Presentation of a Strategy-Based Crowd Flow Dynamics Model and Its Application to Phenomenon Analysis

David Minjae Ko, Sunghoon Kim

Seoul Science High School

0. Abstract

Accurately modeling crowd flow dynamics (CFD) is critical for understanding and managing crowd behavior in complex environments. Existing CFD models often fall short in replicating realistic human movement, particularly in high-density or psychologically interactive scenarios. This study introduces a novel strategy-based CFD simulation framework to enhance the realism and adaptability of crowd flow simulations, with applications in urban planning and safety management. The framework consists of three components: (1) a CFD simulation built using the Discrete Element Method (DEM), integrating psychological and physical interactions among agents; (2) a real-world data transformation pipeline using Independent Instance Maps (IIM), Inverse Perspective Mapping (IPM), and agent tracking for converting video data into Bird's Eye View (BEV) spatial coordinates; and (3) parameter optimization through Dynamic Time Warping (DTW) and Bayesian Optimization to minimize discrepancies between real-world and simulated data. Simulations across four scenarios demonstrated realistic crowd behavior, with flexibility in adjusting parameters like crowd density and obstacle placement. Real-world data transformation achieved high accuracy though challenges were observed in occluded and complex terrain scenarios. Parameter optimization reduced the DTW distance between real and simulated data, indicating strong alignment. The proposed framework effectively replicates real-world crowd movements, validating its potential for practical applications such as disaster prevention, urban design, and event management. Future research will explore dynamic strategy adaptation using AI and improved multi-camera data integration for greater accuracy and applicability.

Keywords: Crowd Flow Dynamics, Pedestrian Flow, Discrete Element Method, Crowd Localization, Real-World Data Transformation, Simulation Framework.

1. Introduction

Simulation research generally refers to the field that leverages various mathematical and computational techniques to model and predict complex system behaviors. The goal of simulation studies is to enhance predictive accuracy through precise numerical analysis. For instance, in climate modeling, fluid dynamics, and structural analysis, researchers solve equations based on physical laws to forecast future states of systems. These models are highly effective when system behaviors are determined by specific physical laws.

Crowd Flow Dynamics (CFD) is a field that applies physical models to analyze the movement of crowds exhibiting fluid-dynamic patterns due to specific motivations or directions. In 2022, the tragic Itaewon crowd crush incident, known as the "Itaewon Disaster," occurred, caused by crowd turbulence in a narrow passage. Such incidents, common at concerts and events, underscore the need for further CFD research to understand dense crowd dynamics. CFD often relies on simulations to analyze crowd movements, yet these simulations present unique challenges compared to standard simulations. First, crowds consist of numerous individual agents, each operating through distinct decision-making processes and interacting with their environment to create complex movement patterns. Second, crowd behaviors are influenced not only by physical laws but also by social and psychological factors. For example, crowd movements in panic situations diverge markedly from typical behaviors and exhibit unpredictable patterns. As a result, existing CFD studies often focus on simulating crowd movements across different scenarios rather than validating the accuracy of simulations against real-world phenomena. Consequently, CFD simulations face inherent limitations when applied to predictive modeling.

This study aims to address these limitations by developing a "Strategy-Based Crowd Flow Model" framework that enhances CFD simulation validity through comparison with real-world crowd data, while also providing high adaptability. Here, the term "strategy" refers to the interactions among crowd members that cannot be fully explained by physical laws. The core idea is to create a framework where strategies can dynamically adjust, enabling optimized simulations. Specifically, this study involves the following processes to establish the framework:

A. Development of a Strategy-Based CFD Simulation: Incorporating the decision-making processes of individual agents to realistically replicate crowd movements. The Discrete Element Method (DEM) is employed to accurately model interactions among agents.

B. Conversion of Real-World Crowd Data: Using Crowd Localization to obtain crowd positions from images and applying Inverse Perspective Mapping (IPM) to reconstruct this data, allowing real-world data to correspond with the simulation.

C. Strategy Optimization through Data Comparison: Dynamic Time Warping (DTW) is used to compare simulation outcomes with real-world data, enabling optimization of simulation parameters based on this comparison.

