications are considered: virtual character animation, and more specifically the animation of virtual signer, and the definition of data-driven crowd simulation paradigms. In both cases, we show how the data can be used in conjunction with a numerical model of the considered phenomenon. Moreover, we highlight the potential virtuous circle that enables to acquire better data and define better models by capitalizing on previous knowledges. This opens a large variety of problems ranging from machine learning to numerical simulation.

n d'ordre : ???

Université de Bretagne Sud

Centre d'Enseignement et de Recherche Y. Coppens - rue Yves Mainguy - 56000 VANNES

Tél : + 33(0)2 97 01 70 70 Fax : + 33(0)2 97 01 70 70