



La simulazione parametrica come strumento per informare la rappresentazione

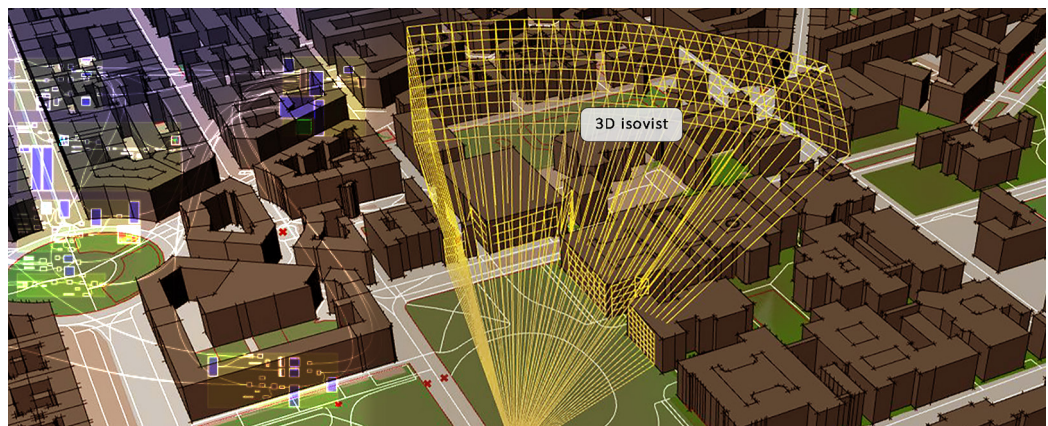
Gabriele Stancato
Barbara Piga

Abstract

Scopo di questo articolo è illustrare una modalità di simulazione parametrica, sviluppata all'interno del progetto EU H2020 EIT Digital 2019 "AR4CUP", come interazione agente-ambiente, per lo sviluppo di un'APP in realtà aumentata. L'APP consente la raccolta di feedback da parte di utenti in esplorazione di un ambiente urbano. Perciò, scopo della simulazione è sviluppare progressivi stress test in-vitro, generando in tempi rapidi svariati scenari alternativi di interazione e relativi dati. La simulazione, sviluppata secondo la metodologia agent-based modeling è stata sperimentata alterando ripetutamente il numero degli agenti, le loro posizioni di partenza e gli elementi attrattori sul territorio. Lo sviluppo di questa emulazione implica stabilire criteri per il campo visuale degli agenti, regole per l'interazione con l'ambiente e attivazione di elementi funzionali della APP. In questo modo è stato possibile generare una base dati con i quali informare le rappresentazioni degli esiti. Infine, attraverso la reiterazione del processo è stato possibile ottenere dati consistenti e sono state stabilite le modalità di visualizzazione più efficaci rispetto agli obiettivi di progetto.

Parole chiave

agent-based modeling; simulazione parametrica; clustering; app; realtà aumentata.



Obiettivi

In questo articolo illustriamo la struttura di una modalità di simulazione parametrica ideata all'interno del progetto *European Union's Horizon 2020 research and innovation program EIT Digital 2019 "AR4CUP" (Augmented Reality for Collaborative Urban Planning)*. Principale scopo dell'algoritmo è realizzare esperimenti *in-vitro* dell'utilizzo dell'APP AR4CUP, in fase di progettazione e sviluppo. La APP prevede la raccolta dati di feedback degli utenti, rispetto ai progetti urbani visualizzati in tramite realtà aumentata e la rappresentazione dei dati raccolti. Perciò, abbiamo configurato una simulazione con strumenti generativi che ci permettesse lo sviluppo di diversi scenari alternativi di interazione funzionale sui quali affinare le rappresentazioni degli input. Il vantaggio atteso da questa metodologia è la possibilità di effettuare rapidamente centinaia di esperimenti cambiando alcune variabili, ad esempio il numero dei partecipanti, prima ancora di coinvolgere un'utenza reale. La costruzione della simulazione ha pertanto consentito la realizzazione di uno *stress test in-vitro* del processo di utilizzo della APP in realtà aumentata in ambiente urbano, il che ha permesso di ottimizzare le rappresentazioni finali dei dati raccolti. Questo processo comporta la riproduzione in maniera semplificata ma credibile di una modalità di interazione tra persone e ambiente. A tale scopo abbiamo svolto le simulazioni basandoci su condizioni urbane esistenti a cui abbiamo applicato la metodologia *Agent-based modeling*. Sulla base dei risultati abbiamo comparato diverse modalità di restituzione cartografica dei dati. L'intero processo ha quindi informato lo sviluppo della APP [1].

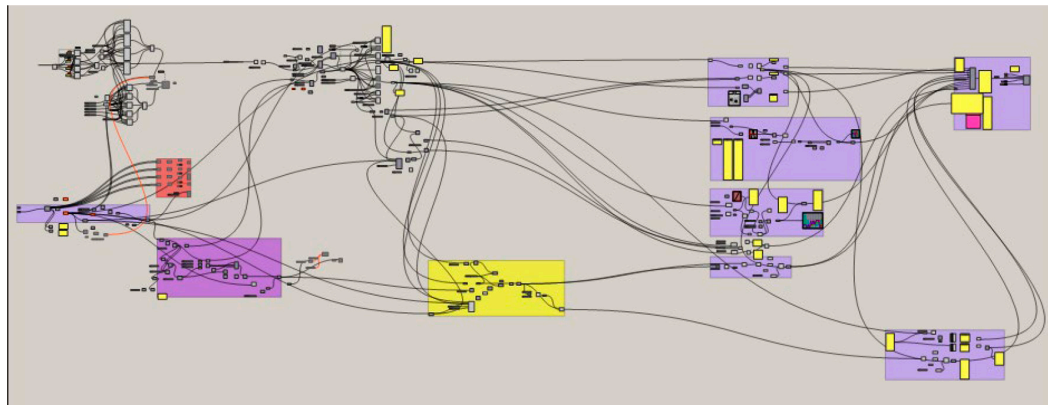


Fig. 1. Struttura complessiva dell'algoritmo.

Metodo

Per costruire l'algoritmo (fig. 1) di simulazione è stato usato il software Grasshopper 1.0.007 in ambiente Rhinoceros 6.21 con *plugin* quali: Elk OSM; Quelea; DeCoding Spaces; Python editor. Le componenti iniziali dello script generativo hanno lo scopo di ricostruire un ambiente virtuale, sulla base dei dati estratti dalle Open Street Maps (OSM), adatto alla simulazione *agent-based*. Tale ambiente deve essere sufficientemente semplificato da ridurre il peso computazionale della simulazione. Lanciata la simulazione gli agenti si muovono nell'ambiente urbano simulato e attivano le funzioni di interazione. I risultati sono quindi stati clusterizzati utilizzando programmi realizzati *ad hoc* in linguaggio Python.

Come condizioni iniziali per le simulazioni sono stati stabiliti punti di partenza per gli agenti virtuali secondo due modalità (fig. 2): 1) posizione prestabilita; 2) condizione di dispersione, ovvero agenti generati in punti casuali della zona. Abbiamo quindi impostato il sistema per valutare attrattori prestabiliti o per rilevare orientamenti 'liberi' degli agenti. Con questo metodo è possibile simulare quattro condizioni ambiente-agente: a) *sameness-spread*: gli agenti hanno uguale interesse per tutti i luoghi e muovono da posizioni randomizzate sul

territorio; b) *sameness-gate*: gli agenti hanno uguale interesse per tutti i luoghi ma si muovono da posizioni predefinite; c) *attractor-spread*: l'agente è consapevole dei luoghi ma deve raggiungerli da posizioni randomizzate; d) *attractor-gate*: l'agente è consapevole dei luoghi e inizia la ricerca da punti imposti.