Through this framework, the study aims to validate CFD simulations, allowing them to be applied to predictive modeling. Ultimately, this research seeks to expand the application of CFD across diverse fields, enhancing its practical utility.

2. Literature Review and Background

2.1. Crowd Flow Dynamics (CFD)

Crowd Flow Dynamics (CFD) refers to the research field that models and simulates crowd movements, which often exhibit irregular patterns based on crowd density, through physical models to analyze various phenomena. Traditional CFD models have encountered significant limitations, such as restricted degrees of freedom in simulating human movement strategies or an inability to capture realistic crowd dynamics. This study aims to develop a more adaptable simulation framework that provides enhanced validity. CFD research generally diverges into distinct types depending on simulation methodologies and objectives.

CFD simulation methods are broadly categorized into two approaches based on their representation of movement: the Discrete Model and the Continuous Model. The Discrete Model approach models crowd strategies by defining potentials or by creating specific rules related to human interaction, focusing intensively on agents' strategic responses to particular environments. Notably, research by Nam developed a CFD simulation using P*FLOW, a potential pathline method, to define environmental potentials and enable agent strategies based on these dynamics[1]. His additional studies explored correlations between traffic congestion and crowd strategies, such as situations involving multiple groups moving in opposite directions and the resulting degree of congestion[2].

The Discrete Model has advantages, including easier parameter design to understand and apply pedestrian route choices based on environmental factors and lower computational time complexity. However, its discontinuous representation of human movement often leads to a gap between simulated and real-life dynamics, presenting a fundamental limitation in accurately depicting environments. Figure 1 illustrates an implementation example of the Discrete Model.

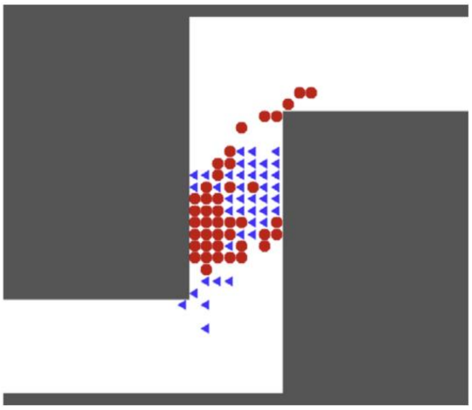


Figure 1: Implementation Example of the Discrete Model

The Continuous Model typically uses the Discrete Element Method (DEM) to simulate continuous movements by calculating forces and interactions between discretized human elements. Studies applying DEM to CFD focus on creating continuous movement rather than specific agent strategies, aiming to observe crowd flow adaptations under conditions similar to real-life scenarios. For instance, Langston, Masling, and Asmar modeled individuals as three circles within a DEM framework to investigate optimal escape patterns and characteristics in various environmental situations[3]. For this study, realism in simulation was critical, so the Continuous Model was selected despite its higher time complexity for improved environmental accuracy.

CFD research typically pursues one of two primary objectives. The first objective is to identify strategies that facilitate crowd flow under conditions of crowd turbulence, congestion, or varied environmental factors. This approach is frequently adopted by studies employing the Discrete Model, as well as those like Xu's research on crowd strategies in congested environments[4]. The second objective focuses on achieving realistic movement simulations that allow for post-incident analysis. This approach emphasizes obtaining a high degree of macroscopic similarity between the model and real-world crowd movements and often involves Continuous Models, where dynamic strategy adjustments are more challenging.

Various methodologies have been used to incorporate human strategies within CFD. The direct implementation approach involves replicating human movement patterns, while other methods utilize genetic algorithms or reinforcement learning to allow agents to learn strategies that prevent collision while reaching target destinations. Sagredo-Olivenza's study applied genetic algorithms in CFD to achieve complex crowd strategies in a 3D simulation with minimal computation[5]. Additionally, a study by Lee and Won implemented Deep Reinforcement Learning (Deep RL) within CFD, where a policy function generates actions based on state information, such as crowd density and current speed[6]. Here, rewards were structured around factors such as distance from the target, collision avoidance, and smoothness of movement. Figure 2 illustrates the method by which the agent's state information is acquired in Deep RL simulations.

