3D Interactive Visualization of Crowd Simulations at Urban Scale

Bruno Fanini[95]
Luigi Calori[96]

Keywords: Crowd Simulation, 3D Visualization, Real-Time, OpenSceneGraph

Abstract
Digital representations of the urban environment have constantly increased their complexity: starting from fixed chunks of 2D segment depicting the building base to complex data-sets comprising several layer of information such as multi-resolution GIS data combined with on field 3D data acquisition. Cities in particular and urban development in general emerge from the bottom up, so crowd is an essential part of the city environment but it is often absent from most of the interactive system used to present urban data and landscapes. This paper presents a modular, real-time system based on the open-source framework OpenSceneGraph, providing a realistic and efficient 3D visualization in real-time of complex animated pedestrians within large urban environments, along with tools to visualize, explore and study macroscopic and emerging properties. The system allows to represent large crowd simulations or automatized tracking of real pedestrians streamed from external applications: the component-based design aims to provide efficient workflow for a multi-disciplinary team, including 3D modelers, urban planners, computer scientists, experts of crowd dynamics and other professionals. To evaluate quality, performance and interaction, the system was applied to the case study of *"T-Days"* in Bologna historical center, using pre-existing 3D open-data assets: some results in terms of performance and workflow among different professionals will be presented.

1. Introduction

The city can be treated as a complex system, specifically, during the last two decades the research focus changed from aggregate equilibrium systems to more evolving systems whose macroscopic properties emerge from the bottom up [BATTY08], [PUMAIN03]. Some researches also describe a generic mechanism through which human social activity influences the urban space and its relation with movement [HILLIER01]. To study and analyze urban mobility in such contexts, in particular pedestrian mobility, dynamical systems are often used to model the city, with their physical constraints and networks. There are thus several reasons to simulate and represent mobility models:

- Emergency situations, such as building evacuations, to understand and to identify criticalities
- Architectural study, for instance, studying functional design of a specific building or adding realism to the visualization
- Urban study, for instance, representing a realistic crowd both from a motion and an aesthetical point of view, as support to urban planning
- Adding an artificial life layer to virtual 3D reconstructions, for instance, populating ancient cities

A crowd simulated by a computer is basically a collection of simple units (pedestrians) that share a common environment with different interactions. Realism and correctness of such interactions depend on the simulated urban system complexity and which level of detail the project aims to model. To simulate crowds in urban environments, *particle-based* approaches are typically used: each pedestrian is treated as an individual with a finite size and an internal state (microscopic approach), and all the interactions are modeled according to physical and

95 CNR ITABC, via Salaria, Km 29.300 - 00171 Monterotondo (Rome). Mail: bruno.fanini@gmail.com
96 CINECA, Via Magnanelli 6/3 - 40033 Casalecchio di Reno (Bologna). Mail: l.calori@cineca.it

social laws. Digital representations of the urban environment have constantly increased their complexity: starting from fixed chunks of 2D segment depicting the building base to complex data-sets comprising several layer of information such as *multi-resolution* GIS data combined with on field 3D data acquisition. From a perceptual point of view, the increase in rendering quality of available cityscapes put in greater evidence a quite essential component of a city: the humans that live there. Cities in particular and urban development in general emerge from the bottom up, so crowd is an essential part of the city environment but it is often absent from most of the interactive system used to present urban data and landscapes.
This paper will discuss a visualization system, its architecture and how this system can be fed by existing simulators or external automatic detection systems to allow interactive exploration of complex 3D urban environments populated by animated 3D crowds. Some results in terms of workflow and performance of the component-based approach will be shown, applied to the case study of *"T-Days"* in Bologna.

2. Common bottlenecks

Although inner details of different pedestrian mobility simulators or systems for automatic detection of real people won't be discussed here, they typically have CPU related bottlenecks. For instance, using a simulation approach, intensive loads are related to collision detection, artificial intelligence, decision making, etc. while automatic detection of real pedestrian can involve complex segmentation algorithms [MUNDER08], [SCHINDLER10], [RUJIKIETGUMJORN13]. These computations generally produce a *state* or collection of states: typical approaches provide (or allow to query) the crowd state at a given time. Since these processes do not involve graphics or do not necessarily need a visual representation, they can be deployed as external services, with the sole purpose of crowd state computation: the *micro-dynamics service*.
The graphical representation (3D) on the other hand, has its own bottlenecks, specifically when dealing with large urban 3D representations in real-time. The need for a realistic rendering of a virtual environment is common to many applications: in this context it is natural for instance to include believable pedestrian animations, such as walking, but a single pedestrian can consume a large section of graphical resources. GPU usage can grow really high when the 3D representation introduce advanced features such as per-pedestrian animations, realistic lighting and real-time shadows, within a large and complex environment on geometry level and its related resources [RYDER05], [BOSTROM12].
These issues and challenges clearly need also a multi-disciplinary approach: different professionals have to organize an efficient pipeline and coordinate different tasks for a given project. A typical scenario would include urban planners, computer scientists, 3D modelers, simulation modelers, etc. usually rising different workflow bottlenecks.