Le quattro condizioni sopra menzionate sono state avviate con quantità di agenti progressivamente crescenti (da un minimo di 10 fino a 100) e mantenute per un tempo sufficiente a produrre a ogni simulazione almeno 10 interazioni per agente con le funzioni APP emulate dallo *script*.

Poiché la simulazione è orientata a emulare l'esperienza di un utente che esplora una zona urbana attraverso un dispositivo mobile in *augmented reality*, abbiamo limitato il campo visivo degli agenti alla sola visione frontale binoculare [Spector 1990], senza considerare l'estensione della visione periferica, anche in coerenza con il Field Of View (FOV) di dispositivi cellulari maggiormente diffusi (in media intorno ai 60°). L'interazione degli agenti con la APP avviene all'interno dello spazio visibile da ciascuno di loro (fig. 3), la visione degli agenti è controllata tramite plugin DeCoding Spaces [Bielik 2012].

All'interno dell'estensione dei 60° vengono quindi proiettati 600 vettori in campo planare (10 per ogni grado di campo visivo) che intercettano i volumi dell'ambiente costruito e determinano il tracciato della isovist direzionale [Benedikt 1979; Davis & Benedikt 1979; Batty 2001].

Abbiamo inoltre stabilito la profondità di visuale (connessa all'orientamento verso gli oggetti da parte degli agenti virtuali) a 400 m, come distanza accettabile nel raggiungimento di mete a piedi [Atash 1994; Aultman-Hall et al. 1997; Southworth 2005]; questi parametri ci hanno permesso di contenere l'onere di calcolo e quindi alleggerire la simulazione nel suo insieme. Considerando la densità dei vettori proiettati nel campo visuale e la distanza massima di osservazione si può stimare che al margine esterno dei 400 m è possibile intercettare geometrie non inferiori a 1,65 m. Questa dimensione viene considerata accettabile in quanto i manufatti edilizi nella zona interessata dal modello hanno minima dimensione pari a 3,17m. Le impostazioni del comportamento esplorativo dell'*Agent-based model* seguono criteri cinestetici particolarmente noti e consolidati nella letteratura di settore [Helbing & Molnár 1995; Bonabeau 2002; Pan et al. 2007]; questi sono attuati tramite il plugin Quelea-beta 0.2 e sono esposti di seguito.

a) *Avoidance*: impone un comportamento di evitamento degli ostacoli, è stato adottato per indurre gli agenti a girare attorno agli edifici.

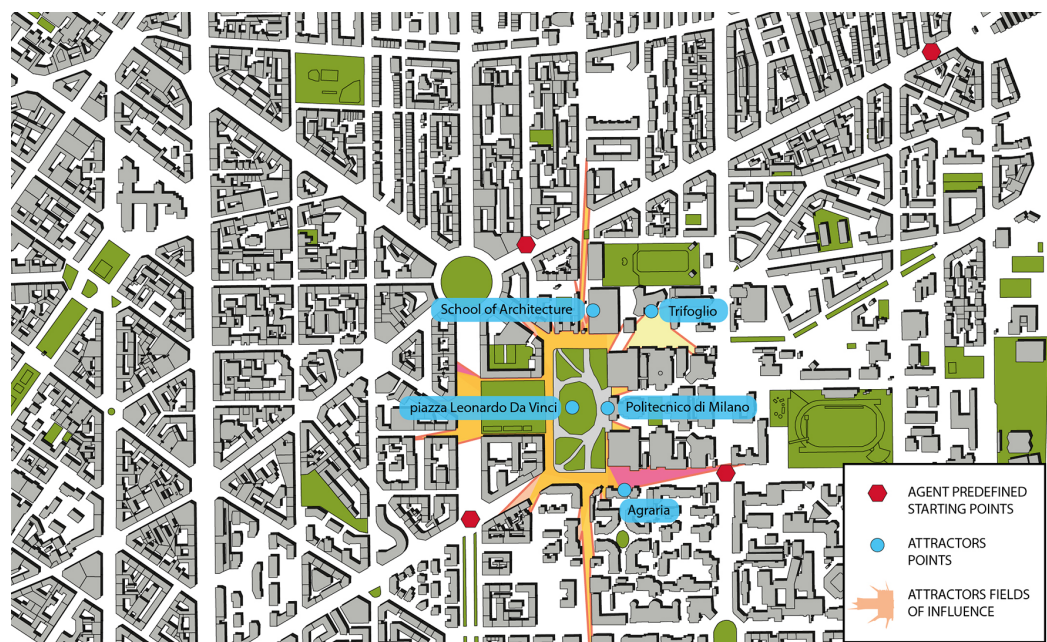


Fig. 2. Keymap, posizioni predefinite degli agenti e degli attrattori.

b) *Containing*: mantiene gli agenti all'interno di un bounding box che racchiude tutti gli edifici generati, la scelta di questo parametro evita che gli agenti si incammino oltre i confini della zona durante il proprio girovagare.

c) *View*: questo parametro fa sì che quando la visuale di una agente è bloccata da un altro agente, il primo si scansi per guadagnare una nuova visuale.

d) *Seek*: questo parametro è fondamentale nella condizione di presenza di un attrattore e stabilisce un comportamento di ricerca mirata.

e) *Arrive*: questo parametro stabilisce che gli agenti rallentino il proprio cammino una volta raggiunto l'attrattore.

Gli agenti rispondono solo ai parametri impostati, quindi rappresentano una estrema semplificazione rispetto a una condizione reale, inoltre nel modello non sono considerati come ostacoli elementi puntuali quali gli alberi. Abbiamo tralasciato questi elementi, nonostante abbiano un ruolo nelle interazioni reali, in quanto l'obiettivo dell'algorithm non era riprodurre con esattezza un comportamento umano, ma di produrre una base di dati plausibile sulla quale verificare e affinare gli strumenti di analisi e rappresentazione programmati ad hoc per la APP finale. Infatti, i record generati durante le simulazioni rappresentano principalmente informazioni posizionali e temporali.

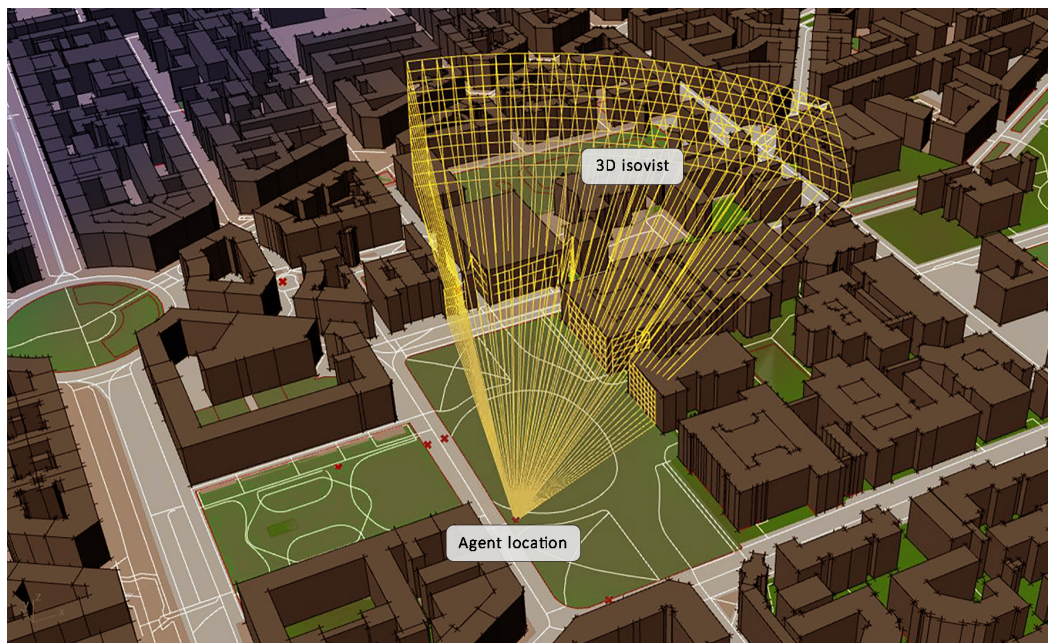


Fig. 3. Ambiente virtuale, localizzazione di un agente e rispettiva isovist.