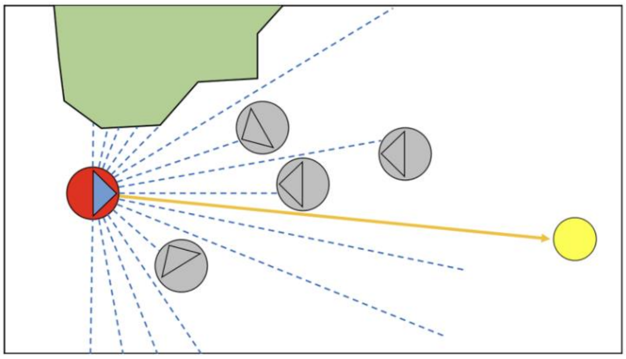


Figure 2: Application of Deep RL in CFD Simulation

This study diverges from conventional CFD approaches by proposing a strategy-based simulation framework that dynamically adjusts strategies, thereby allowing for the quantitative validation and optimization of crowd movement realism. Unlike previous CFD simulations, which often relied on subjective assumptions or estimations of realism, this research offers a novel, validity-focused approach by quantifying realism to optimize strategies—setting it apart as a more rigorous and adaptable CFD methodology.

2.2. Discrete Element Method (DEM)

In CFD simulations that use the Continuous Model, individual agents are modeled as particles to which physical laws are applied. A commonly used technique in this context is the Discrete Element Method (DEM), which analyzes interactions and movements of discrete particles by simulating particle contact and collision dynamics[7]. DEM is broadly categorized into two approaches: Soft Sphere DEM and Hard Sphere DEM[8]. Soft Sphere DEM permits particle overlap and calculates repulsive forces based on the degree of overlap, making it suitable for rapid calculations of contact and deformation forces. In contrast, Hard Sphere DEM does not allow particle overlap; instead, it uses restitution coefficients to model momentum changes after collisions. This study adopts the Soft Sphere DEM method for its efficiency in calculating contact forces and deformation. Figure 3 compares the mechanisms of Soft Sphere DEM and Hard Sphere DEM.

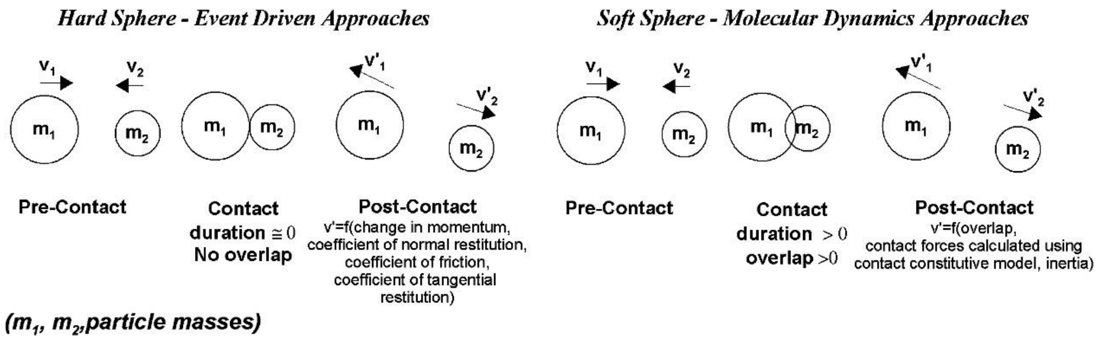

Hard Sphere - Event Driven Approaches
Soft Sphere - Molecular Dynamics Approaches
v1
v2
m1
m2
v'1
v'2
Pre-Contact
Contact
duration ≅ 0
No overlap
Post-Contact
v'=f(change in momentum, coefficient of normal restitution, coefficient of friction, coefficient of tangential restitution)
Pre-Contact
Contact
duration > 0
overlap >0
Post-Contact
v'=f(overlap, contact forces calculated using contact constitutive model, inertia)
(m1, m2,particle masses)

Figure 3: Comparison of Hard Sphere DEM (left) and Soft Sphere DEM (right)

Given that DEM calculates interactions between all contacting particles, an efficient algorithm for detecting contact particles is essential. This study employs the Linked-List Cell Method (LCM), which partitions the computational domain based on a grid and selects particles per cell to calculate contact status[9]. Figure 4 illustrates the operation of LCM, contrasting a scenario where all particle pairs are compared versus a grid-partitioned approach using LCM.

