Optimal crowd editing

Pierre Allain, Nicolas Courty, Thomas Corpetti

HAL Id: hal-00904008

https://hal.archives-ouvertes.fr/hal-00904008

Submitted on 13 Nov 2013

Optimal Crowd Editing

Pierre Allain[a,*], Nicolas Courty[b], Thomas Corpetti[c]

[a]*Inria, Rennes, France*
[b]*Université de Bretagne Sud, Vannes, France*
[c]*CNRS, Rennes, France*

Abstract

Simulating realistic crowd behaviors is a challenging problem in computer graphics. Yet, several satisfying simulation models exhibiting natural pedestrians or group emerging behaviors exist. Choosing among these model generally depends on the considered crowd density or the topology of the environment. Conversely, achieving a user-desired kinematic or dynamic pattern at a given instant of the simulation reveals to be much more tedious. In this paper, a novel generic control methodology is proposed to solve this crowd editing issue. Our method relies on an adjoint formulation of the underlying optimization procedure. It is independent to a certain extent of the choice of the simulation model, and is designed to handle several forms of constraints. A variety of examples attesting the benefits of our approach are proposed, along with quantitative performance measures.

Keywords: Crowd editing, crowd simulation, agent steering, control, adjoint methods, optimal control strategy

1. Introduction

Animating 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. Nevertheless, several models exist, that can either rely on per-individual strategies, usually

---

*pierre.allain.t@gmail.com, +33 299847123

grouped under the denomination of steering behaviors or microscopic models, or in contrast on a global definition of the crowd flow and properties: the macroscopic models. Assessing the quality of those models is usually difficult, since several criteria can be evaluated: computation performances, presence of emerging behaviors, individual trajectories respecting the least effort principles, etc. In the context of civil engineering, those models can provide a lot of information about potentially dangerous areas or bottlenecks where problems are likely to occur. The context of graphics is slightly different since animators usually have a precise idea of what they want to show, which may differ from what is actually given by the simulation models. A possibility is then to tweak the different parameters of the used model, in a trial-and-error fashion, until the simulation is acceptable. It is easy to understand that this strategy, depicted in Figure 1.a, is likely to fail for most simulation models regarding the complexity of the dynamics inherent to the crowd if the animator tries to obtain a desired kinematic or dynamic pattern at a given time in the simulation process. This calls for automatic procedures that can reach those constraints while preserving the specificities of the crowd models. This notion of controlling a crowd is investigated in this paper. In practice, because of the discrete nature of pedestrians (and thus the highly discontinuous nature of the solution space), the use of simple gradient based methods yields some difficulties related to collision handling between individuals. This issue has already been encountered in the control of multibody dynamics in <Twigg and James, 2007>, where the authors suggest to select a solution from a set of samples (computed on-line on a cluster of machines). In this paper, we propose a combination of a gradient based method along with an heuristic to explore several distinct paths in the solution space.

Contributions. In the presented article, the control of a crowd is seen as an editing process, in the sense that the trajectories produced by the simulation model are deformed to achieve users constraints (Figure 1.b) 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. Those two types of constraints are illustrated in the rest of the paper. Our optimization process uses recipes from optimal control of variational models by formulating the problem with the adjoint theory <Lions, 1971>. It is
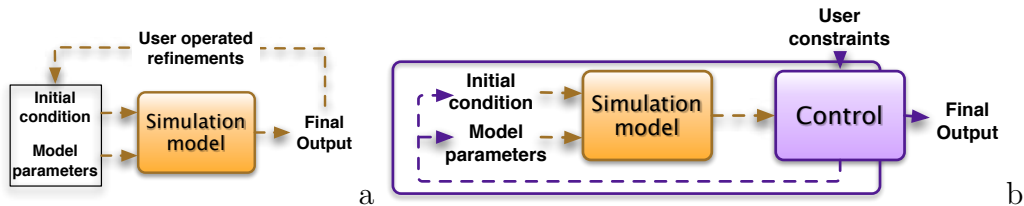
Figure 1: Differences between a user operated control of a crowd simulation model (a) and the control procedure (b) proposed in this paper.

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 model and on the nature of the constraint imposed by the animator. In that sense our method augments the latent qualities of a given crowd model but is not meant to produce systematically more realistic simulations than advanced techniques. In other words, the resulting scene after control using the model and the users constraints is optimal, but not the model which basically stays the same.

Outline of the paper. First, a presentation of the related work is performed in Section 2, and an overview of the control process is given in Section 3. The optimal control of dynamical system, along with its requirements are then presented (Section 4). Algorithmic aspects of this control will notably be explained in this part. In Section 5, a variety of control are applied on a generic dynamic crowd model to illustrate the power of our method and quantitative convergence results are also presented, before a discussion and a conclusion end the paper (Section 7).

2. Related Work

We first begin by giving the main approaches of crowd simulation, which we dissociate from the idea of crowd control. We also discuss other control works such as fluid ones.

2.1. 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 member of the crowds as agents with specific behaviors. Sophisticated behaviour models seek autonomous agents endowed with goals and specific attributes <Musse and Thalmann, 2001; Sung, Gleicher, and Chenney, 2004>, 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 <Reynolds, 1987>. 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 <Helbing, Farkas, and Vicsek, 2000>, reproducing experimental observations <Paris, Pettre, and Donikian, 2007>, principle of least effort <Guy, Chhugani, Curtis, Dubey, Lin, and Manocha, 2010> or vision based strategies <Ondřej, Pettré, Olivier, and Donikian, 2010>. The most recent methods allow to simulate large scale crowds at interactive framerates with convincing emergent behaviors of the groups. Another recent trend is to capture from the real world heterogeneous behaviours of pedestrians to add variety in the simulation and possibly realism. This is the case in <Lee, Choi, Hong, and Lee, 2007; Lerner, Chrysanthou, and Lischinski, 2007>, where authors capture individual trajectories of pedestrians and reuse them in an online fashion. 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 <Hughes, 2002; Treuille, Cooper, and Popovic, 2006; Pimenta, Michael, Mesquita, Pereira, and Kumar, 2008; Narain, Golas, Curtis, and Lin, 2009>. 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.

2.2. 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 <Ulicny, Ciechomski, and Thalmann, 2004> 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 <Chenney, 2004; Metoyer and Hodgins, 2004; Jin, Xu, Wang, Huang, and Zhang, 2008; Park, 2010; Patil, van den Berg, Curtis, Lin, and Manocha, 2011>. In <Jin et al., 2008>, Jin and colleagues define those fields as a combination of radial basis functions defined by the user. Park <Park, 2010> 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 <Courty and Corpetti, 2007>. 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 <Kwon, Lee, Lee, and Takahashi, 2008> or mesh deformation techniques <Takahashi, Yoshida, Kwon, Lee, Lee, and Shin, 2009; Henry, Shum, and Komura, 2012> to edit existing crowd animations.

In a sense, our method can be related to previous works on control techniques applied to fluid animation <Fattal and Lischinski, 2004; McNamara, Treuille, Popović, and Stam, 2004; Shi and Yu, 2005>, more specifically from the work of McNamara and his colleagues <McNamara et al., 2004> which use and adjoint method to compute the simulation gradients. In <Wojtan, Mucha, and Turk, 2006>, Wojtan and colleagues adapted the idea of the adjoint method to the control of particles, in the specific context of flocking applications using in practice linearized models associated with implicit schemes.

Our contribution mainly differs from these works in the fact that any user constraints, either defined at a particle level or on higher representation quantities (i.e. eulerian density, velocity, vorticity, etc.) can be given to the system whatever the controlled model, and automatically adapts the adjoint procedure, which tightly couples the crowd model and the constraints. Hence, it allows to augment existing simulation models with very various inputs, provided the model is differentiable by the state it drives as described in the following Sections.

3. Overview of the control policy

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.

Simulation. We refer to the dynamical model used for crowd simulation as a unified model $\mathbb{M}$ driving the state of the crowd $\mathbf{X} \in \mathbb{R}^{N_\mathbf{x}}$, which evolution is governed by the following partial differential equation:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = 0, \tag{1}$$

$$\text{with: } \mathbf{X}(t_0) = \mathbf{X}_0. \tag{2}$$

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. In this case, considering N pedestrians whose individual state is composed of q quantities (such as position and/or velocity), we have $N_\mathbf{X} = Nq$.

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 <Narain et al., 2009>), 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. These techniques are used to formalize the constraints in Section 4 when they are related to the environment.

Control. Control aims at automatically defining the best control parameters $U \in \mathbb{R}^{N_U}$ of a system. The optimality of these parameters depends on the difference between the state trajectory produced by the system and one or multiple external constraints $\mathbf{Y}$. This difference is evaluated thanks to a "sensor" which thanks to an *observation operator* is able to make the translation $\mathbf{Y} = \mathbb{H}(\mathbf{X})$. The resulting error is processed by a controller which will provide corrected system parameters (*i.e.* U) and, thanks to system's integration, a corrected trajectory $\mathbf{X}(t)$. Figure 2 shows an overview of the control policy.