Per evidenziare i luoghi in cui tendono a concentrarsi i fenomeni registrati si è optato per la clusterizzazione sulla base dei dati posizionali, in questo modo si aggregano i dati rispetto a condizioni locali omogenee. Il software di clusterizzazione è stato sviluppato ad hoc con linguaggio di programmazione Python 3.8, in particolare sono state adottate le librerie: 1) *Scikit-Learn 0.22.1* per le sezioni di machine learning; 2) *Matplotlib 3.1.1* per la costruzione dei grafici statistici; 3) *Ipyleaflet 0.12.2* per le rappresentazioni cartografiche.

Il sistema adottato per la clusterizzazione è stato il Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Il vantaggio di questo metodo di clusterizzazione consiste nel poter utilizzare raggruppamenti con disposizioni non necessariamente regolari [Ester et al. 1996; Tufféry 2011], in altri termini i punti analizzati possono distribuirsi in una qualunque configurazione non sferica nello spazio [2]. Per ciascun cluster si procede quindi al calcolo del centroide delle posizioni quale punto rappresentativo di un gruppo di osservazioni.

Rappresentazione dei risultati

Nel passaggio dalla rappresentazione in ambiente virtuale alla restituzione su mappa strutturata secondo coordinate GPS sono state affrontate questioni relative alla corretta corrispondenza con il sistema cartesiano XYZ. Per allineare il modello alle mappe finali si è quindi adottato il sistema di proiezione *Universal Transverse Mercator* (UTM) per la zona T32 corrispondente a Milano. Poiché le coordinate cartesiane degli spostamenti degli agenti sono calcolate relativamente all'origine del modello è stato necessario relativizzare le coordinate alle UTM corrispondenti nella realtà, ed eseguire quindi la conversione dei valori ottenuti in coordinate GPS. Per eseguire questo doppio passaggio con il massimo della precisione è stata preferita la libreria *python UTM 0.5.0*. Per stressare la contrapposizione tra mappa di fondo e tracciamento dei risultati abbiamo optato per la modalità di visualizzazione *Stamen Toner* fornita da *CityTracking* [3], mantenendo comunque le opzioni per livelli Ortofoto Satellitare e OSM a scopo di confronto.

Le mappe principali che abbiamo adottato per rappresentare i fenomeni spazio-temporali e aggregativi sono due:

a) Mappa delle posizioni clusterizzate (figg. 4-7, a sinistra): questa mappa mostra i centroidi dei raggruppamenti di agenti. In questa mappa i centroidi sono rappresentati come cerchi il cui raggio è proporzionale al numero di utenti contenuti nel cluster.

b) *Heatmap* di stazionamento (figg. 4-7, a destra): rappresenta le posizioni puntuali assunte dagli utenti durante le interazioni con le funzioni della APP, mentre il tempo di permanenza è rappresentato tramite un gradiente cromatico che mostra i tempi simulati minori (min. 10 s) in toni del blu e i tempi di permanenza maggiori in toni del rosso (max. 400 s). Da questa mappa è possibile verificare quali sono le localizzazioni in cui l'interazione è stata più impegnativa in termini temporali.

Abbiamo reiterato il procedimento descritto decine di volte verificando a ogni passaggio la qualità del dato e l'utilità della risposta grafica, fino al raggiungimento dell'obiettivo progettuale auspicato.



Fig. 4. Condizione *sameness-spread*. Nessun attrattore, agenti generati su posizioni *random* (sinistra: centroidi dei cluster, destra: *heatmap*).

Caso Studio

L'area selezionata come caso applicativo per la simulazione è la zona di piazza Leonardo Da Vinci, sede del Politecnico di Milano. L'intorno considerato nel modello copre una superficie di circa due Km². Gli attrattori predefiniti sono: 1) l'area verde di piazza Leonardo Da Vinci; 2) l'ingresso principale del Politecnico in piazza Leonardo Da Vinci; 3) l'edificio soprannominato Trifoglio in via Edoardo Bonardi; 4) la facciata della facoltà di architettura in via A. M. Ampère.

La mappa OSM è stata inizialmente integrata con viali non presenti nel file originario. Il Politecnico di Milano è qui rappresentato come un insieme di edifici singoli anziché come un unico blocco, questa scelta permette di considerare validi i percorsi interni usualmente utilizzati da docenti, studenti e personale impiegato.

Gli attrattori predefiniti sono stati scelti sulla base della nostra familiarità e conoscenza professionale del luogo. Lo spazio di influenza di un attrattore è stato considerato come luogo dei punti da cui è visibile la sua posizione puntuale.

Una volta lanciata la simulazione è stato possibile comparare gli esiti con parametri diversi, le variabili che influenzano l'algoritmo sono il numero degli agenti generati, la posizione di partenza di questi e la presenza di attrattori verso i quali possano rivolgere la propria attenzione.

Nell'interpretazione delle mappe, si deve considerare che gli agenti tendono a utilizzare percorsi quanto più brevi possibili mentre i fruitori di un luogo potrebbero seguire andamenti più articolati. Da questa tendenza derivano andamenti trasversali ai complessi edilizi e a disegnare traiettorie rettilinee quando gli agenti intercettano un attrattore. Inoltre, maggiore è la distanza dei punti di partenza dagli attrattori, maggiore è la difficoltà per gli agenti di intercettarli e di conseguenza maggiore sarà la probabilità che gli agenti rimangano distanti disegnando geometrie globulari.



Fig. 5. Condizione attrattore-spread. Attrattori predeterminati, agenti generati su posizioni random, (sinistra: centroidi dei cluster, destra: heatmap).

Nelle simulazioni in cui sono mantenute libere sia le posizioni di partenza sia gli attrattori (*sameness-spread*, fig. 4), la simulazione produce una distribuzione centrifuga in cui gli agenti tendono a interagire tra loro piuttosto che con l'ambiente. Le aree di interazione sono disposte lontano dagli spazi che nella realtà sono deputati all'interazione sociale, quello che osserviamo è un effetto per il quale gli edifici funzionano esclusivamente come barriere e non come elementi orientatori. Poiché questa condizione non è realistica, soprattutto in relazione alla realtà di piazza Leonardo Da Vinci, abbiamo deciso di impostare il sistema su posizioni di partenza predeterminate note per essere nodi viari e di trasporto pubblico rilevanti.

Nelle simulazioni in cui le posizioni di partenza sono prestabilite ma gli attrattori sono lasciati indefiniti (*sameness-gate*), gli agenti orientano le proprie preferenze lungo il perimetro stradale che percorrono nella perlustrazione. Come si può osservare dalla fig. 5 il tracciato percorso dagli agenti segue una logica di intercettazione reciproca ovvero, i sottogruppi tendono a ricercarsi tra loro in quanto le varie posizioni rientrano nei rispettivi raggi di osservazione. Gli edifici sono ancora considerati come meri ostacoli e questo si traduce in azioni di aggiramento delle barriere. Questo caso può essere interpretato come una routine di passaggio indifferente attraverso il quartiere, questa simulazione non è stata considerata coerente con l'uso della APP e si è perciò deciso di fissare la posizione degli attrattori nelle successive simulazioni.