3. System architecture

The idea behind the system structure is a component-based client-sever architecture (Fig. 1): a set of modular and reusable software components able to provide a parallel pipeline among different professionals and an efficient, realistic visualization of large crowds within complex 3D urban datasets. The developed visualization modules are based on the OpenSceneGraph[97] framework [WANG10]: this C++ *rendering middleware* is widely used within simulation contexts due to its performance, portability and scalability. It is based on scene-graph structures, able to define spatial and logical relationships among different 3D models (nodes) in a virtual environment, suitable for large urban scenes. It provides support for:

- Object-oriented functionalities (re-usable components)
- Loading of common 3D formats (Alias Wavefront OBJ, 3D Studio, COLLADA, ESRI Shapefile, Autodesk, etc...)
- Management of large 3D environments, using spatial *segmentation* of the virtual world

97 http://www.openscenegraph.com

* Remote loading (URL[98])
* Efficient management of *Level-of-detail* and *instancing*: multi-resolution 3D models use appropriate representation detail depending on user position, while instancing techniques are able to reduce memory footprint through node sharing.
* *Paging*: scene portions ("*pages*") are loaded and unloaded at run-time on the main scene-graph, reducing memory footprint and GPU load depending on current view.

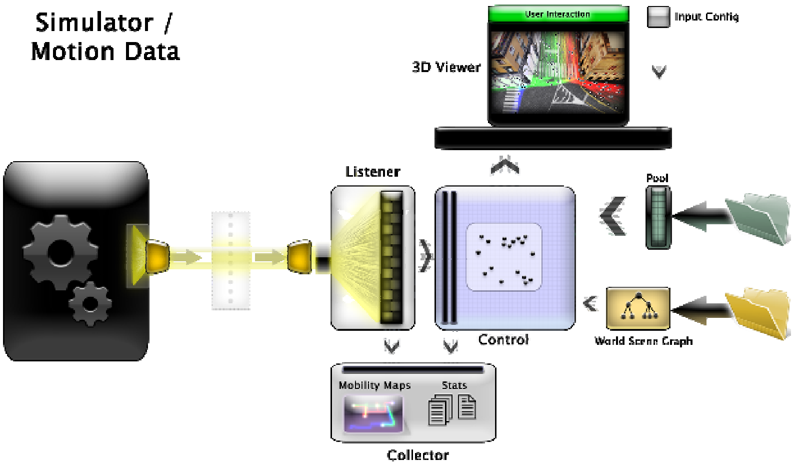

Figure 1: **System overview**

As previously mentioned, the micro-dynamics process can be thought as a service and – since its CPU consuming nature – can rely on a high-performance dedicated server machine (Fig. 1, left side). The task of such service is basically to compute and provide crowd states on client requests, specifically a function *S(t)*, returning the crowd state at time *t* within a well-defined spatial domain. In a simulation process or automatized detection (motion data), a crowd state contains the state of each single particle (pedestrian) inside the domain, with each particle identified by a unique numerical ID.
The representation system (client-side) is able to provide different real-time visualizations tools through a set of separable modules (Fig. 1, right side). Both micro-dynamics service and representation process have to define a common protocol for *inter-process*[99] communications: crowd states are passed between the two sides using a "CrowdConnector" object based on *Boost library* (http://www.boost.org/), able to pack and unpack data sent and received (Fig. 2). This allows external simulators and applications to communicate with the client visualization system on the same machine or different machines over the web.