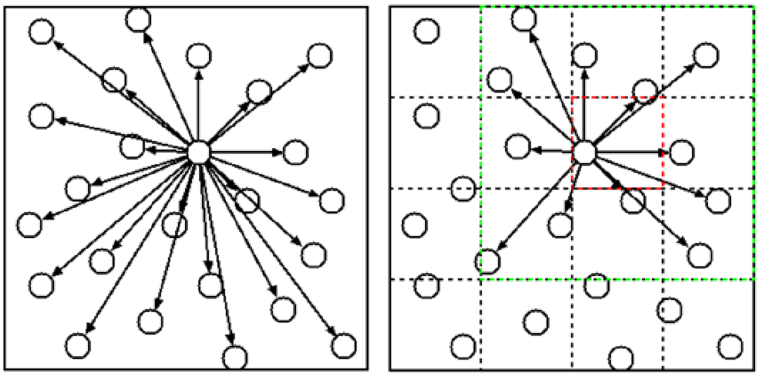


Figure 4: Comparison of All-Pairs Calculation (left) and Grid Partitioning with LCM (right)

2.3. Verlet Integration

Verlet Integration is a numerical method commonly used to integrate Newton's equations of motion, and it is frequently applied in simulations involving molecular dynamics and computer graphics to calculate the trajectories of individual particles[10]. Compared to simpler methods like Euler integration, Verlet Integration provides enhanced numerical stability for physical systems without increasing execution time. This method calculates trajectories with greater accuracy since its error term is of the fourth order, whereas Euler integration has a first-order error and Runge-Kutta has a second-order error[11].

Verlet Integration has also been extended to accommodate scenarios where the time interval, Δt, is not constant. In this study, we employed the Velocity Verlet method, which is widely used for its ease of implementation. Velocity Verlet improves upon the standard Verlet method by calculating velocity alongside acceleration, thereby addressing the method's need for initial values in approximation[12]. The particle motion is derived through the following iterative steps with interval Δt:

(1) $v_{t+\frac{1}{2}\Delta t} = v_t + \frac{1}{2} a_t \Delta t$

(2) $x_{t+\Delta t} = x_t + v_{t+\frac{1}{2}\Delta t} \Delta t$

(3) $v_{t+\Delta t} = v_{t+\frac{1}{2}\Delta t} + \frac{1}{2} a_{t+\Delta t} \Delta t$

First, partial velocity is calculated by integrating the current acceleration in step (1). Then, using this intermediate velocity, the displacement term is added in step (2). Finally, the new acceleration, derived from the potential after updating displacement, is used to finalize the velocity adjustment in step (3). By iterating this process for individual agents within the crowd using Velocity Verlet, we simulate the overall movement dynamics of the crowd.

2.4. Crowd Localization

Crowd Localization is a process used in the analysis of crowd movements that involves identifying the positions of individual agents within an image. Unlike Crowd Counting, which typically provides an aggregate count of individuals, Crowd Localization allows for more precise crowd analysis by recognizing each agent individually. Large-scale crowd datasets are typically annotated through various methods depending on the type of data labeling: Total Count Annotation, which records only the number of people; Dot Annotation, which distinguishes individual heads with points; and Box Annotation, which creates bounding boxes around individual agents. Because Crowd Localization requires detailed data at the Dot Annotation or Box Annotation level, its development has been slower than that of Crowd Counting.