Nelle simulazioni in cui è stata stabilita la presenza di attrattori predeterminati e agenti che nascono su posizioni random (*attractor-spread*) (fig. 6), gli agenti individuano come fulcro di interesse lo spazio aggregativo vuoto rappresentato da piazza Leonardo Da Vinci. La geometria dei percorsi mostra che gli attrattori costituiscono un campo di influenza centripeto. Su queste considerazioni abbiamo impostato una quarta modalità di simulazione stabilendo posizioni predeterminate sia per gli attrattori che per gli agenti. Questa ulteriore impostazione rappresenta un sistema più coerente con la realtà e l'uso dell'area di piazza Leonardo Da Vinci.

Nelle simulazioni in cui i punti di partenza sono stati predeterminati e gli attrattori prefissati (*attractor-gate*), gli agenti si sono concentrati nell'area interessata dagli attrattori sia in termini posizionali che temporali. La prossimità di due punti di partenza all'attrattore posto sull'edificio di Agraria ha determinato uno sbilanciamento posizionale delle interazioni in piazza Leonardo Da Vinci. Come si può osservare in fig. 7 e fig. 8, gli agenti tendono a spostarsi a sud verso via Celoria nel convergere verso la centralità della piazza. Poiché gli edifici del Politecnico si frappongono tra i punti di partenza e gli attrattori, e in particolare verso il gruppo generato alla stazione di Lambrate, si verifica un utilizzo intenso dei percorsi interni al Campus quali scorciatoie per raggiungere gli obiettivi.



Fig. 6. Condizione *sameness-gate*. Nessun attrattore, agenti generati su posizioni predeterminate (sinistra: centroidi dei cluster; destra: heatmap).

Conclusioni

La strutturazione e spazializzazione delle interazioni degli agenti ha consentito di sviluppare numerosi scenari alternativi e confrontare preventivamente comportamenti emergenti a cui la APP deve poter far fronte. Il processo non può essere completamente automatizzato ma deve essere informato da un professionista che con la propria competenza individui attrattori effettivi del territorio, in modo da qualificare la simulazione.

Ad ogni modo, l'utilizzo della strategia procedurale descritta si è dimostrato efficace quale stress test degli strumenti di raccolta e rappresentazione dei dati di posizionamento degli agenti. La reiterazione di simulazioni verificate di volta in volta con la rappresentazione cartografica e statistica dei dati ci ha permesso di verificare la tenuta qualitativa dei costrutti di base rispetto a cambiamenti delle dinamiche di risposta degli utenti. Dagli esiti delle indagini tramite simulazione si è identificato come utile strumento le rappresentazioni cartografiche interattiva, ovvero sia la possibilità di attivare indipendentemente diversi livelli di informazioni e di selezionare campi di informazione specifici selezionando elementi puntuali. La simulazione si è rivelata un valido strumento per anticipare eventuali problemi tecnici e per indirizzare in maniera efficace la comunicazione verso gli utenti e verso i *developer*.

Fig. 7. Condizione attractor-gate. Attrattori su posizioni predeterminate, agenti generati su posizioni predeterminate (sinistra: centri dei cluster; destra: heatmap).

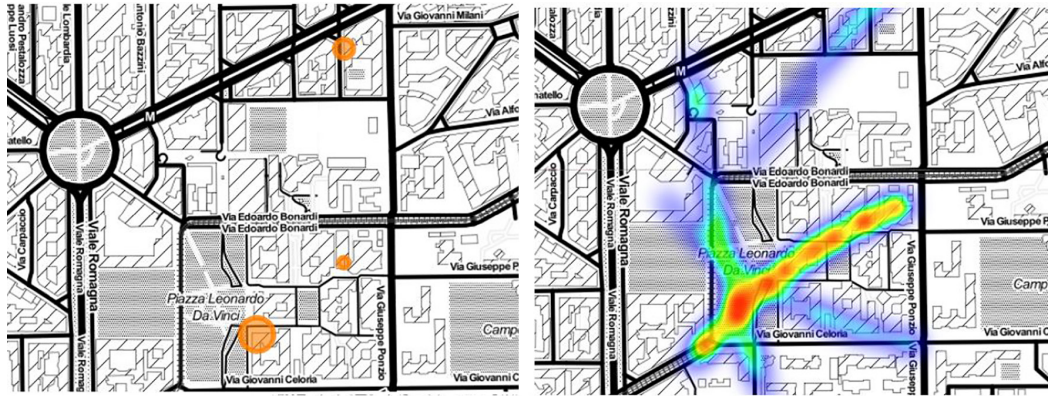


Fig. 8. Esempi di aggregazione di agenti virtuali in due diverse simulazioni, i numeri all'interno dei cerchi mostrano l'ammontare degli agenti in un gruppo.

Note

- [1] La APP AR4CUP verrà rilasciata sul mercato ad aprile 2020.
- [2] 2.3. Clustering – scikit-learn 0.22.1 documentation. Scikit-learn.org. Retrieved 6 February 2020 da: <https://scikit-learn.org/stable/modules/clustering.html#>.
- [3] Stamen Maps, Maps.stamen.com. <http://maps.stamen.com/#terrain/12/37.7706/-122.3782>.

Ringraziamenti

Questo progetto ha ricevuto un finanziamento dal programma di ricerca e innovazione Horizon 2020 dell'Unione Europea EIT Digital 2019 AR4CUP | Augmented Reality for Collaborative Urban Planning - La responsabilità di questa pubblicazione è esclusivamente dell'autore. L'Unione Europea non è responsabile per qualsiasi uso che possa essere fatto delle informazioni in essa contenute.

Riferimenti bibliografici

- Atash Farhad (1994). Redesigning Suburbia for Walking and Transit: Emerging Concepts. In *Journal of Urban Planning and Development*, 120 (1), pp. 8-57.
- Aultman-Hall Lisa, Roorda Matthew & Baetz Brian W. (1997). Using GIS for Evaluation of Neighborhood Pedestrian Accessibility. In *Journal of Urban Planning and Development*, 123 (1), pp. 10-17.
- Batty Michael (2001). Exploring Isovist Fields: Space and Shape. In *Architectural and Urban Morphology. Environment and Planning B: Planning and Design*, 28 (1), pp. 123-150.
- Benedikt Michael L. (1979). To Take Hold of Space: Isovists and Isovist Fields. In *Environment and Planning B: Planning and Design*, 6 (1), pp. 47-65.
- Bielik Martin S. (2012). Parametric Urban Patterns: Exploring and integrating graph-based spatial properties in parametric urban modelling. In Achten Henri; Pavlicek Jiri; Hulin Jaroslav; Matejovska Dana (Eds.). *Digital Physicality – Proceedings of the 30th ECAADe Conference - Volume 1*, Czech Technical University in Prague, Faculty of Architecture (Czech Republic), 12-14 september 2012, pp. 701-708.
- Bonabeau Eric (2002). Agent-based modeling: Methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences of the United States of America*, 99 (suppl. 3), pp. 7280-7287.
- Davis Larry S., & Benedikt Michael L. (1979). Computational models of space: Isovists and isovist fields. In *Computer Graphics and Image Processing*, 11(1), pp. 49-72.
- Ester Martin, Kriegel Hans-Peter & Xu Xiaowei (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD-96 Proceedings*, pp. 226-231.
- Helbing Dirk, & Molnár Péter (1995). Social force model for pedestrian dynamics. In *Physical Review E*, 51(5), pp. 4282-4286.
- Macal Charles M. & North Michael J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference*, pp. 2-15.
- Pan Xiaoshan et al. (2007). A multi-agent-based framework for the simulation of human and social behaviors during emergency evacuations. In *AI & SOCIETY*, 22(2), 2007, pp. 113-132.
- Southworth Michael (2005). Designing the Walkable City. In *Journal of Urban Planning and Development*, 131(4), pp. 246-257.
- Spector Robert H. (1990). Visual Fields. In Walker H.K., Hall W.D., Hurst J.W. (eds.). *Clinical Methods: The History, Physical, and Laboratory Examinations*. 3rd edition. Boston: Butterworths, Chapter 116. pp. 565-572.
- Tufféry Stéphane (2011). *Data Mining and Statistics for Decision Making: Tufféry/*Data Mining and Statistics for Decision Making. Hoboken: John Wiley & Sons, Ltd.