98 Geometry nodes and resources can be remotely streamed (http://...)
99 A set of methods for the exchange of data among multiple processes

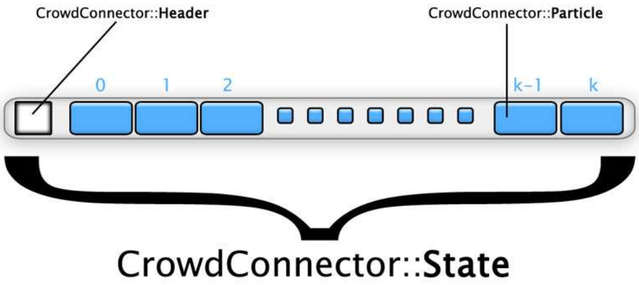

Figure 2: **Crowd state message structure containing k particles**

As part of the protocol, each particle of a crowd state is defined by a set of attributes (e.g.: ID, position, velocity, etc...) that depends on project needs and complexity of information layers to be represented or analyzed on client visualization.
The *Listener* module is able to receive a state provided by a CrowdConnector to feed the main control, updating the local crowd state and its representation. This module can be configured to query remote micro-dynamics service at given frequencies (polling time), influencing motion reliability on visualization side (Fig. 3).

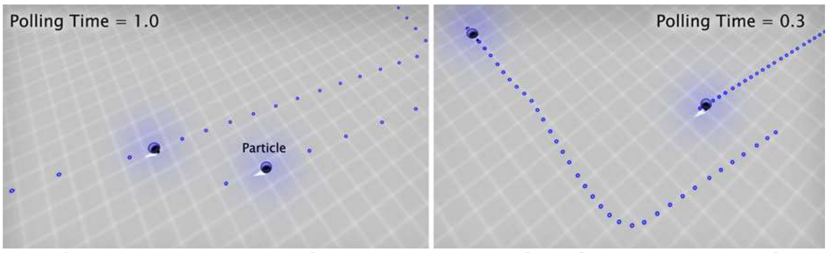

Figure 3: **Sample polling: 1 per second and 0.3 per second (increased motion reliability)**

Control module is the core of the architecture: it is responsible for components advancement and interpolation of crowd states, providing smooth representation of pedestrian motions at different polling times and to manage local 3D crowd domain. This approach allows to apply local transformations to received particles, in this case a domain mapping (geo-referencing operators for example) of a crowd into a specific urban location.
Progressing through the system modules, *World Scene-Graph* and *Pool* components are specifically designed for 3D modelers, architects and urban planners: such professionals can operate on these modules to provide 3D graphical representation of (possibly large) urban elements and pedestrian appearances (avatars), typologies and how they are associated.
Specifically, *Pool* module manages 3D animated virtual characters instances and allows advanced graph techniques for efficient particle representation. These 3D representations are stored into the Pool and can be accessed remotely: this design allows a 3D modeling team to operate independently from other architecture components. Each entry in the Pool maintains animation sequences for a single pedestrian, able to be linked at run-time to a subset of current particles. The Pool component provides two main functionalities, that can be also performed interactively at run-time:

1. *Loading*: Pool is populated with sequence-nodes. This also allows remote 3D model loading (URL)

2. *Linking*: depending on current project policy, the *p* entries in the Pool are associated to local *n* particles (the crowd) maintained by main control. Typically p < n.

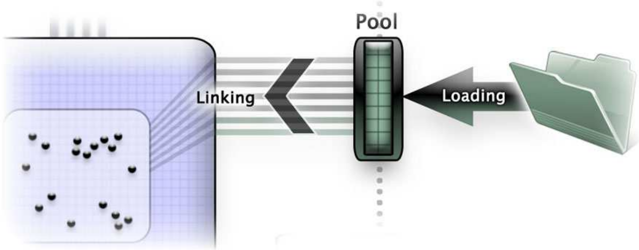

Figure 4: **Pool: loading and linking phases**

Since animating 3D large crowds often raises issues that can be addressed using approximation techniques [KAVAN08], a special approach is used to handle pedestrian animations efficiently without sacrificing realism: *3D frames*. A 3D frame is a static representation at given time of a single pedestrian, extracted from the character "walking cycle" animation: this action typically can be performed on a bone-animated 3D model. Successively, a sequence-node is generated: it acts as a controller and progresses through the k frames of the character (Fig. 5, left), also managing different animation speeds without additional computation loads on client machine, since 3D frames are pre-computed.