This paper focuses on the controller which is indeed of great importance in this kind of approach. In our case it is an *optimal controller* based on *variational data assimilation using adjoint formulation.*
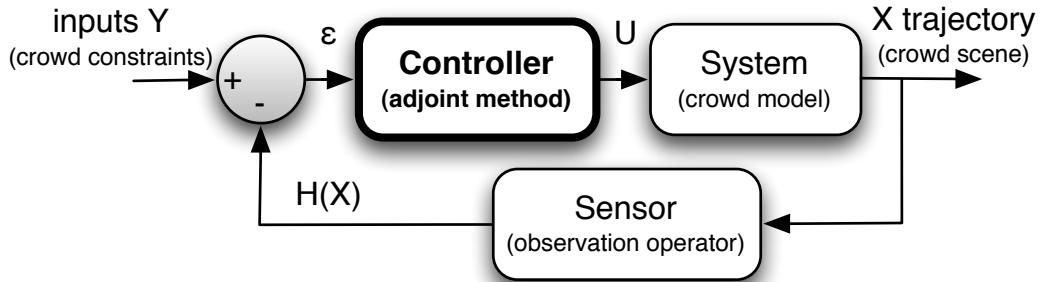

Figure 2: Functional scheme of the method. Each loop computes the difference between the input constraint $\mathbf{Y}$ and the observed state $\mathbb{H}(\mathbf{X})$, this difference supplies the assimilation process deriving from the model $\mathbb{M}$ which will give new states $\mathbf{X}(t)$.

Unlike lot of engineering control applications, the control loop is not expressed in the frequency domain, and $\mathbf{X}(t)$ 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.

4. Optimal Control

In this section, we will now explain how the controller is able to find optimal parameters for the crowd model $\mathbb{M}$.

4.1. Problem statement

The model of crowd can be defined as in Equation 1, which integration produces a state trajectory $\mathbf{X}(t)$. In order to give the model an opportunity to fulfill input constraints $\mathbf{Y}$, it is necessary to introduce a degree of freedom in the model. This can be accomplished by adding a term in charge of capturing the model deviation, leading to the new following equation:

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \boldsymbol{\epsilon}_{\mathbb{M}}(t), \tag{3}$$

7

with $\boldsymbol{\epsilon}_{\mathbb{M}} \in \mathbb{R}^{N\mathbf{x}}$ being called the model's error. One can notice this control parameter can be seen as a **time-evolving steering term** for crowd models.

The initial condition $\mathbf{X}(t_0)$ can also be considered as a degree of freedom, leading to the equation:

$$\mathbf{X}(t_0) = \mathbf{X}_0 + \boldsymbol{\epsilon}_0, \tag{4}$$

where $\boldsymbol{\epsilon}_0 \in \mathbb{R}^{N\mathbf{x}}$ stands for the *initial condition error*.

The input constraint $\mathbf{Y}$ is related to the system state by a non-linear function $\mathbb{H}$. The difference between $\mathbf{Y}$ and what is actually observed $\mathbb{H}(\mathbf{X})$ is the *observation error* $\boldsymbol{\epsilon}_{\mathbb{H}} \in \mathbb{R}^{N_{\mathbf{Y}}}$:

$$\mathbb{H}(\mathbf{X}) = \mathbf{Y} + \boldsymbol{\epsilon}_{\mathbb{H}}. \tag{5}$$

Equations 3, 4 and 5 form an *optimal system* (see <Lions, 1971>). The problem is then to find the state trajectory $\mathbf{X}(t)$ satisfying the system with the lowest errors possible, *i.e.* respecting as much as possible the model, the reference initialization condition and the input constraints. The controlled system, as presented in Figure 2, can be identified as Equations 3 and 4. The control parameters can therefore be defined as: $U = (\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$. It is then possible to express the problem as the minimization of the cost function:

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0) = \frac{1}{2} \int_{t_0}^{t_f} \|\boldsymbol{\epsilon}_{\mathbb{M}}\|_Q^2 \, dt + \frac{1}{2} \int_{t_0}^{t_f} \|\boldsymbol{\epsilon}_0\|_B^2 \, dt + \frac{1}{2} \int_{t_0}^{t_f} \|\mathbf{Y} - \mathbb{H}(\mathbf{X})\|_R^2 \, dt. \tag{6}$$

with Q, B and R being the respective covariance matrices of errors $\boldsymbol{\epsilon}_{\mathbb{M}}$, $\boldsymbol{\epsilon}_0$ and $\boldsymbol{\epsilon}_{\mathbb{H}}$. They can be seen as penalization factors of the associated errors. In case of diagonal matrices, they represent the weights given to the model, the initial condition and the input constraints. These covariances are crucial for the assimilation of observations. Once the model, the reference initial condition and the input constraints are set, they remain the only parameters to be configured by the user for the scene edition. For example, setting an unalterable crowd configuration at t_0 amounts to assign B a null value. More details about the definition of these matrices are given below.

Parameterization of the error covariance matrices. As stated above, these quantities act as weights on model, initialization and constraints.

Therefore the most decisive part of the parameterizing falls on the user since it is related to the balance one expects between model's consistency and constraints achievement. For example, a larger (resp. smaller) value of the error covariance matrix R will give less importance (resp. more importance) to the constraints during the control process. The same applies for covariances matrix Q related to the model and for B related to the initial condition. In practice, dimensional analysis can be used as a first guess on their value. For instance, knowing that model covariance unit dimension $[Q]$ equals $[\mathbf{X}]^2 . s^{-2}$ (where $[\bullet]$ is the dimension of the quantity $\bullet$), $[R]$ equals $[\mathbf{Y}]^2$ and $[B]$ is $[\mathbf{X}]^2$, it is possible to use the expected range of values of involved quantities and thus automatically compute covariances. Let us also precise that in some cases it is possible to dynamically tune these covariances. For instance, let us consider a model $\mathbb{M}$ being very reluctant to match a constraint $\mathbf{Y}$ that a user absolutely wants to be reached. Therefore, it is necessary to radically either increase Q or decrease R, yielding a possibly too fast gradient descent speed and consequently divergence. A solution to this issue can be to tune these covariances along gradient descent according to the error between $\mathbf{Y}$ and the corresponding observation $\mathbb{H}(\mathbf{X})$. For example, when error decreases, R will decrease too so the weight of $\mathbf{Y}$ increase and the gradient descent speed keeps in a convergence range. This kind of approach can be seen as an "at all costs" method since it ensures the constraint to be fulfilled. On the other hand, the model dynamics is progressively neglected along the control process, leading to an half-optimal solution in the sense that this strategy gives results essentially optimal for constraints.

Remark on constraints. The input constraints do not necessarily exist during the whole sequence. Inputs may be sparser than the time discretization. Instead of $\mathbf{Y}(t)$, we consider in this case the set:

$$(\mathbf{Y}_i, t_i, R_i), \tag{7}$$

meaning that input $\mathbf{Y}_i$ is set at time t_i with a covariance (*i.e.* weight) R_i.

4.2. Adjoint formulation

This part presents a method to minimize the cost function $\mathcal{J}$ and as such a way to estimate the optimal control parameters $(\boldsymbol{\epsilon}_{\mathbb{M}}, \boldsymbol{\epsilon}_0)$ required in Equations 3 and 4. Those parameters produce the optimal state trajectory. Minimizing the cost function in Equation 6 requires to cancel its derivatives

$\frac{\partial \mathcal{J}}{\partial \epsilon_\mathbb{M}} \delta \epsilon_\mathbb{M}$ and $\frac{\partial \mathcal{J}}{\partial \epsilon_0} \delta \epsilon_0$. In practice, due to the size of the system's state, a direct gradient descent is not possible since it would require the integration of the whole model for each gradient direction computation, each component of the state, and each time-step. The *adjoint formulation* allows to overcome this computational wall by expressing the gradient simply through one model integration, and its descent by one other. More details about adjoint techniques can be found in <Lions, 1971> and applications in computer graphics in <McNamara et al., 2004; Wojtan et al., 2006>.

4.2.1. Gradient computation

By introducing the adjoint variable $\boldsymbol{\lambda} \in \mathbb{R}^{N_\mathbf{X}}$, it is proved that the gradient is given by the adjoint trajectory $\boldsymbol{\lambda}(t)$ obtained by the backward integration of the following equation:

$$-\frac{\partial \boldsymbol{\lambda}}{\partial t} + (\partial_\mathbf{X} \mathbb{M})^* \boldsymbol{\lambda} = (\partial_\mathbf{X} \mathbb{H})^* R^{-1} (\mathbf{Y} - \mathbb{H}(\mathbf{X})), \tag{8}$$

$$\text{with: } \boldsymbol{\lambda}(t_f) = 0.$$

The derivation of the model and of the observation operator, respectively:

$$\partial_\mathbf{X} \mathbb{M} \in \mathbb{R}^{N_\mathbf{X}} \times \mathbb{R}^{N_\mathbf{X}}$$

$$\text{and } \partial_\mathbf{X} \mathbb{H} \in \mathbb{R}^{N_\mathbf{Y}} \times \mathbb{R}^{N_\mathbf{Y}},$$

both along with their adjoint[1] are needed to achieve this integration. This part constitutes the more tedious part of the method since such derivations can be quite tricky in some cases (but also very simple in others). However, these derivations have to be formulated and implemented only once for given crowd model and input constraint type (*i.e.* a given observation operator $\mathbb{H}$). We finally outline that there exist automatic differential tools (like TAPE-NADE <Hascoët and Pascual, 2012>) that enable to derive linear tangent and associated adjoint of complex models based on heuristics, conditionals, etc.