Autori

Stancato Gabriele, Politecnico di Milano, gabriele.stancato@polimi.it
Piga Barbara E.A., Politecnico di Milano, barbara.piga@polimi.it

Per citare questo capitolo: Stancato Gabriele, Piga Barbara (2020). La simulazione parametrica come strumento per informare la rappresentazione/Parametric simulation as a tool for reporting representation. In Arena A., Arena M., Brandolino R.G., Colistra D., Ginex G., Mediatì D., Nucifora S., Raffa P. (a cura di). *Connettere. Un disegno per annodare e tessere. Atti del 42° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione/Connecting. Drawing for weaving relationships. Proceedings of the 42th International Conference of Representation Disciplines Teachers*. Milano: FrancoAngeli, pp. 829-846.



Parametric Simulation as a Tool to Inform Representation

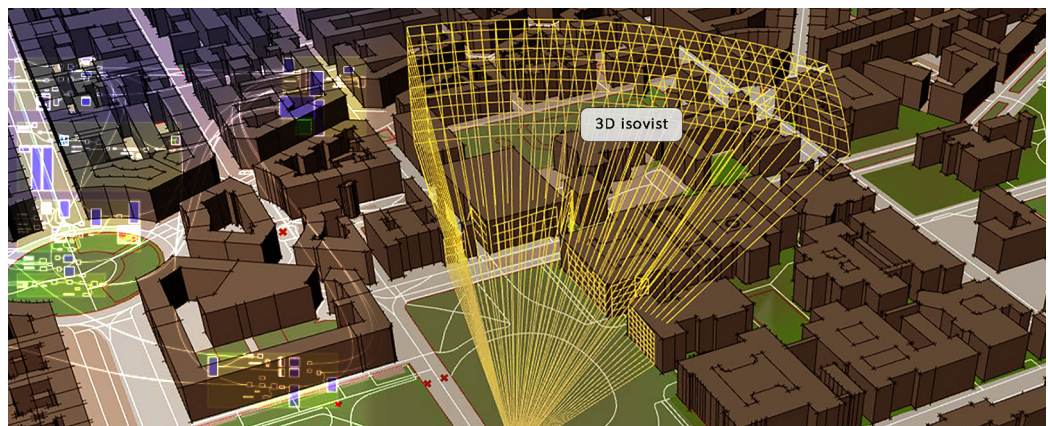
Gabriele Stancato
Barbara Piga

Abstract

The purpose of this article is to illustrate a parametric simulation mode, developed within the EU H2020 EIT Digital 2019 “AR4CUP” project, as an agent-environment interaction, for the development of an Augmented Reality app. The app makes it possible to collect feedback from users exploring an urban environment. Therefore, the purpose of the simulation is to develop progressive in-vitro stress tests, quickly generating various alternative interaction scenarios and related data. The simulation, developed according to the agent-based modelling methodology, has been tested by repeatedly altering the number of agents, their starting positions and the attractors on the territory. The development of this emulation involves establishing criteria for the field of view of the agents, rules for interaction with the environment and activation of functional elements of the app. This made it possible to generate a database with which to report representations of the results. Finally, by repeating the process it was possible to obtain consistent data and to establish the most effective ways of visualising the project objectives.

Keywords

agent-based modelling; parametric simulation; clustering; app; augmented reality.



Objectives

In this article we illustrate the structure of a parametric simulation mode conceived within the European Union's Horizon 2020 research and innovation program, EIT Digital 2019 "AR4CUP" | (Augmented Reality for Collaborative Urban Planning). The main purpose of the algorithm is to perform *in-vitro* experiments on the use of the AR4CUP app, in the design and development phase. The app provides for the collection of user feedback data, with respect to the urban projects displayed in Augmented Reality, and the representation of the data collected. Therefore, we configured a simulation with generative tools that would allow us to develop different alternative scenarios of functional interaction on which to refine input representations. The expected advantage of this methodology is the ability to quickly carry out hundreds of experiments by changing some of the variables, for example the number of participants, before involving real users. The construction of the simulation therefore enabled us to carry out an *in-vitro* stress test of the process of using the app in Augmented Reality in an urban environment, which made it possible to optimise the final representations of the data that was collected. This process involves the simplified but credible reproduction of a mode of interaction between people and the environment. For this purpose, we carried out the simulations based on existing urban conditions to which we applied the Agent-based modelling methodology. Based on the results obtained, we compared different ways of cartographic data return. The entire process therefore moulded the development of the app [1].

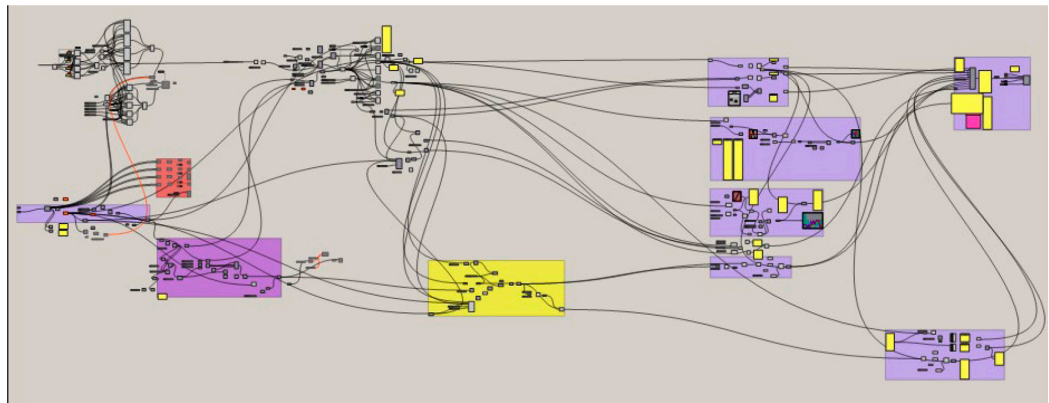


Fig. 1. Overall structure of the algorithm.

Method

To build the simulation algorithm (fig.1) Grasshopper 1.0.007 software was used in Rhinoceros 6.21 environment with plugins such as: Elk OSM; Quelea; DeCoding Spaces; Python editor. The initial components of the generative script aim to reconstruct a virtual environment, based on data extracted from Open Street Maps (OSM), suitable for agent-based simulation. This environment must be sufficiently simplified to reduce the computational weight of the simulation. Once the simulation is launched, the agents move within the simulated urban environment and activate the interaction functions. The results were then clustered using ad-hoc programs in Python.

As initial conditions for the simulations, starting points for virtual agents were established in two ways (fig. 2): 1) predefined position; 2) dispersion condition, i.e. agents generated at random points in the area. We then set up the system to evaluate pre-established attractors or to detect "free" agent orientations. With this method it is possible to simulate four environment-agent conditions: a) sameness-spread: agents have equal interest in all places and move from randomised positions on the territory; b) sameness-gate: agents

have equal interest in all places but move from predetermined positions; c) attractor-spread: the agent is aware of the places but must reach them from randomised positions; d) attractor-gate: the agent is aware of the places and starts the search from set points.