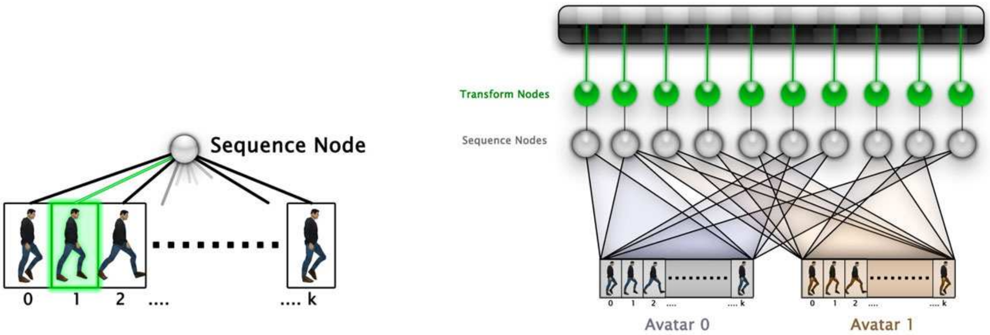

Figure 5: **Sequence-node controller and multiple animated character instancing**

This approach also allows advanced instancing techniques provided by OpenSceneGraph framework to reduce memory footprint, to differentiate representations managed by the Pool and also differentiate walking speeds (Fig. 5, right). Reuse of resources is maximized, providing a fine aesthetic approximation of 3D walking pedestrians with minimal computations involved. Additionally, this technique abstracts from bone complexity of 3D animated model and also supports level of detail features provided by OpenSceneGraph, allowing high detailed pedestrians even from close distances.

The *Collector* module is the most permeable component of the whole architecture: it is able to track single particles states and produce macroscopic visual properties such as mobility-maps, visual propensities and other statistics over the observed temporal slice (Fig. 6).

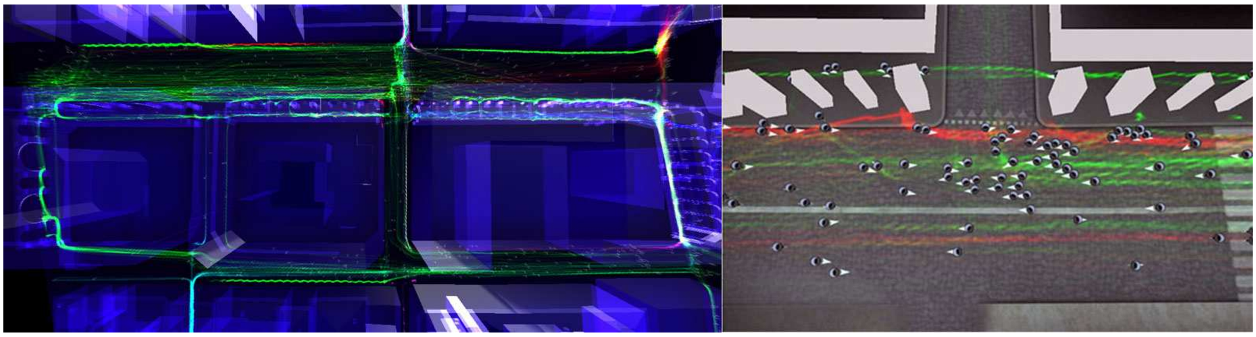

Figure 6: **Collector: mobility-maps and trail system**

This module operates on the crowd states received by the Listener and also on those processed by main control to gather mobility data. Since the *multi-dimensional* nature of urban models, additional traceable information can be visually overlayed or remotely accessed by other professionals, with the purpose of identifying criticalities of pedestrian mobility within the domain or studying the quality of the provided 2D dynamics simulator.
At last, the *Viewer* is the *Front-End* component for final users: it assembles the final scene from virtual population nodes handled by main control, urban scene-graph and information overlay nodes produced or updated by the collector module. Front-End provides real-time navigation inside the virtual world using different interaction devices, alongside an interface layer to enable or disable specific application features. 3D visualization supports a wide range of customizations such as different camera setups, stereo support and many others, depending on project objectives.

4. T-Days Case Study

To evaluate quality, performance and interaction of the developed system, we applied the architecture to the case study of "T-Days" in Bologna historical center, using pre-existing 3D *open-data* assets. Each weekend, via Rizzoli, via Indipendenza and via Ugo Bassi (creating a "T" shape) are only opened to pedestrians and bicycles that safely wander through commercial activities and historical attractions in the center. In this case, an external simulator[100] has been used as micro-dynamics service: behavioral and decision making aspects of each virtual pedestrian were treated as 2-Dimensional problem and the simulation model was progressively improved by a dedicated team through observations on the field.