The adjoint variable trajectory $\boldsymbol{\lambda}(t)$ states for the discrepancy between the state trajectory $\mathbf{X}(t)$ and the input constraints $\mathbf{Y}$. Once available, one

---

[1] The adjoint operator is defined as $< Ax, y > = < x, (A)^* y >$ and is equivalent to a simple transposition in the case of matrices operator.

can simply write the variation of the control parameters:

$$d\boldsymbol{\epsilon}_{\mathbb{M}}(t) = Q\boldsymbol{\lambda}(t), \tag{9}$$

$$d\boldsymbol{\epsilon}_0 = \frac{1}{t_f - t_0} B\boldsymbol{\lambda}(t_0). \tag{10}$$

Let us now turn to how these variational parameters are used to converge to an optimal solution.

4.2.2. Gradient descent

The variational control parameters $(d\boldsymbol{\epsilon}_{\mathbb{M}}, d\boldsymbol{\epsilon}_0)$ can be used in two distinct ways, each one providing specific kind of solutions.

Local descent

In this case, $(d\boldsymbol{\epsilon}_{\mathbb{M}}, d\boldsymbol{\epsilon}_0)$ are directly used to compute a state trajectory offset $d\mathbf{X}(t)$. The equation driving this trajectory reads:

$$\frac{\partial d\mathbf{X}}{\partial t} + (\partial_{\mathbf{X}}\mathbb{M}) \, d\mathbf{X} = d\boldsymbol{\epsilon}_{\mathbb{M}}, \tag{11}$$

$$\text{with: } d\mathbf{X}(t_0) = d\boldsymbol{\epsilon}_0. \tag{12}$$

For each gradient descent iteration n, the state trajectory is updated by:

$$\mathbf{X}^{n+1}(t) = \mathbf{X}^n(t) + d\mathbf{X}(t). \tag{13}$$

This way of using control parameters can be seen as fully variational since only derived models are integrated during the control process.

Global descent

Here, the control parameters are first computed globally using their previous values. For each gradient descent n, they are updated such as:

$$\boldsymbol{\epsilon}_{\mathbb{M}}^{n+1}(t) = \boldsymbol{\epsilon}_{\mathbb{M}}^n(t) + d\boldsymbol{\epsilon}_{\mathbb{M}}(t), \tag{14}$$

$$\boldsymbol{\epsilon}_0^{n+1} = \boldsymbol{\epsilon}_0^n + d\boldsymbol{\epsilon}_0. \tag{15}$$

The state trajectory is then computed using Equations 3 and 4 of the optimal system. Therefore, this way of using control parameters can be seen as half-variational. This way of using the adjoint method is rather close to the functional scheme shown in Figure 2, whereas the local descent is rather based on a derived system. We will now see how these two ways behave.

4.3. Optimization strategy

The adjoint formulation is generally used in an Eulerian framework where the local and the global gradient descent provide quite coherent and rather similar results. For a Lagrangian framework as proposed in this paper, they can be very different. The reason comes from the obstacle and agents avoidance model. One can easily understand a crowd can exhibit very different trajectories for two different, even close, initial conditions. In some case for example, a pedestrian will pass left an obstacle, and for a slightly different initialization he will pass right. This will produce a trajectory gap, likely to generate other gaps, and so on. This point, we call *tangling issue*, turns out to be critical for the local gradient descent which is a fully variational method. This kind of issue is not new and can be solved using Monte-Carlo techniques, as for example in <Khan, Balch, and Dellaert, 2005; Smith, Gatica-Perez, and Odobez, 2005>, or using a selection on a set of pre-computed trajectories (<Twigg and James, 2007>). However, in our application both strategies are ineffective since the system state is too big and would require a number of pedestrians or pre-computed trajectories too large regarding to the actual computational capabilities.

Indeed, the time integration of the derived model of Equation 11 is not able to make a call for such radical decision. The trajectory offset $d\mathbf{X}(t)$ it produces can only account for trajectory variations, not trajectory *creation*. In an optimization point of view, the local gradient descent is hence likely to yield local minima for which pedestrians trajectories can look unnatural.

On the other hand, the global gradient descent calls on the "real" crowd model through Equation 3. During the optimization, trajectories are therefore allowed to jump from different solutions, abiding by the chaotic nature of crowd. However, convergence is not assured since between two iterations, part of the discrepancy carried by $\epsilon_{\mathbb{M}}(t)$ can become obsolete and produce irregularities.

A good solution can be found in mixing the both approaches, *i.e.* alternating the local descent for some iterations and the more global gradient descent. In our experimentations, this strategy prevented from falling too rapidly in local minima and yielded satisfying results and computational efficiency.

5. Crowd model

We recall here that the proposed control scheme is not dependent of the crowd model $\mathbb{M}$. However, $\mathbb{M}$ should be a dynamical model explicitly driving a state $\mathbf{X}$ through time evolution equations. As such, geometric approaches such as Reciprocal Velocity Obstacles, as defined in <van den Berg, Lin, and Manocha, 2008>, can not be controlled by our method, but the range of possibilities is nevertheless very wide. In this Section, the crowd simulation model against which we tested our optimal crowd editing strategy is presented. This model is a very generic dynamical model mainly based on Helbing's model <Helbing et al., 2000> in its use of pedestrian interaction forces. We note that in reference <Wojtan et al., 2006> the necessary derivations are also given for the Reynold's flocking model <Reynolds, 1987>.

5.1. Presentation

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 :

$$
\begin{cases}
\dfrac{\partial y_i}{\partial t} & = u_i, \tag{16} \\[2mm]
m_i \dfrac{\partial u_i}{\partial t} & = \mathbb{F}_i, \tag{17}
\end{cases}
$$

with y_i being the position of the pedestrian, u_i its velocity and m_i its mass that we value as $70\,kg$. 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 induced 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 <Helbing et al., 2000> by decomposing this force into four major components:

$$
\mathbb{F} = \underbrace{\mathbb{F}_{\text{will}}}_{\text{source}} + \underbrace{\mathbb{F}_{\text{fatigue}}}_{\text{friction}} + \underbrace{\mathbb{F}_{\text{sociological}}}_{\text{interactions}} + \mathbb{F}_{\text{obstacle}} \tag{18}
$$

At first, 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 W, and the level of determination as

α (valued $140\,N$ in experiments), leading to the source force: $\mathbb{F}_{\text{will},i} = \alpha_i W_i$, and in crowd space:

$$\mathbb{F}_{\text{will}} = \underline{\boldsymbol{\alpha}} \mathbf{W}. \tag{19}$$

In our experiments, $W(\mathbf{x})$ is computed on a grid using the Eikonal equation in order to simulate path-planning (like in <Treuille et al., 2006>), and we simply obtain $W_i = W(y_i)$.

But a pedestrian will be slowed 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:

$$\mathbb{F}_{\text{fatigue}} = -\underline{\mathbf{k}}\mathbf{u}, \tag{20}$$

where $\mathbf{k}$ are the the fatigue coefficients of pedestrians (valued $140\,kg.s^{-1}$ 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 $f_{ij}^{\mathbf{y}}$ relating pedestrians i and j therefore reads:

$$f_{ij}^{\mathbf{y}} = -a e^{-\frac{\|y_j - y_i\| - (r_i + r_j)}{b}} e_{ij}, \tag{21}$$

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 $f_{ij}^{\mathbf{y}}$, it is possible to express the sociological force in the crowd space writing:

$$\mathbb{F}_{\text{sociological}} = \boldsymbol{A}_{f^{\mathbf{y}}} \mathbf{1}, \tag{22}$$

with $\mathbf{1} = [1, 1, ..., 1, 1]^T$. In our experiments, the connectivity of $\boldsymbol{A}$ is set such that pedestrians distant of more than $3\,m$ 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 paper 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 $\mathbf{x}_{obs}$, and its intensity is given by a function of the distance from the i-*th* pedestrian position to this point. This distance and this direction read:

$$D_{\text{obs},i} = \|\mathbf{x}_{\text{obs}} - y_i\|. \tag{23}$$

$$e_{\text{obs},i} = \frac{\mathbf{x}_{\text{obs}} - y_i}{\|\mathbf{x}_{\text{obs}} - y_i\|} \tag{24}$$

We obtain for each pedestrian i the obstacle force:

$$w^{\mathbf{y}}_{\text{obs},i} = -ae^{-\frac{D_{\text{obs},i} - r_i}{b}} e_{\text{obs},i}, \tag{25}$$

leading to the expression: $\mathbb{F}_{\text{obstacle}} = \mathbf{w}^{\mathbf{y}}_{\text{obs}}$. Here again, $D_{\text{obs}}(\mathbf{x})$ and $e_{\text{obs}}(\mathbf{x})$ are computed on a grid using the Eikonal equation.