The above four conditions were initiated with progressively increasing numbers of agents (from a minimum of 10 up to 100), and were maintained long enough to produce at least 10 interactions per agent with the app functions emulated by the script at each simulation. Since the simulation is directed at emulating the experience of a user exploring an urban area through a mobile device in Augmented Reality, we have limited the agents' field of view to binocular frontal vision only (Spector 1990), without considering the extension of peripheral vision, even in coherence with the Field OfView (FOV) of the most widespread mobile devices (on average around 60 degrees). The agents' interaction with the app takes place within the space visible by each of them (fig. 3): the agents' vision is controlled by the DeCoding Spaces plugin [Bielik 2012].

Within the 60° extension, 600 vectors are then projected in a planar field (10 for each degree of visual field), which intercept the volumes of the built environment and determine the path of the directional isovist [Benedikt 1979; Davis & Benedikt 1979; Batty 2001]. We also established the depth of view (related to the orientation towards objects by virtual agents) at 400 m, as an acceptable distance when reaching goals on foot [Atash 1994; Aultman-Hall et al. 1997; Southworth 2005]; these parameters allowed us to contain the calculation burden and thus lighten the simulation as a whole.

Considering the density of the vectors projected into the field of view and the maximum distance of observation, it can be estimated that at the outer margin of 400 m it is possible to intercept geometries not less than 1.65 m. This dimension is considered acceptable since the building structures in the area covered by the model have a minimum size of 3.17 m.

The exploratory behaviour approaches of the Agent-based model follow kinaesthetic criteria that are particularly well known and well established in industry articles (Helbing & Molnár 1995; Bonabeau 2002; Pan et al. 2007); these are implemented through the Quelea-beta 0.2 plugin, and are shown below.

a) Avoidance: this imposes a behaviour of avoiding obstacles; it has been adopted to induce agents to go around buildings.

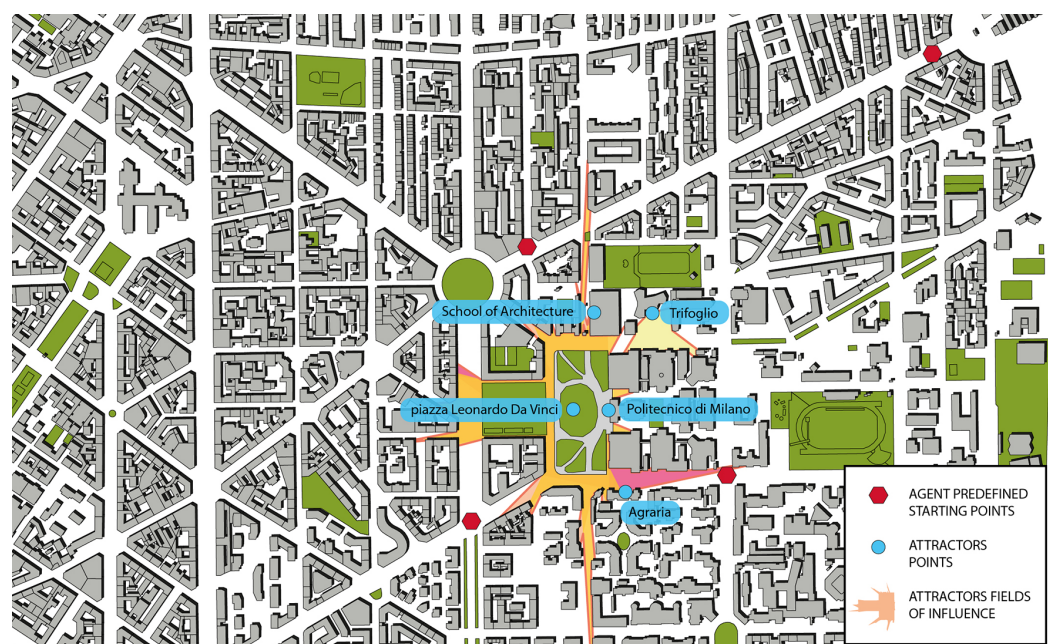


Fig. 2. Keymap, default positions of agents and attractors.

- b) Containing: keeps agents inside a bounding box that encloses all the buildings generated. The choice of this parameter prevents agents from going beyond the boundaries of the area during their wanderings.
- c) View: This parameter makes sure that when the view of one agent is blocked by another agent, the first one will move aside, to gain a new view.
- d) Seek: this parameter is fundamental in the condition of presence of an attractor and establishes a targeted search behaviour.
- e) Arrive: this parameter establishes that the agents slow their pace once they reach the attractor.

The agents respond only to the set parameters, so they represent an extreme simplification compared to a real condition. Moreover, in the model, point elements such as trees are not considered to be obstacles. We omitted these elements, even though they play a role in real interactions, since the objective of the algorithm was not to exactly reproduce a human behaviour; but to produce a plausible database on which to verify and refine the analysis and representation tools programmed ad hoc for the final app. Indeed, the records generated during the simulations represent mainly positional and temporal information.

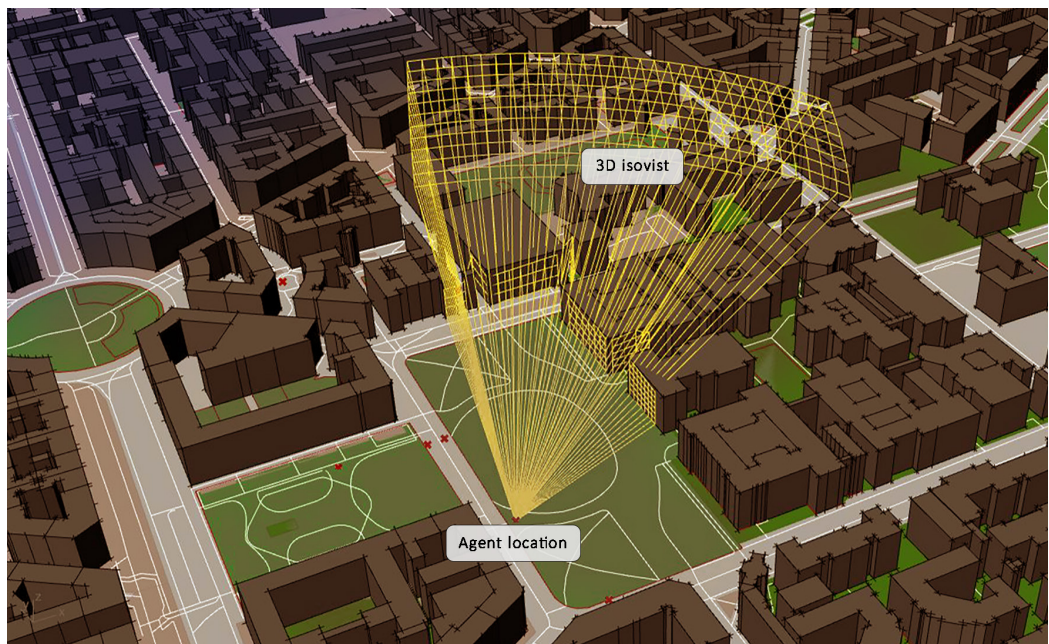


Fig. 3. Virtual environment, location of an agent and the respective isovist.