100 Provided by Department of Physics (http://www.fisica-astronomia.unibo.it)

Figure 7: **identified area for the simulation and 3D environment**

3D application client on the other hand, requires in this case efficient visualization and animation of relatively large crowd: the real 3D scenario of Bologna city center needs a proper balance between realism (building details, porticoes and column geometries, etc...) and sufficient interactivity for a *real-time* targeted application.

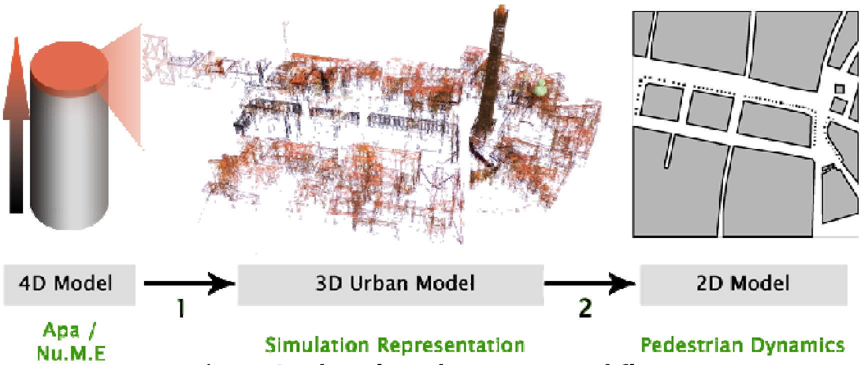

Figure 8: **virtual environment workflow**

A 3D section (domain) has been extracted from existing open-data projects (*Nu.M.E.*[101] and *Apa* movie[102] [GUIDAZZOLI12], [BOCCHI00]) based on real city plan and cartographic data, successively re-organized into native OpenSceneGraph format with related resources (Fig. 8). The advanced functionalities provided by the system such as *level of detail* and *paging* techniques (spatial segmentation of large complex 3D city portions into "pages") allow to maintain an interactive frame-rate on client side visualization, even with realistic lighting effects obtained through scalable real-time self-shadowing. A separate modeling team independently worked on pedestrian 3D representations using *Blender* software (http://www.blender.org) and then an exporter able to progressively populate the Pool.

To test visualization performance and measure quality of the system, frames-per-second (FPS) has been monitored using maximum level of detail both for pedestrian representations and urban geometries, with real-time shadows enabled (Fig. 9).

101 http://www.cineca.it/en/progetti/new-electronic-museum-city-four-dimensions-virtual-bologna
102 http://www.cineca.it/en/node/4614

Figure 9: **Real-time 3D visualization with self-shadowing and animated 3D characters**

Registered results were highly interactive and comfortably over real-time thresholds. Within this case study, specific attributes have been identified for crowd state transmissions: particle ID, position, velocity and propensity. Collector module provided some insights regarding pedestrian mobility within this specific project, visually highlighting macroscopic properties as shown in Fig.