We now dispose of a complete differentiable dynamical model stating the evolution of

$$\mathbf{X} = \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix}, \text{ by the model:}$$

$$\mathbb{M}(\mathbf{X}) = \begin{bmatrix} -\mathbf{u} \\ -\underline{\mathbf{m}}^{-1}\mathbb{F}(\mathbf{y}, \mathbf{u}) \end{bmatrix}. \tag{26}$$

5.2. Derivation

As stated in Equation 8, the control process requires the adjoint derivation of the presented model which reads:

$$(\partial_{\mathbf{X}}\mathbb{M})^* = \begin{bmatrix} 0 & \partial_{\mathbf{y}}\mathbb{F}^T \\ 1 & \partial_{\mathbf{u}}\mathbb{F}^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -\underline{\mathbf{m}}^{-1} \end{bmatrix}. \tag{27}$$

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

$$\partial_{\mathbf{y}}\mathbb{F}^T = (\underline{\boldsymbol{\alpha}}\underline{J_W})^T + \boldsymbol{L}_{f\partial\mathbf{y}} + (\underline{\mathbf{w}}^{\partial\mathbf{y}}_{\text{obs}})^T, \tag{28}$$

$$\partial_{\mathbf{u}}\mathbb{F}^T = -\underline{\mathbf{k}}, \tag{29}$$

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

6. Results

We illustrate the control concepts with several constraints, either expressed at pedestrian or continuum levels. All the computations were done on a standard laptop. The optimization time is linear in terms of number of individuals and time steps in the initial sequence. For every experiments, the computation time never exceeded half an hour for a non-optimized version of the code. We note that the user can have access to and examine the different trajectories from the early stages of the optimization process, and since the convergence has roughly an exponential decrease behavior, a good hint of the final result can be obtained only after a few minutes of convergence.

In all the following experiments, we set the model covariance (*i.e.* the control steering weighting) as:

$$Q = \begin{bmatrix} Q_{\mathbf{y}} & 0 \\ 0 & Q_{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & I_{Nq} 100\, m^2.s^{-4} \end{bmatrix}, \tag{30}$$

with I_{Nq} being the identity matrix of size Nq. This covariance means the positions evolution of the model is fully trusted, but the velocities evolution is more uncertain. However, it is possible to assign $Q_{\mathbf{y}}$ a value, which would allow the control to slightly modify the pedestrian positions without respect to the dynamics of the system.

We also chose to ignore here corrections on the initial condition $\mathbf{X}(t_0)$ and therefore rather focus on the behavior of the control during the sequences. Consequently, we set B as null.

All time integrations were performed using the fourth-order Runge-Kutta method.

6.1. Per-pedestrians constraints
"Perfect" rendez-vous

The first experiment aims at testing the procedure by constraining the positions y_1 and y_2 of only two pedestrians in the crowd, yielding a simple observation operator and adjoint derivation:

$$\mathbf{Y} = \mathbf{y_Y}, \ \mathbb{H}(\mathbf{X}) = \mathbf{y}, \ (\partial_{\mathbf{X}} \mathbb{H})^* = \begin{bmatrix} I_{Nq} \\ 0 \end{bmatrix}. \tag{31}$$

The constraint matrix covariance is zero everywhere except for pedestrians 1 and 2:

$$R_{i,j} = \begin{cases} 0 & \text{if } i \neq j, \\ 10\, m^2 & \text{if } i = j = 1 \text{ or } i = j = 2 \\ \infty & \text{otherwise.} \end{cases} \tag{32}$$
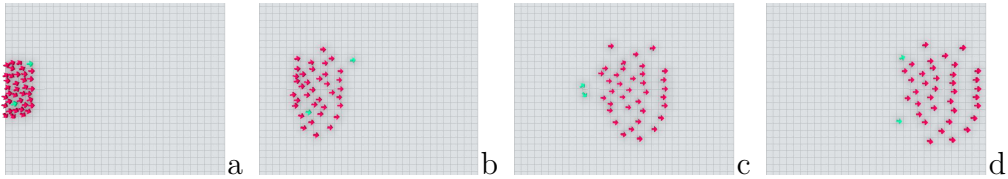
Figure 3: **Rendez-vous experiment.** In these illustrations, pedestrians are represented by arrows to clearly distinguish the two involved pedestrians . The meeting point is located behind the group, and should be reached at $t = 20s$. (a) $t = 0s$ (b) $t = 10s$ (c) $t = 20s$ (d) $t = 30s$.

From a starting group of 38 members, two people are given a rendez-vous constraint at a given time $t_{\mathbf{Y}} = 20\,s$ and in a specific position (illustrated in Figure 3). Hence, the editing process has to find a path for both individuals through the group to fulfill the constraint. In this experiment we use the local gradient descent. The optimization is processed in the interval $[t_0, t_f]$. 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 3 and in the accompanying video. After the rendez-vous, the two pedestrians tend to return to the place they occupied in the crowd before the optimization process. This is an effect of the local gradient descent since the derivation of repulsive interactions between agents (and with obstacles) acts as a spring. Therefore, for such model, one can state:

$$d\mathbf{X}(t) \underset{t \to \infty}{=} 0. \tag{33}$$

In the optimized scene we present, the controlled pedestrians don't reach their original place essentially because of the interactions cut-off in adjacency matrix $\boldsymbol{A}$. This yields rather a plastic deformation of the resulting trajectory $\mathbf{X}(t)$ instead of a theoretically elastic one. Let us mention that in a case where the user would prefer the pedestrians to ignore their original state, he can simply perform the optimization in the interval $[t_0, t_{\mathbf{Y}}]$ and use the natural model of the crowd after $t_{\mathbf{Y}}$.

Of course, it is important to mention that obtaining such a scenario only by simulating a crowd model would be very difficult to obtain and would require 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.

Letter constraints

The second experiment considers two groups of people, each one evolving in opposite directions. We aim at making each group to form, after crossing each others, the letters "SG" (for *Social Groups*). Here again the observation operator is set as in Equation 31, and the constraint covariance is set on the basis of Equation 32.

As can be observed in Figure 4(a), the simulation without any control creates two homogeneous flows without any specific pattern. In this experiment, the major difficulty comes from the tangling 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 4.3, is in practice handled using the global gradient descent.

The complete experimentation as well as comparisons with a simple simulation of the crowd model aiming at forming the required configuration are visible in the accompanying video. In figure 4(a), we illustrate the trajectories of the two groups without any control whereas figure 4(b) presents trajectories after 200 iterations of the editing process, highlighting the benefit of the control. As shown in the videos, the remaining trajectories are consistent and yield a more natural evolution than using a simple crowd simulation to reach this goal without control. To assess some quantitative values, the figure 4(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 in part due to the chaotic behavior of repulsive Lagrangian models that are likely to generate different scenario for two closed configurations, or in other words, to the tangling issue.