Given the close relationship between Crowd Localization and Crowd Counting, understanding Crowd Localization involves a brief review of recent advancements in Crowd Counting. Many modern Crowd Localization studies have moved beyond traditional feature-based analysis and now leverage deep learning models, such as CNNs. Some approaches use Density Maps, representing crowd density on a per-pixel or patch basis, or generate Segmentation Maps directly for Crowd Counting[13]. However, these methods are primarily focused on counting or estimating general density, making them less suited for precise Crowd Localization applications.

Crowd Localization approaches are divided into two main types: Detection-Based Method and Heuristic Method. Detection-Based Method predicts the positions of individuals using bounding boxes through object recognition. Advanced models, such as R-CNN (Regions with Convolutional Neural Networks) and OverFeat with LSTM layers for enhanced individual recognition, fall into this category. However, these methods often suffer performance drops when individual agents overlap or when there is significant variation in image quality and scale. Despite attempts to improve bounding box precision, such as through the YOLO model, the Detection-Based approach remains limited in its ability to pinpoint exact locations in dense crowds[14].

Heuristic Method is typically used with high-resolution data and employs Density Maps or Segmentation Maps from Crowd Counting models to directly perform Crowd Localization. Heuristic methods often rely on Gaussian assumptions to identify the highest confidence position within a local neighborhood as the individual's head, or use algorithms to reconstruct the Density Map. Although this approach can facilitate easier location determination compared to Detection-Based Models, it faces challenges in reliably distinguishing individual agents and remains vulnerable to variations in crowd density and size. To address these challenges, a study proposed the Independent Instance Map (IIM), which includes a Binarization Module (BM) to convert Density Maps into IIMs[15]. The BM applies localized thresholds to create a Threshold Map within the Confidence Map, which then undergoes binarization to extract individual agents. Figure 5 illustrates the structure of this model, which uses pretrained models like VGG-16 + FPN and HRNet-W48 as confidence predictors.

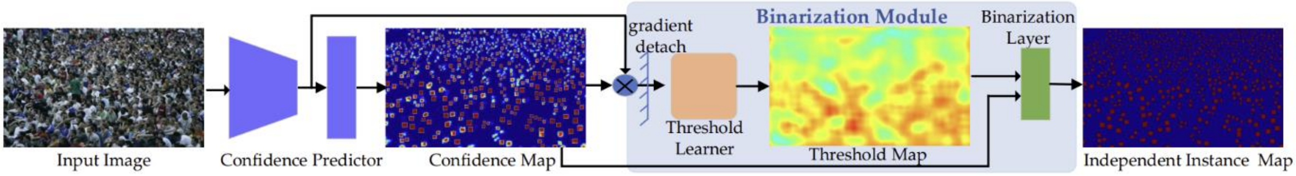


Figure 5: Crowd Localization Framework with IIM Conversion using Binarization Module

In this study, we include a process to capture real-world crowd data from video and convert it into a format applicable to simulations. For CFD research applications, this data transformation process must be effective for high-density and size-variant crowd data. Consequently, we adopted the IIM model from the NWPU-Crowd dataset, recognized as a State-of-the-Art technique with computational efficiency suitable for video expansion, to achieve optimal performance.

2.5. Inverse Perspective Mapping (IPM)

Inverse Perspective Mapping (IPM) is a technique that transforms a 2D image captured by a camera into a Bird's Eye View (BEV) representation, revealing true spatial coordinates. This technology is widely applied in autonomous driving systems to correct perspective distortions introduced during image capture, thereby reinterpreting the 2D image as a consistent 3D scene in BEV format. In this study, IPM is essential for transforming real-world crowd data into spatial data that can be applied within a simulation environment. IPM begins with calculating the perspective projection matrix from World coordinates to Image coordinates, using a homogeneous coordinate system to facilitate translation and rotation operations.

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R}|\mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \mathbf{H} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

In this matrix equation, X, Y, and Z represent spatial coordinates in the World coordinate system. Matrix $(\mathbf{R}|\mathbf{t})$ accounts for rotation and translation, transforming World coordinates into the Camera coordinate system, while matrix $\mathbf{K}$ includes the intrinsic parameters of the Camera. The product of these matrices, denoted $\mathbf{H}$, is known as the Homography Matrix. Here, s represents the z-coordinate in the Camera coordinate system, acting as a scaling factor for the final result. Figure 6 illustrates the derivation process and relationships between coordinate systems.