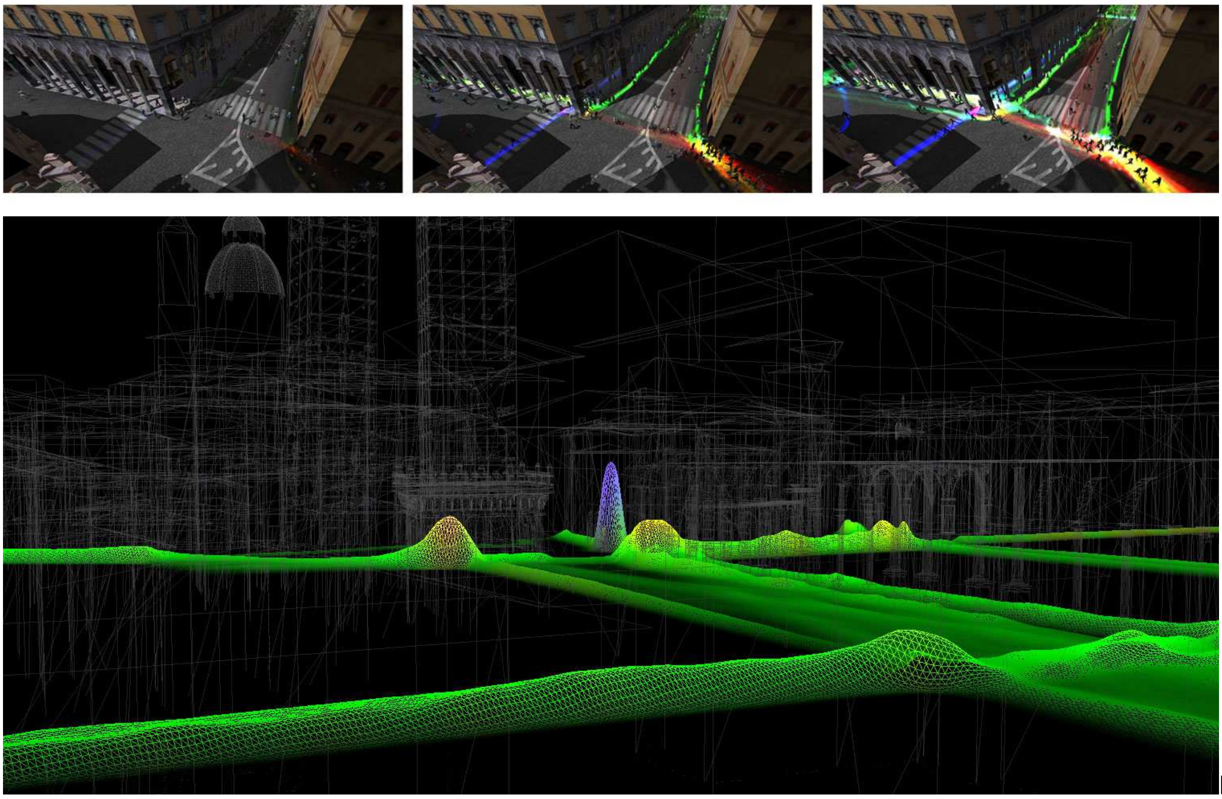

Figure 10: **Collector: evolution of mobility-maps in real-time application (top) and crowd density 3D graphs as output of the case study.**

10.

Output data from Collector also provided 3D graphs as overlays for urban model, identifying mobility criticalities, with focus on pedestrian densities under certain circumstances, such as under via Rizzoli porticoes.

5. Conclusions

Using a component-based approach, the presented system has the goal of providing an efficient client-server architecture for real-time visualization of pedestrian mobility models (implemented by simulators or real pedestrian tracking) within complex 3D multi-resolution urban scenarios. Another objective is to establish a distributed workflow, that often in these context involves different professionals with different skills: for instance, 3D modelers, architects, urban planners, computer scientists and pedestrian simulation professionals. Through the definition of particle attributes and transmission protocol between the two sides, the system is able to receive data streams from multiple simulators or automatized detection systems, providing to final users a versatile tool to compare different pedestrian micro-dynamics, even on the same virtual environment. Furthermore, the ability of the system to store crowd states on server side, also enables *on-demand* visualizations of time-slices for client users.

The developed system and its application to a few case studies, proved a smooth scalability to large 3D cityscapes with a realistic visualization using OpenSceneGraph and an efficient workflow for a multi-disciplinary team, including a 3D modeling dedicated lab, computer scientists, experts of crowd dynamics and other professionals, providing a modular tool to observe, study and visualize a 3D evolutionary urban model.

Bibliography

Batty M. *"Cities as complex systems"*, 2008.

Pumain D. *"Scaling Laws and Urban Systems"*, 2003

Hillier B. *“A Theory of the City as Object”*, 2001

Leggett R. *"Real-Time Crowd Simulation"*, 2004

Fanini B. *"An agent-based Architecture for Large Virtual Landscapes"*, 2009

Kavan L. et al. *"Polypostors: 2D Polygonal Impostors for 3D Crowds"*, 2008

Guidazzoli A., Delli Ponti F., Liguori M.C. *"27 Secoli di Storia di Bologna raccontati in un cartoon 3D"*, 2012

Bocchi F., Bonfigli M.E., Calori L., Guidazzoli A., Petrone M. *"4D Bologna Virtual Environment. The Nu.M.E. Project state of the art"*, 2000

Ryder G., Day A.M. *"Survey of real-time rendering tecniques for crowds"*, 2005

Bostrom H., Wensby L. *"Large-scale agent-based pedestrian and crowd simulation in real-time"*, 2012

Schindler K. et al. *"Automatic Detection and Tracking of Pedestrians from a moving Stereo Rig"*, 2010

Rujikietgumjorn S. and Collins R.T. *“Optimized Pedestrian Detection for Multiple and Occluded People”*, 2013

Munder S., Schnorr C., Gavrila D.M. *“Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models”*, 2008