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 <Takahashi et al., 2009>, our method produces trajectories which match as much as possible the dynamics induced by the original crowd model, which is not the case in most geometric approaches.
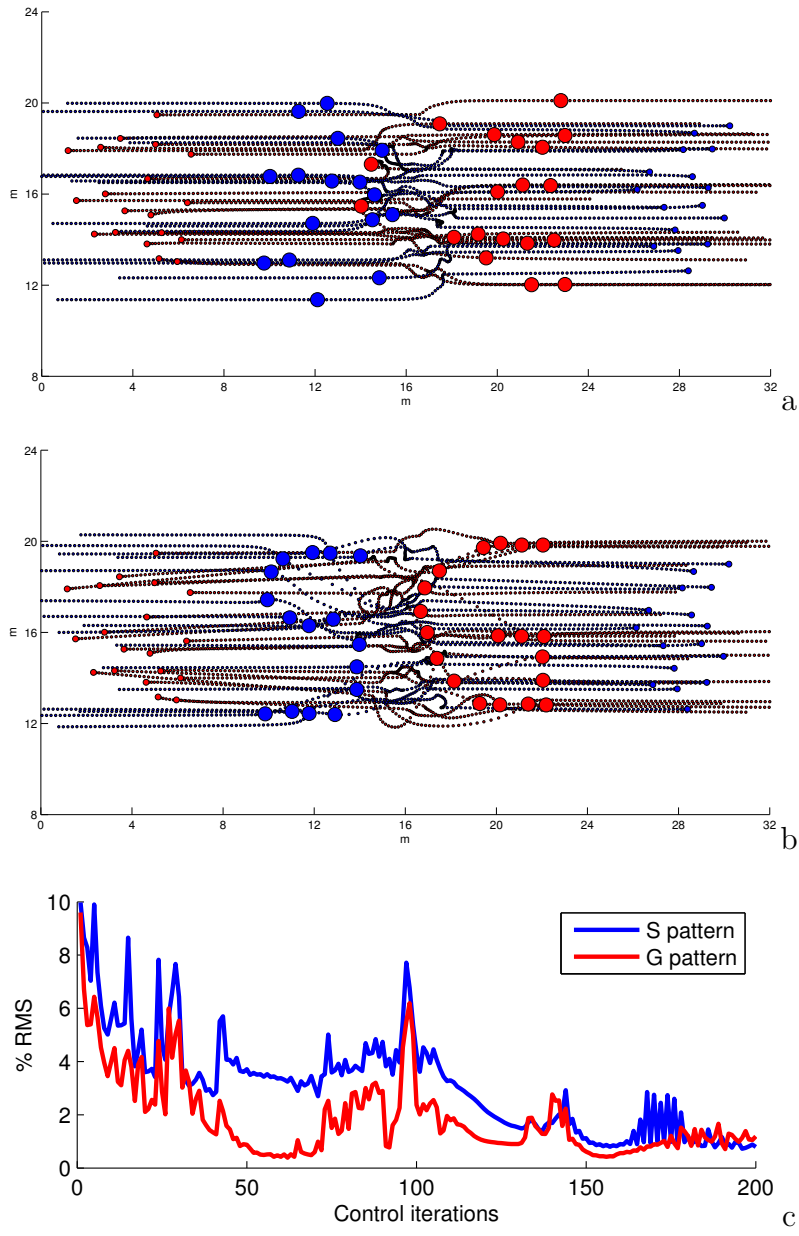
18

Figure 4: **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 user-desired positions.

19

6.2. Continuum constraints

This time, the constraint lies in a different space than $\mathbf{X}$. Indeed, $\mathbf{Y}$ refers here to Eulerian data expressed on a grid. For the different constraints to come in this section, the observation operators and their associated derivation can be found in <Allain, Courty, and Corpetti, 2012>.

Motion transfer from a video

Here, we follow the idea proposed in <Courty and Corpetti, 2007; Patil et al., 2011>, which consists in capturing a velocity field from a video, and then use it as a constraint to modify the global crowd motion. From user point of view, the input is therefore a video sequence, and from control point of view, the constraint is the velocity field extracted from the video. Such constraint allows the user to enrich a crowd scene with real crowd footage, or with original video content as we will show. We refer to this idea as *video-based motion transfer* for crowd. The experiment is illustrated in Figure 11. From an abstract video of moving shapes (courtesy of BBC motion video), a dense velocity field is extracted with a Lucas-Kanade filter <Lucas, 1984>. This time-varying velocity field then serves as a time-varying constraint $\mathbf{Y}(t)$ in the proposed control process. For this experiment we use the local gradient descent since no crowd tangling is expected. The covariance associated to the flow constraint is set as $R = I_{2M}0.05\, m^2.s^{-2}$, with M being the number of discrete meshes of the grid on which the flow is expressed. Extracts of the results are presented in Figure 11 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 root mean square (RMS) between the constraint velocity fields and the observed ones at each time step. Results are presented in Figure 5. The RMS of the output of the simple simulation and of the optimized scene, with respect to $\mathbf{Y}$, are presented. For comparison's sake, the RMS was also computed for the simple advection method proposed in <Courty and Corpetti, 2007>. In addition, we present the same error for the simple simulation of a mixture of both the dynamic crowd model and the advection method. In this purpose the user-desired velocity field $\mathbf{U}$ was simply added to the velocity term u_i in Equation 16. This way of integrating the motion field in the evolution equation is classical for most flow field based approaches (see for instance <Jin et al., 2008> or <Patil et al., 2011>). From all the four cases, our method achieves the best RMS performance. Let us note that the
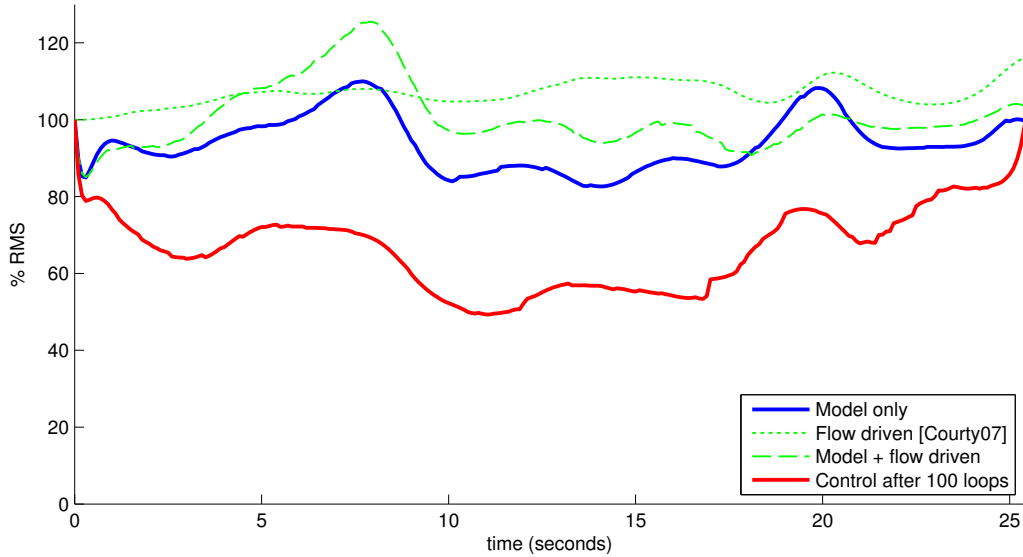
Figure 5: **Comparisons between methods for the video-based motion transfer.** Evolution of the normalized RMS along the animation.

RMS is computed over the entire Eulerian domain (a 32×16 grid) and that pedestrians do not cover the entire domain. Therefore, some error will always reside in the area of the domain empty of agents. Even so, our method manage to reach an average of 60% error in the main crowd stream time interval, when the other methods shows an average of 90-100%. In all cases, error is valued 100% at t_0 because the crowd doesn't move at this time. This can be corrected in the proposed method by simply attributing a value to the initialization covariance B. At t_f the error also reach such value, but this time because the crowd has left the domain.

Here, and contrary to <Courty and Corpetti, 2007>, it is important to note that the produced pedestrian trajectories match to a certain extent the dynamics of the original model (for instance, no collisions between individuals). This was not the case in the mentioned work, 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 6 for a schematic of the scene).

21

This kind of situation is very frequently described in other works as an interesting configuration for the examination of emerging behaviors, see for instance <Treuille et al., 2006>, <Ondřej et al., 2010> or <Bicho, Rodrigues, Musse, Jung, Paravisi, and Magalhães, 2012>. Our aim is to show that it is possible to change the magnitude of the bottleneck by imposing a patch constraint (*i.e.* in a specific part of the environment) on the 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.
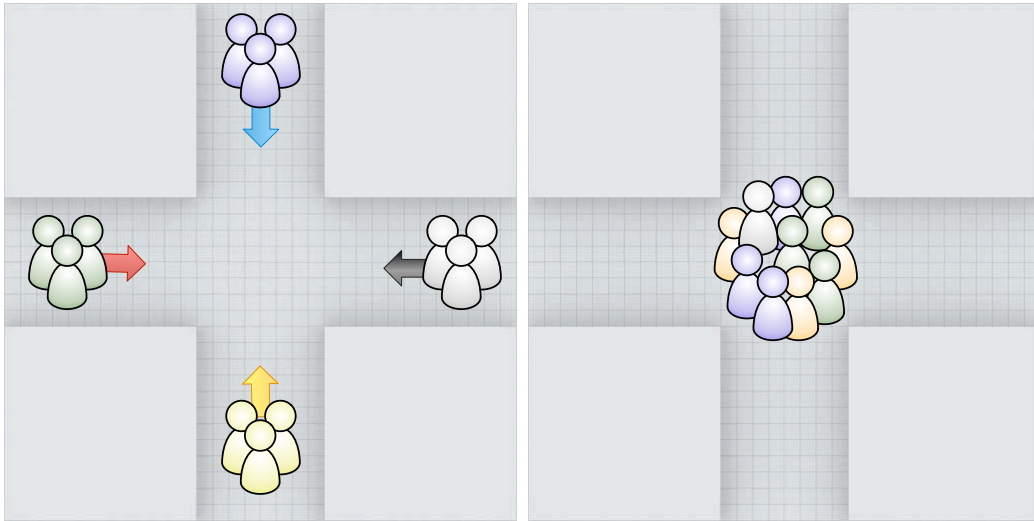

Figure 6: In this example, four groups of 38 people are trying to reach the opposite side, creating a congestion in the crossing zone.