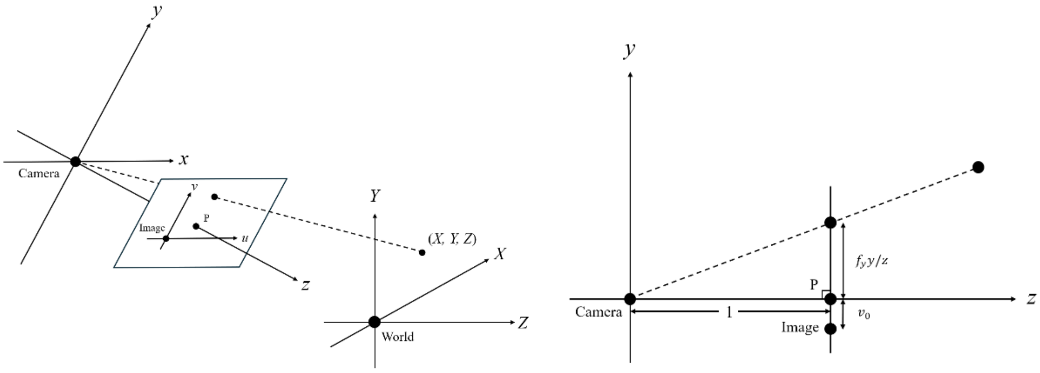


Figure 6: Relationship Between Coordinate Systems and Perspective Projection Calculation

IPM, however, requires the reverse operation to convert an image in the Camera coordinate system into BEV data. This is achieved using the Backward Method, which accesses the original image pixel values from each pixel in the BEV data, ensuring all pixels in the BEV image are populated.

IPM translates 3D spatial data from the World coordinate system into a 2D image, but in reversing this projection, there is information loss due to dimensional reduction. To counteract this, assumptions are made about object height, with autonomous systems typically assuming $Z = 0$ for road surfaces. However, because our study involves Crowd Localization focusing on individuals' heads, a height assumption of $Z = 1.7$ m, reflecting the average height in South Korea, is applied to IPM.

Given camera calibration data, the Homography Matrix $\mathbf{H}$ can be calculated directly from the above equation. In this study, however, no calibration data is provided, as is common with crowd datasets like CCTV footage, where ground surfaces remain fixed while individual agents move. Here, Direct Linear Transformation (DLT) enables Homography estimation without calibration data[16]. DLT accepts pairs of World coordinates (X, Y, Z) and corresponding Image coordinates (u, v), provided as (X_i, Y_i, Z_i) and (u_i, v_i) for multiple points. Rewriting the above equation yields relationship equations for each pair, expressed as:

$$\begin{cases} -X_i h_{11} - Y_i h_{12} - Z_i h_{13} - h_{14} + u_i(X_i h_{31} + Y_i h_{32} + Z_i h_{33} + h_{34}) = 0 \\ -X_i h_{21} - Y_i h_{22} - Z_i h_{23} - h_{24} + u_i(X_i h_{31} + Y_i h_{32} + Z_i h_{33} + h_{34}) = 0 \end{cases}$$

By compiling the equations for all pairs, they form a matrix equation in the format:

$$\mathbf{Ah} = \begin{pmatrix} -X_1 & -Y_1 & -Z_1 & -1 & 0 & 0 & 0 & 0 & u_1X_1 & u_1Y_1 & u_1Z_1 & u_1 \\ 0 & 0 & 0 & 0 & -X_1 & -Y_1 & -Z_1 & -1 & v_1X_1 & v_1Y_1 & v_1Z_1 & v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -X_n & -Y_n & -Z_n & -1 & 0 & 0 & 0 & 0 & u_nX_n & u_nY_n & u_nZ_n & u_n \\ 0 & 0 & 0 & 0 & -X_n & -Y_n & -Z_n & -1 & v_nX_n & v_nY_n & v_nZ_n & v_n \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{34} \end{pmatrix} = 0$$

By applying Singular Value Decomposition (SVD) to matrix $\mathbf{A}$, the least-squares solution vector $\mathbf{h}$ can be derived, yielding the Homography Matrix $\mathbf{H}$. Since $\mathbf{H}$ contains 11 independent elements and each pair provides two linear equations, a minimum of six independent point pairs are required to construct $\mathbf{H}$.