In order to highlight the places where the recorded phenomena tend to be concentrated, we opted for clustering on the basis of positional data. In this way data are aggregated with respect to homogeneous local conditions. The clustering software was developed ad hoc with Python 3.8 programming language, and in particular the following libraries were adopted: 1) Scikit-Learn 0.22.1 for machine learning sections; 2) Matplotlib 3.1.1 for the construction of statistical graphs; 3) Ipyleaflet 0.12.2 for map representations.

The system adopted for clustering was the Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The advantage of this clustering method is that it is possible to use groupings with arrangements that are not necessarily regular [Ester et al. 1996; Tufféry 2011], in other words the points analysed can be distributed in any non-spherical configuration in space [2]. For each cluster the centroid of the positions is then calculated as a representative point of a group of observations.

Representation of results

In the transition from the representation in virtual environment to the return on a map structured according to GPS coordinates, issues related to the correct correspondence with the Cartesian system XYZ have been addressed. To align the model with the final maps, the Universal Transverse Mercator (UTM) projection system was adopted for the T32 zone corresponding to Milan. Since the Cartesian coordinates of the agents' movements are calculated in relation to the model origin, it was necessary to relativise the coordinates to the corresponding UTM in reality, and then convert the values obtained into GPS coordinates. The python library UTM 0.5.0 was preferred, in order to perform this double step with maximum precision. To highlight the contrast between the background map and result tracking, we opted for the Stamen Toner display mode provided by CityTracking, while maintaining the options for Satellite Orthophoto and OSM levels for comparison purposes.

There are two main maps that we used to represent the space-time and aggregative phenomena:

a) Clustered positions map (figs. 4-7, left): This map shows the centroids of the agent groupings. In this map the centroids are represented as circles whose radius is proportional to the number of users contained in the cluster.

b) Standing heatmap (figs. 4-7, right): represents the point positions taken by users during interactions with the app functions, while the standing time is represented by a chromatic gradient that shows the simulated shorter times (min. 10 s) in blue tones and the longer dwell times in red tones (max. 400 s). From this map it is possible to verify which are the locations where the interaction has been most challenging in terms of time.

We repeated this procedure dozens of times, verifying at each step the quality of the data and the usefulness of the graphic response, until the desired design objective was achieved.



Fig. 4. Sameness-spread condition. No attractors, agents generated on random positions (left: cluster centroids; right: heatmap).

Case Study

The area selected as an application case for the simulation is the area of Piazza Leonardo Da Vinci, the location of the Politecnico di Milano. The space considered in the model covers an area of about 2 sq. km. The predefined attractors are: 1) the green area of Piazza Leonardo Da Vinci; 2) the main entrance of the Politecnico in Piazza Leonardo Da Vinci; 3) the building known as "Trifoglio" in Via Edoardo Bonardi; 4) the facade of the Faculty of Architecture in Via A. M. Ampère.

The OSM map was initially integrated with avenues not present in the original file. The Politecnico di Milano is here represented as a set of single buildings rather than as a single block. This choice makes it possible to consider as valid the internal paths usually used by teachers, students and staff.

The predefined attractors were chosen based on our familiarity with and professional knowledge of the place. The space of influence of an attractor was considered as the place of the points from which its point position is visible.

Once the simulation was launched, it was possible to compare the results with different parameters, the variables that influence the algorithm are the number of agents generated, their starting position and the presence of attractors to which they can turn their attention.

When interpreting maps, it should be considered that agents tend to use the shortest possible routes, while the users of a place may follow more meandering paths. This leads to transversal trends in building complexes and in drawing straight trajectories when agents intercept an attractor: Furthermore, the greater the distance of the starting points from the attractors, the greater the difficulty for the agents to intercept them, and consequently the greater the probability that the agents will remain distant by drawing globular geometries.



Fig. 5. Attractor-spread condition. Predetermined attractors, agents generated on random positions (left: cluster centroids; right: heatmap).

In simulations in which both the starting positions and the attractors are kept free (sameness-spread, fig. 4), the simulation produces a centrifugal distribution in which the agents tend to interact with each other, rather than with the environment. The areas of interaction are arranged far away from the spaces that in reality are devoted to social interaction; what we observe is the effect that buildings function exclusively as barriers, rather than as orientating elements. Since this condition is unrealistic, especially in relation to the setting of Piazza Leonardo Da Vinci, we decided to set the system to predetermined starting positions which are known to be important road and public transport junctions. In simulations where the starting positions are predetermined but the attractors are left undefined (sameness-gate), the agents steer their preferences along the road perimeter they travel in the exploration. As we can observe from fig. 5, the path travelled by the agents follows a logic of mutual interception, that is, the subgroups tend to search for each other because the various positions fall within their respective observation radii. Buildings are still considered as mere obstacles, and this translates into actions to circumvent barriers. This case can be interpreted as a routine of indifferent passage through the neighbourhood, this simulation was not considered consistent with the use of the app, and it was therefore agreed upon to fix the position of the attractors in subsequent simulations. In the simulations in which the presence of predetermined attractors and agents that arise on random positions (attractor-spread) has been established (fig. 6), the agents iden-

tify the empty aggregative space represented by Piazza Leonardo Da Vinci as the focus of interest. The geometry of the paths shows that the attractors constitute a centripetal field of influence. Based on these considerations, we set a fourth simulation mode by establishing predetermined positions for both attractors and agents. This further approach represents a system that is more consistent with the setting and the use of the area of Piazza Leonardo Da Vinci.

In simulations where the starting points have been predetermined and the attractors set (attractor-gate), the agents were concentrated in the area affected by the attractors both in positional and temporal terms. The proximity of two starting points to the attractor located on the Agraria building determined a positional imbalance of interactions in Piazza Leonardo Da Vinci. As can be observed in fig. 7 and fig. 8, the agents tend to move south towards Via Celoria, converging around the central part of the square. As the buildings of the Polytechnic stand between the starting points and the attractors, and in particular towards the group generated at Lambrate station, there is an intense use of the routes within the Campus as shortcuts to reach the objectives.



Fig. 6. Sameness-gate condition. No attractors, agents generated on predetermined positions (left: cluster centroids; right: heatmap).

Conclusions

The structuring and spatialisation of the agents' interactions has made it possible to develop numerous alternative scenarios and to compare in advance emerging behaviours that the app must be able to cope with. The process cannot be fully automated, but must be formed by a professional who with his expertise identifies actual attractors of the territory, in order to qualify the simulation.

In any case, the use of the described procedural strategy has proved to be effective as a stress test of the tools for collecting and representing the positioning data of agents. The repetition of simulations verified from time to time with the cartographic and statistical representation of the data allowed us to verify the quality of the basic constructs with respect to changes in user response dynamics. From the results of the investigations through simulation, interactive cartographic representations have been identified as a useful tool; namely, the possibility to independently activate different levels of information and to select specific information fields by selecting point elements. The simulation has proved to be a valid tool for anticipating possible technical problems, and effectively directing communication towards users and developers.

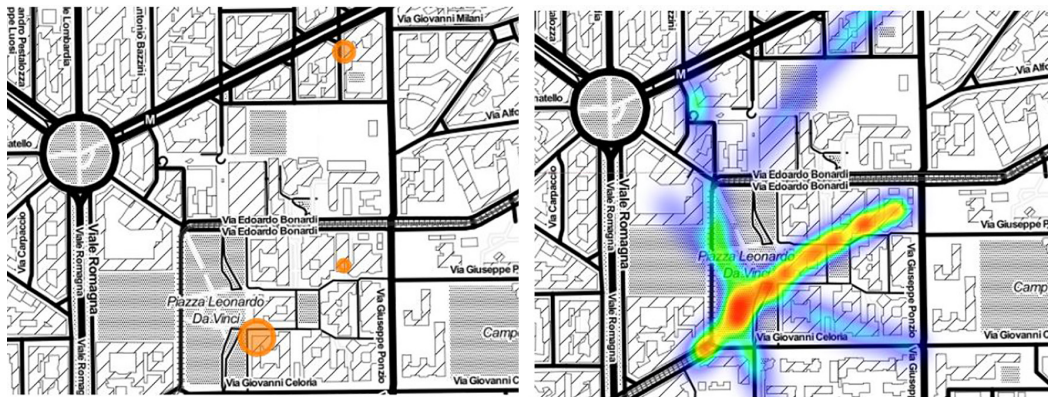


Fig. 7. Condition attractor-gate. Attractors on predetermined positions, agents generated on predetermined positions (left: cluster centroids; right: heatmap).