The vorticity observation operator $\mathbb{H}(\mathbf{X})$ is obtained by the Eulerian cross-product differentiation of the Lagrangian velocities. This operator is thus highly non-linear since projections between these two spaces highly involves positions. It's expression and linearization is given in <Allain et al., 2012>. Hence, we impose a vorticity constraint for a duration of five seconds between the 12th and the 17th second of the animation (Figure 7.b), with a covariance set as $R = I_M 0.2\,s^{-2}$.

The original observed vorticity (Figure 7.a) of the output of the simple crowd simulation 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 its goal.

After optimization, one can observe the vorticity shape obtained (Figure 7.c) matches much better the constraint. The resulting paths can be seen in the
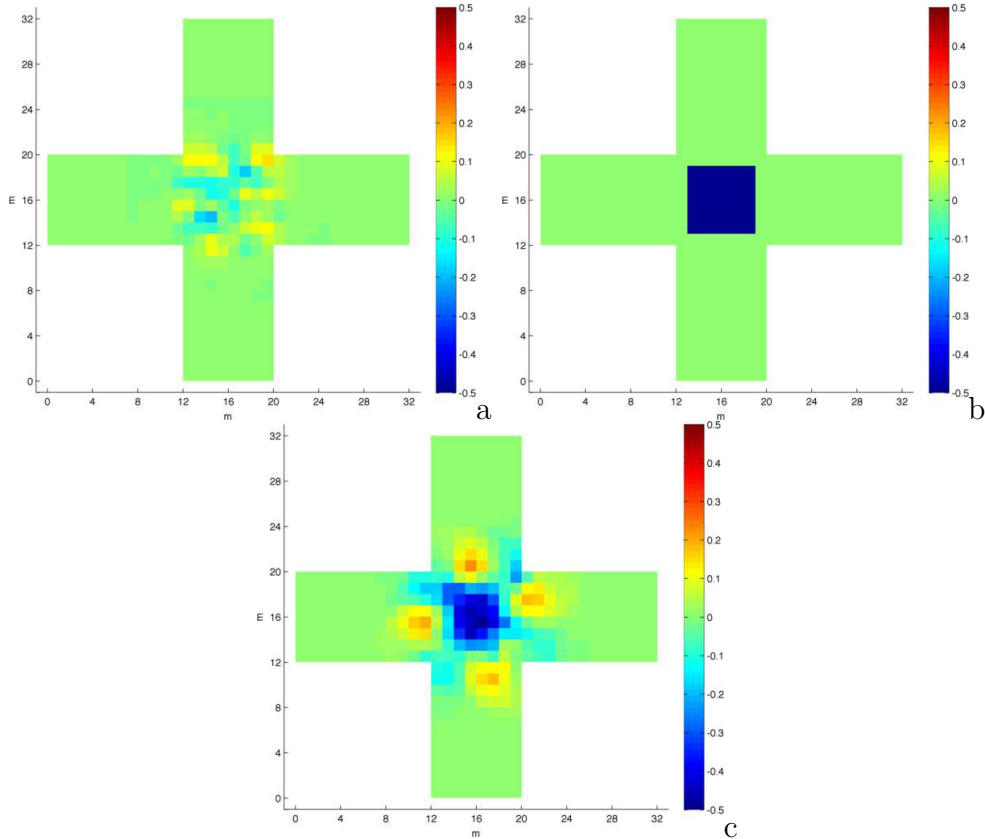

Figure 7: Vorticity in the scene at time $t_{\mathbf{Y}} = 16.5\,s$. The grid dimension is 32×32. (a) Observed vorticity of the output of the crowd simulation, (b) user constraint patch (homogeneous positive vorticity in the crossing zone), (c) output of the crowd optimization.

accompanying videos and in Figure 8. It is noticeable that at the beginning of the sequence, each group has chosen a side and that at the instant of the constraint a whirling pattern has emerged as a result of the control process.

Also, one can notice in Figure 9 that the average time is mostly decreasing along the control iterations, which tends to prove that imposing such a vorticity constraint can help to loosen the bottleneck. Among others, this procedure can also be seen as a tool to produce a user-desired emerging behavior even if the crowd model does not create it implicitly.

Of course, more advanced collision avoidance models such as in <van den

23

Berg et al., 2008> produce better results from realism point of view. But let us point out that none of them is able produce such specific behavior at a specific time. Also, we don't aim at improve the model used in this paper, but only to show that high-level constraints can produce similar conduct to advanced models, even in complex situations. Therefore, it is theoretically possible to edit a crowd scene so as to inject any crowd meta-behavior (provided its associated observation operator $\mathbb{H}(\mathbf{X})$ can be defined) such as avoidance vortices.

Density experiment

We propose here to use eulerian density maps as constraints. The crowd is asked to be present in specific areas defined by a user at given times. The density observation operator $\mathbb{H}(\mathbf{X})$ is obtained by the Eulerian convolution of Lagrangian positions of pedestrians. Its expression and linearization is given in <Allain et al., 2012>.

For this experiment, in order to introduce some heterogeneity in the group of pedestrians, we slightly modify the model by randomly setting the radius parameter in the range $[0.25\,m, 0.35\,m]$ for each pedestrian p_i where their mass m_i are proportional to their radius with a mean value of $70\,kg$ for $r_i = 0.3\,m$. In addition, the will parameter α_i is randomly set in the range $140\,N \pm 10\%$ and the fatigue coefficient k_i is set in the range $140\,kg.s^{-1} \pm 10\%$. These modifications provide variability in the simulated crowd dynamics in order to show the ability of our method to deal with heterogeneous crowds.

The original scene presents a crowd of 64 pedestrians heading a common direction after staying still at $t = t_0$ in an empty environment. Three density areas are set as constraints at $t_{\mathbf{Y}_1} = 7\,s$, $t_{\mathbf{Y}_2} = 14.5\,s$ and $t_{\mathbf{Y}_3} = 22\,s$ each lasting for $1\,s$ around their reference time, and such as their location is apart from the original path of the pedestrians (see Figure 10). The crowd needs then to radically modify its global behavior coming from the simulation model. Inputs $\mathbf{Y}_1$, $\mathbf{Y}_2$ and $\mathbf{Y}_3$ are $8\,m \times 8\,m$ areas filled by a homogeneous density values of $1.1\,ped.m^{-2}$ in order to slitghly over-constrain the inputs.

Even if no major trajectories tangling is expected in this scene, we use the global gradient descent strategy in this experiment in order to authorize reconfiguration among pedestrians during the control process. Such reconfiguration is likely to happen because of the introduced variability in the model for this experiment and because of the unnatural constraints as compared to original crowd trajectory. However, a user might prefer to preserve the original pedestrians organization. This is why we also present in the accompanying video the results obtained for this experiments using the local
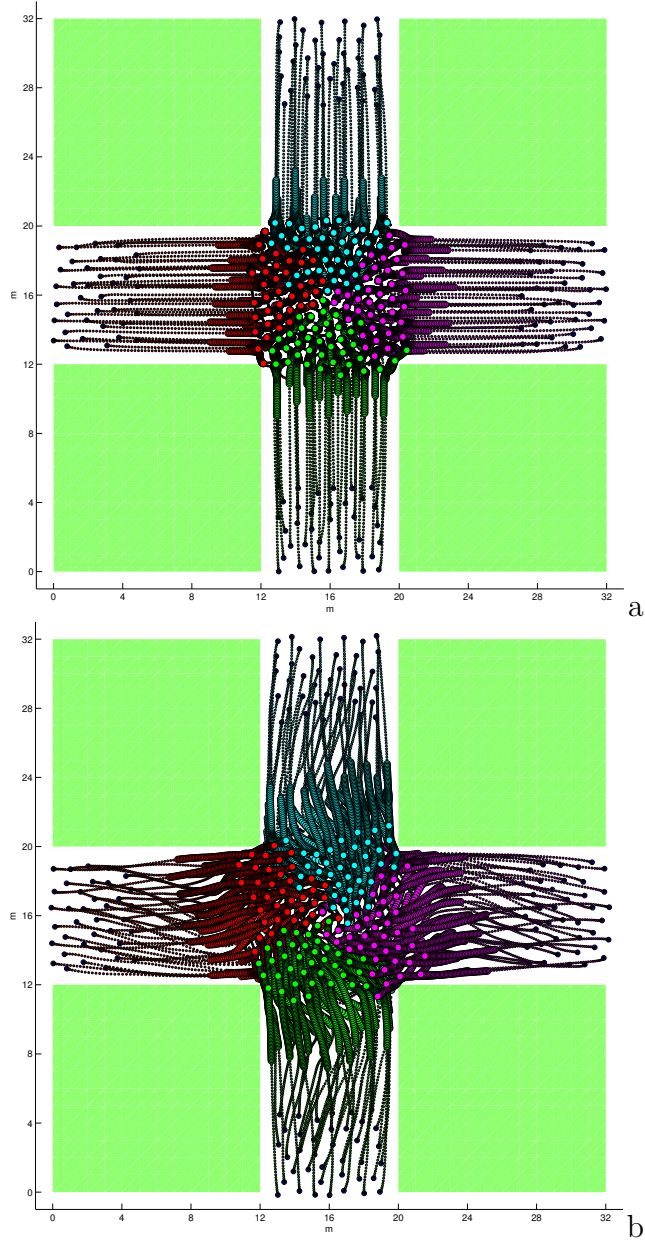
24