2.6. Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is an algorithm used to assess the similarity between two time series by finding an alignment path that minimizes their cumulative distance. When comparing time series data using Euclidean distance on a same-time basis, slight temporal variations or oscillations can cause significant differences in similarity values, even if overall patterns are alike. DTW addresses this issue by allowing for comparisons across different time points, matching similar elements across non-aligned time steps. Figure 7 provides a visual comparison of time series alignment using Euclidean distance and DTW, showing how DTW effectively identifies matching patterns even in misaligned time series.

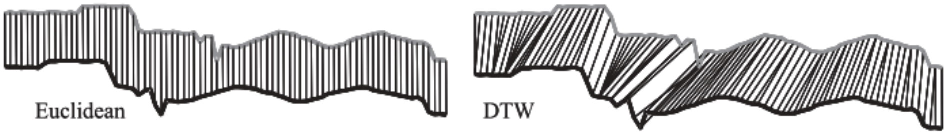


Figure 7: Time Series Analysis Using Euclidean Distance and DTW

To calculate the optimal matching between time series R_i and Q_i, we apply the following formula:

$$\mathrm{dist}\left(r_i^k, q_i^j\right) = \left(r_i^k - q_i^j\right)^2$$

$$\Psi(k,j) = \mathrm{dist}\left(r_i^k, q_i^j\right) + \min\{\Psi(k-1,j-1), \Psi(k-1,j), \Psi(k,j-1)\}$$

Here, $\Psi(k,j)$ represents the minimum cumulative distance up to the (k,j)-th elements of time series R_i and Q_i, with r_i^k and q_i^j denoting the k-th and j-th elements of R_i and Q_i, respectively[17].

In this study, we use 2D crowd position data obtained from both real and simulated environments, making use of a Multi-Dimensional Time Series (MDT) approach. In MDT, two methods for DTW-based similarity measurement exist: Independent DTW ($\mathrm{DTW_I}$) and Dependent DTW ($\mathrm{DTW_D}$).

1. **$\mathbf{DTW_I}$** analyzes each dimension of the MMM-dimensional time series RRR and QQQ independently and sums them, defining DTW as:

$$\mathrm{DTW}(R,Q) = \sum_{m=1}^{M} \mathrm{DTW}(R_m, Q_m)$$

2. **DTW_D** assumes dependence across all dimensions, defining distance as:

$$\mathrm{dist}\left(r^k, q^j\right) = \sum_{m=1}^{M} \left(r_m^k - q_m^j\right)^2$$

Because the crowd location data is given as 2D coordinates, which are inherently dependent, this study applies $\mathrm{DTW_D}$ as a measure of movement similarity between real and simulated crowd data[18].

2.7. Bayesian Optimization

In the implementation of strategies for this study, numerous adjustable parameters are involved, and optimizing these parameters requires iteratively adjusting them to minimize the Dynamic Time Warping (DTW) distance by comparing real crowd data with simulated data. Bayesian Optimization is a widely used methodology for optimizing nonlinear and computationally expensive functions. Its primary feature is a probabilistic approach that models the uncertainty of the objective function, enabling efficient exploration of the parameter space. The core concept of Bayesian Optimization is to use a Gaussian Process (GP) to build a prior distribution over the objective function and employ an Acquisition Function to determine the next point for evaluation, thereby progressing toward optimization[19].

Alternative optimization methods include Grid Search and Random Search. Grid Search evaluates every possible combination within a predefined parameter grid, which can be effective for low-dimensional search spaces but becomes computationally prohibitive as dimensionality increases. Additionally, the fixed grid may overlook critical parameter values. Random Search, which samples random parameter values based on a predefined distribution, is often more efficient than Grid Search but remains limited in high-dimensional search spaces.