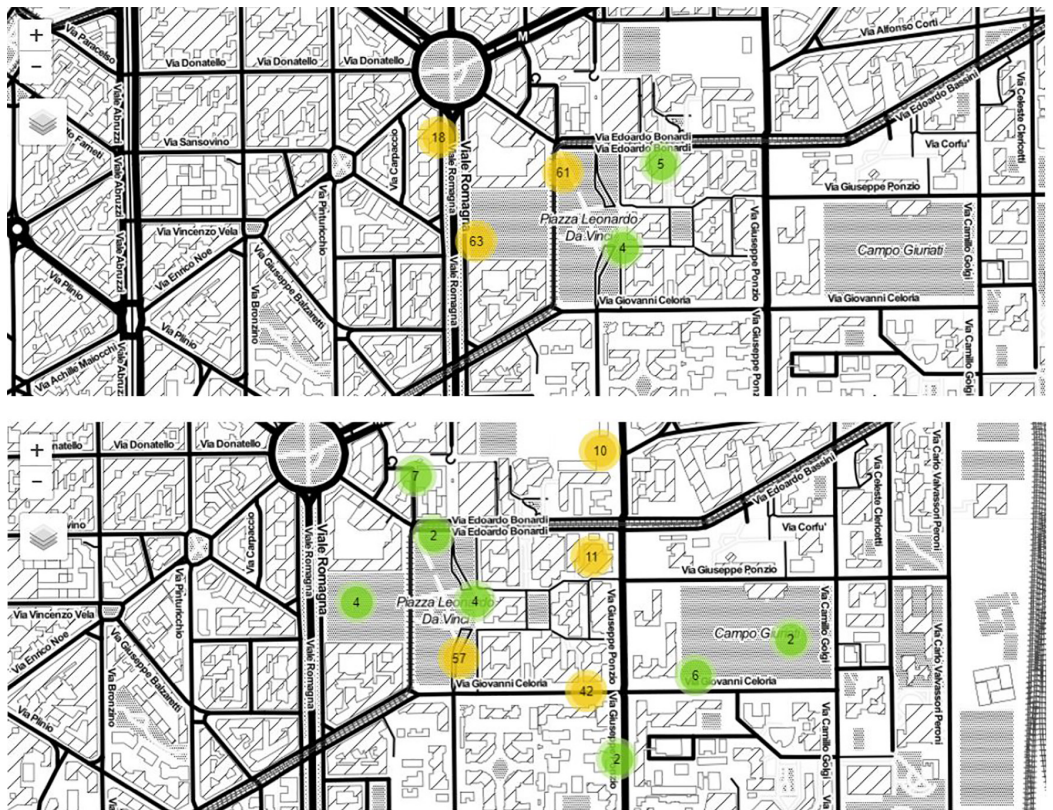


Fig. 8. Examples of aggregation of virtual agents in two different simulations, the numbers inside circles show the agents amount in a group.

Notes

- [1] APP AR4CUP to be launched in the market in April 2020.
- [2] 2.3. Clustering - scikit-learn 0.22.1 documentation. (2020). Scikit-learn.org. Retrieved 6 February 2020, from: <https://scikit-learn.org/stable/modules/clustering.html#>.
- [3] Stamen Maps. Maps.stamen.com: <http://maps.stamen.com/#terrain/12/37.7706/-122.3782>.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation program EIT Digital 2019 AR4CUP | Augmented Reality for Collaborative Urban Planning - The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.

References

- Atash Farhad (1994). Redesigning Suburbia for Walking and Transit: Emerging Concepts. In *Journal of Urban Planning and Development*, 120 (1), pp. 8-57.
- Aultman-Hall Lisa, Roorda Matthew & Baetz Brian W. (1997). Using GIS for Evaluation of Neighborhood Pedestrian Accessibility. In *Journal of Urban Planning and Development*, 123 (1), pp. 10-17.
- Batty Michael (2001). Exploring Isovist Fields: Space and Shape. In *Architectural and Urban Morphology. Environment and Planning B: Planning and Design*, 28 (1), pp. 123-150.
- Benedikt Michael L. (1979). To Take Hold of Space: Isovists and Isovist Fields. In *Environment and Planning B: Planning and Design*, 6 (1), pp. 47-65.
- Bielik Martin S. (2012). Parametric Urban Patterns: Exploring and integrating graph-based spatial properties in parametric urban modelling. In Achten Henri; Pavlicek Jiri; Hulin Jaroslav; Matejovska Dana (Eds.). *Digital Physicality – Proceedings of the 30th ECAADe Conference - Volume 1*, Czech Technical University in Prague, Faculty of Architecture (Czech Republic), 12-14 september 2012, pp. 701-708.
- Bonabeau Eric (2002). Agent-based modeling: Methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences of the United States of America*, 99 (suppl. 3), pp. 7280-7287.
- Davis Larry S., & Benedikt Michael L. (1979). Computational models of space: Isovists and isovist fields. In *Computer Graphics and Image Processing*, 11(1), pp. 49-72.
- Ester Martin, Kriegel Hans-Peter & Xu Xiaowei (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD-96 Proceedings*, pp. 226-231.
- Helbing Dirk, & Molnár Péter (1995). Social force model for pedestrian dynamics. In *Physical Review E*, 51(5), pp. 4282-4286.
- Macal Charles M. & North Michael J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference*, pp. 2-15.
- Pan Xiaoshan et al. (2007). A multi-agent-based framework for the simulation of human and social behaviors during emergency evacuations. In *AI & SOCIETY*, 22(2), 2007, pp. 113-132.
- Southworth Michael (2005). Designing the Walkable City. In *Journal of Urban Planning and Development*, 131(4), pp. 246-257.
- Spector Robert H. (1990). Visual Fields. In Walker H.K., Hall W.D., Hurst J.W. (eds.). *Clinical Methods: The History, Physical, and Laboratory Examinations*. 3rd edition. Boston: Butterworths, Chapter 116. pp. 565-572.
- Tufféry Stéphane (2011). *Data Mining and Statistics for Decision Making: Tufféry/*Data Mining and Statistics for Decision Making. Hoboken: John Wiley & Sons, Ltd.

Authors

Stancato Gabriele, Politecnico di Milano, gabriele.stancato@polimi.it
Piga Barbara E.A., Politecnico di Milano, barbara.piga@polimi.it

To cite this chapter: Stancato Gabriele, Piga Barbara (2020). La simulazione parametrica come strumento per informare la rappresentazione/ Parametric simulation as a tool for reporting representation. In Arena A., Arena M., Brandolino R.G., Colistra D., Ginex G., Mediati D., Nucifora S., Raffa P. (a cura di). *Connettere. Un disegno per annodare e tessere. Atti del 42° Convegno Internazionale dei Docenti delle Discipline della Rappresentazione/Connecting. Drawing for weaving relationships. Proceedings of the 42th International Conference of Representation Disciplines Teachers*. Milano: FrancoAngeli, pp. 829-846.