Figure 8: Trajectories of pedestrians for a duration of 20 seconds. The position of the individuals between the $t = 12\,s$ and $t = 17\,s$ are highlighted: (a) output of the crowd simulation, (b) output of the crowd optimization.
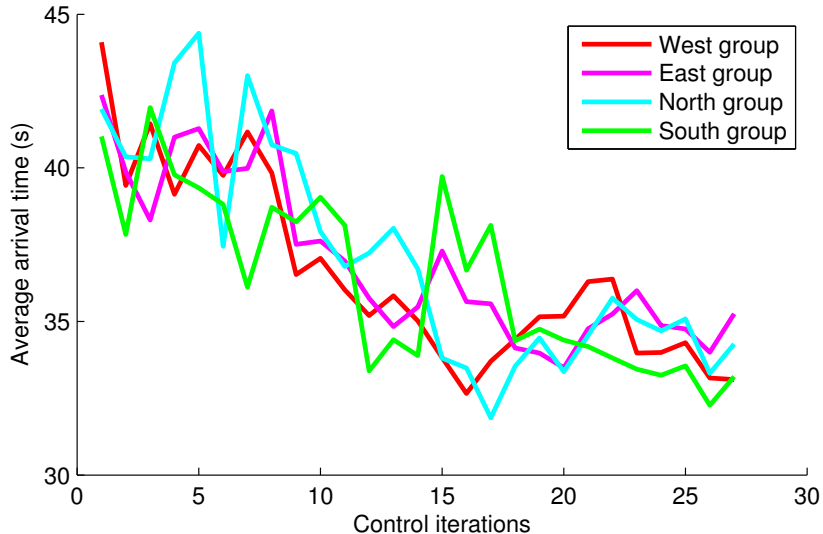
Figure 9: Average arrival time for each group plotted with respect to control iterations.

gradient descent. Covariances are parametered such as: $Q_{\mathbf{u}} = I_{Nq}0.1\,m^2.s^{-4}$ and $R = I_M 10\,ped^2.m^{-4}$, causing a slow but smooth convergence for demonstration's benefit.

In Figure 10(a), we present the output of the simulated model. One can see clearly the input densities do not correspond to the original pedestrians trajectories. Despite that, the control process is able to modify the trajectories along iterations so as to fit the inputs as we can see in Figures 10(b) and 10(c). The final edited scene can be viewed in Figure 13. In the image sequence, one can see that in the final output the crowd matches the three density inputs and that its inner dynamics is preserved all along the scene. This experiment gives another illustration of the large variety of situations the strategy presented in this paper can manage.

7. Conclusion and discussion

In this paper, the question of controlling a crowd simulation in order to fit some users constraints has been examined. We have proposed an original editing process which relies on a formulation of a gradient based approach in a variational setting expressed with the adjoint theory. This control methodology allows notably to control different aspects of the simulation, such as per-pedestrians properties, inter-pedestrians relations or as well continuum
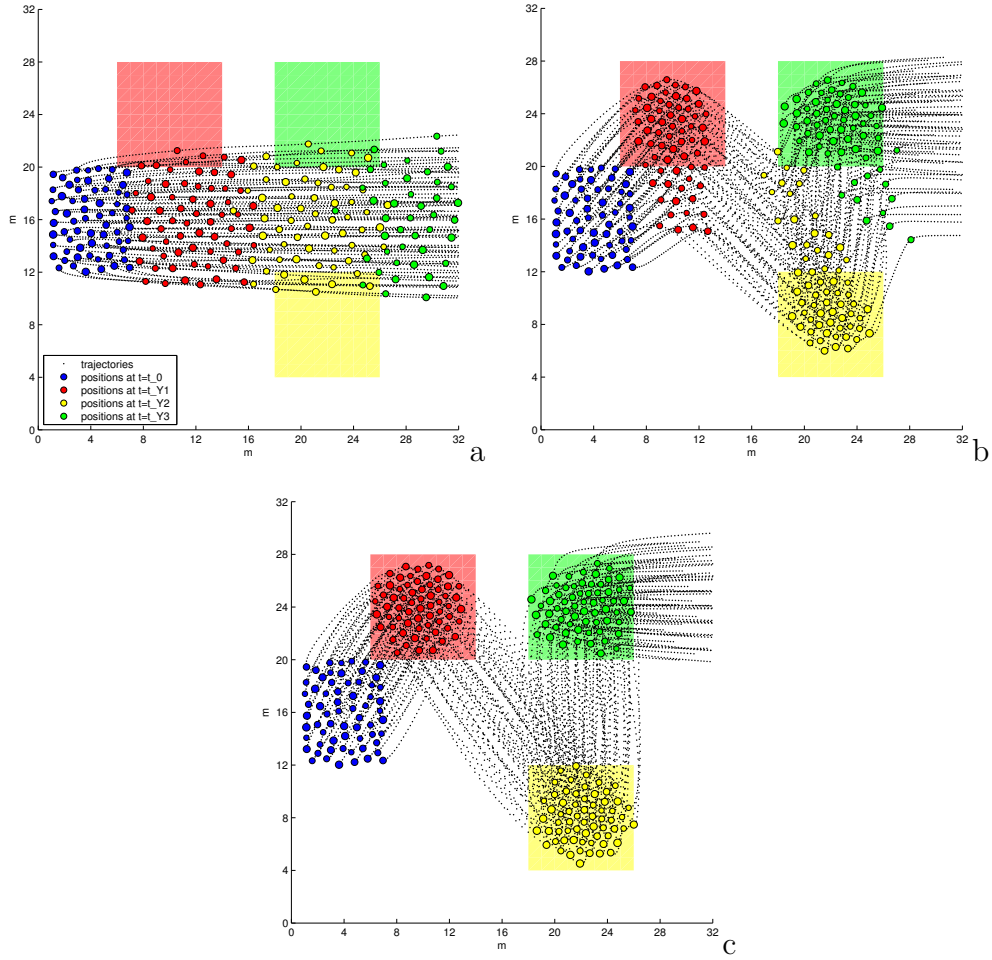
Figure 10: **Density experiment.** A group of pedestrians is heading right after staying still on the left. Figure (a) represents the trajectories at regular time steps obtained after simulation of the model with the input density constraints at $t_{\mathbf{Y}_1} = 7\,s$ (red), $t_{\mathbf{Y}_2} = 14.5\,s$ (yellow) and $t_{\mathbf{Y}_3} = 22\,s$ (green) all set as $1.1\,ped.m^{-2}$. In Figure (b), the trajectories after 50 iterations of the control process are presented. At this stage the model is clearly "bending" to reach the input constraints. In Figure (c), after 200 iterations, the simulation matches the three density inputs.

27

related information. Several examples of this control were given, illustrating the versatile nature of the control methodology. This framework adapts to any dynamical model, provided its analytical derivation, with respect to the state space, can be computed. Regarding the control nature, any type of constraints can be used if an analytic relation with the state space exists and can be differentiated. The output of this procedure is constituted by new pedestrians trajectories that realize a good tradeoff between what is imposed by the dynamical model and the realization of the constraint.

This method is mainly dedicated to users (as for example artists) which would desire to correct the output of a crowd simulation by imposing specific behaviors. In such production pipeline, the user can choose several types of constraints that would best match its need, and also the trade-off between the respect of the considered crowd model and his constraints (expressed through the covariance matrices).

It is important to outline that our technique however may suffer from some defects: first, it can not be used in interactive-time applications, since the optimization is performed globally, *i.e.* over the entire sequence, rather than sequentially, as would a short-horizon filtering. Yet, sequential version of this optimization, like in sequential Monte-Carlo or ensemble methods, could be a natural variant of our variational framework. Second, as far as the method is roughly speaking a gradient descent, there is no theoretical guarantee that a global minimum will be found. In the case where the problem is over constrained (such as in the letter formation example), the resolution may be stuck in unsatisfying situation, typical of local minima. The observation operators should be designed accordingly, and be as well-posed as possible, but this is a classical issue in optimization. Finally, the complexity inherent to the combinatorial explosion of inter pedestrians relative positioning (the tangling issue) can render the optimization process tedious, and calls for a careful choice of the covariance matrices, which can be sometimes tricky. Our future works will consider more advanced sampling strategies to cope with the problem of the avoidance of local minima, and lower the computation cost.

References

Allain, P., Courty, N., Corpetti, T., 2012. Particle swarm control. Research Report 1997, IRISA.

Bicho, A. D. L., Rodrigues, R. A., Musse, S. R., Jung, C. R., Paravisi, M., Magalhães, L. P., 2012. Simulating crowds based on a space colonization algorithm. Computers & Graphics 36 (2), 70 – 79, virtual Reality in Brazil 2011.

Chenney, S., Aug. 2004. Flow tiles. In: Symposium on Computer Animation, SCA'04. ACM SIGGRAPH/Eurographics, Grenoble, France, pp. 233–242.

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

Fattal, R., Lischinski, D., August 2004. Target-driven smoke animation. ACM Transactions on Graphics (Proc. SIGGRAPH 2004) 23, 441–448.

Guy, S. J., Chhugani, J., Curtis, S., Dubey, P., Lin, M. C., Manocha, D., 2010. PLEdestrians: A least-effort approach to crowd simulation. In: Popovic, Z., Otaduy, M. A. (Eds.), Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2010, Madrid, Spain, 2010. Eurographics Association, pp. 119–128.

Hascoët, L., Pascual, V., May 2012. The Tapenade Automatic Differentiation tool: principles, model, and specification. Rapport de recherche RR-7957, INRIA.

Helbing, D., Farkas, I., Vicsek, T., 2000. Simulating dynamical features of escape panic. Nature 407 (1), 487–490.

Henry, J., Shum, H. P. H., Komura, T., 2012. Environment-aware real-time crowd control. In: Proceedings of the 2012 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '12. ACM, New York, NY, USA.

Hughes, R. L., Jun. 2002. A continuum theory of pedestrian motion. Transportation Research part B. 36 (6), 507–535.

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

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

Kwon, T., Lee, K. H., Lee, J., Takahashi, S., Aug. 2008. Group motion editing. ACM Transactions on Graphics (Proc. SIGGRAPH 2008) 27, 80:1–80:8.

Lee, K., Choi, M., Hong, Q., Lee, J., Aug. 2007. Group behavior from video: a data-driven approach to crowd simulation. In: Symposium on Computer Animation, SCA'07. ACM SIGGRAPH/Eurographics, San Diego, California, pp. 109–118.

Lerner, A., Chrysanthou, Y., Lischinski, D., 2007. Crowds by example. Computer Graphics Forum (Proc. Eurographics 2007) 26 (3).

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

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

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

Metoyer, R. A., Hodgins, J. K., 2004. Reactive pedestrian path following from examples 20 (10), 635–649.

Musse, S. R., Thalmann, D., 2001. Hierarchical model for real time simulation of virtual human crowds. IEEE Transactions on Visualization and Computer Graphics 7 (2), 152–164.

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

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

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

Park, M., 2010. Guiding flows for controlling crowds. The Visual Computer 26, 1383–1391.

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

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

Reynolds, C. W., Jul. 1987. Flocks, herds, and schools: A distributed behavioral model. In: Proc. of SIGGRAPH 1987. ACM, pp. 25–34.

Shi, L., Yu, Y., January 2005. Controllable smoke animation with guiding objects. ACM Transactions on Graphics (Proc. SIGGRAPH 2005) 24, 140–164.

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

Sung, M., Gleicher, M., Chenney, S., 2004. Scalable behaviors for crowd simulation. Computer Graphics Forum (Proc. Eurographics 2004) 23 (3), 519–528.

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

Treuille, A., Cooper, S., Popovic, Z., 2006. Continuum crowds. ACM Transactions on Graphics (Proc. SIGGRAPH 2006) 25 (3), 1160–1168.

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

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

van den Berg, J., Lin, M., Manocha, D., May 2008. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In: 2008 IEEE International Conference on Robotics and Automation (ICRA). pp. 1928–1935.

Wojtan, C., Mucha, P., Turk, G., 2006. Keyframe control of complex particle systems using the adjoint method. In: Symposium on Computer Animation'06. pp. 15–23.
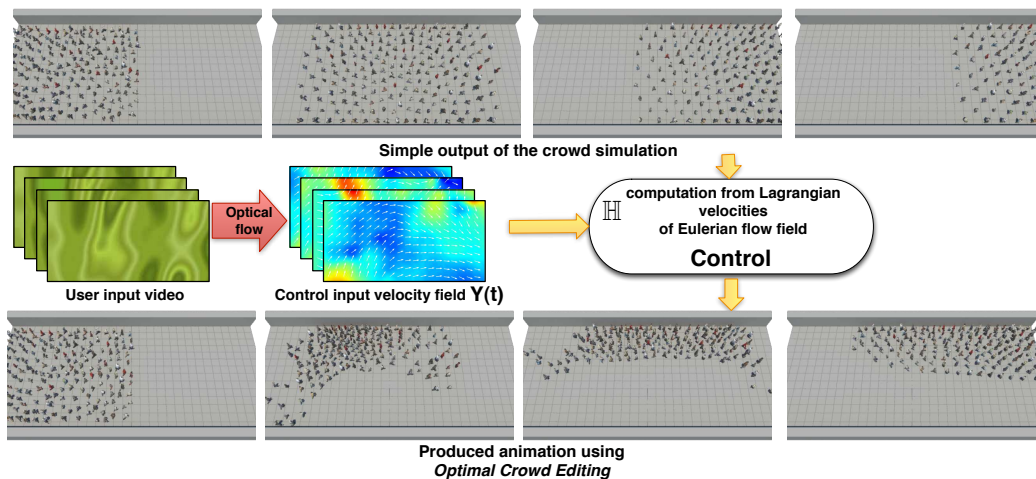
Figure 11: **Video-based motion transfer.** In this experiment the aim is to transfer the motion estimated from a given video to the crowd. A time varying motion field is first estimated thanks to an optical flow estimator. Then, this flow is used as constraints in our control procedure. The observation operator $\mathbb{H}$ relates here the pedestrians (Lagrangian) velocities to the corresponding continuum (Eulerian) flow field.
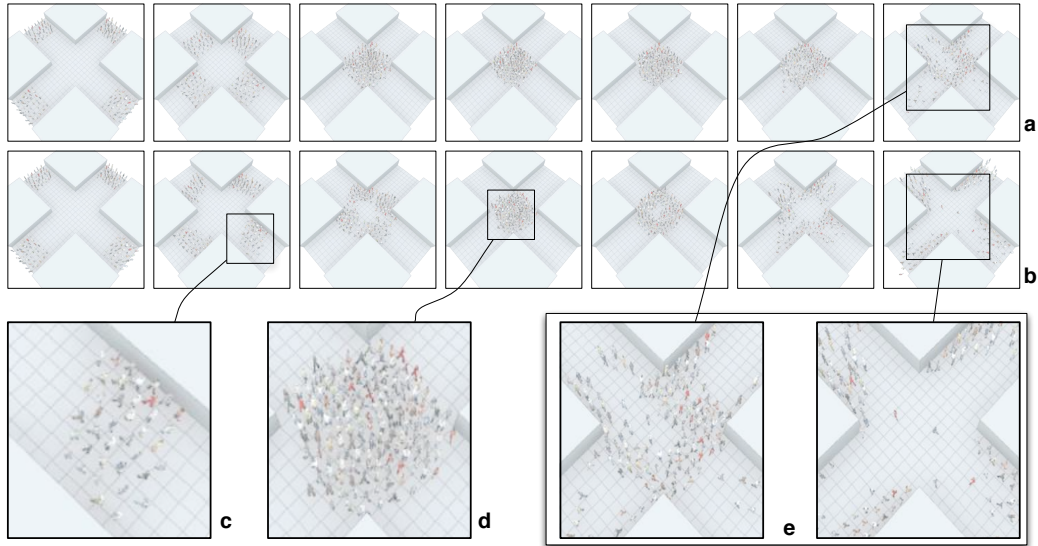
Figure 12: **Vorticity control experiment.** Comparison of seven rendered frames captured at the same instants of the (a) simple output of the crowd simulation (b) controlled simulation after 25 iterations of the control procedure. In focus (c) one can see that the group anticipates the crossing zone 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.
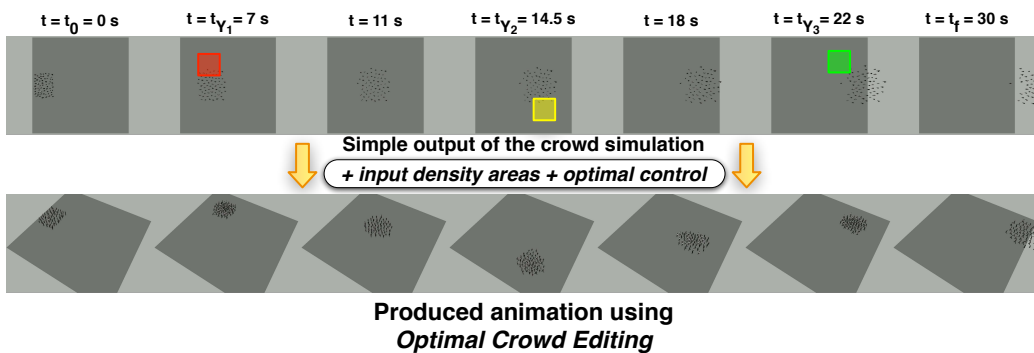

Figure 13: **Density experiment.** We aim at edit crowd motion so as the pedestrians gather in specific locations with a specific density at given times.