Bayesian Optimization, in contrast, leverages prior evaluation results to guide the search, requiring far fewer evaluations to locate optimal parameters compared to Grid or Random Search. In this study, the DTW distance, which serves as a metric for movement similarity, is computationally expensive as it involves running the full simulation process. By employing Bayesian Optimization, significant reductions in computational cost are expected.

3. Methods

3.1. Tools and Materials

The entire framework was implemented in a virtual environment based on Python 3.8.19 and tested in a CPU-only environment without GPU acceleration. The key packages and modules used in this virtual environment are listed in Table 1.

Table 1. Key Packages and Modules

Name	Version	Purpose
numpy	1.23.5	Matrix operations
pandas	2.0.3	Data management
matplotlib	3.7.5	Data and result visualization
opencv-python	4.10.0.84	Image and video visualization
ffmpeg-python	0.2.0	Image and video management
imageio	2.34.2	GIF management
moviepy	1.0.3	Video management
fastdtw	0.3.4	DTW application
scipy	1.10.1	Linear algebra operations
torch	2.3.1	IIM model implementation
scikit-optimize	0.10.2	Bayesian optimization

The IIM model was trained using the NWPU-Crowd dataset, which contains 5,109 crowd images, with annotations provided by prior researchers. To validate the framework's ability to accurately transform image data into video-based real-world crowd data, the "Indoor and Outdoor Crowd Characteristics Data" provided by the National Information Society Agency (NIA) in South Korea was utilized. This dataset, available on the AI Hub (aihub.or.kr), includes 228,195 crowd images derived from 400 videos.

To test the complete framework, a sample analysis scenario was established. This scenario focuses on Tokyo's Shibuya Scramble Crossing, a well-known example of complex crowd flow due to mixed pedestrian movement in multiple directions. For this purpose, three videos of the crossing were collected from iStock, with stock video IDs 1474927019, 1167282548, and 1167927700.

3.2 Development Process

3.2.1 Development of a Strategy-Based CFD Simulation

Implementation of the Physical Model

The CFD model aims to realistically simulate crowd movements based on the Discrete Element Method (DEM). The model includes two primary elements: crowds (Humans) and obstacles (Obstacles). Crowds are represented as discrete elements within DEM, each characterized by scalar attributes (e.g., radius, mass, speed) and vector attributes (e.g., position, velocity, acceleration, force, and directional movement goals). For each frame, the Linked-List Cell Method (LCM) is used to detect neighboring agent pairs, then calculate forces based on psychological and physical interactions. The movements of agents are updated using the Verlet

Integration algorithm.

Obstacles are treated as immovable structures, represented as polygons with assumed infinite mass relative to the crowd elements. Various model parameters allow for optimizing and adapting the simulation. *Figure 8* shows the relationship between crowds and obstacles as implemented through DEM.

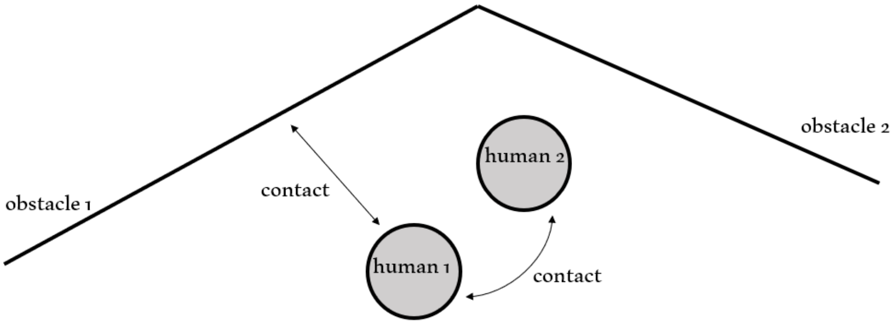


Figure 8. Relationship Between Crowds and Obstacles in the DEM-Based Model

The simulation process repeats the following steps: detecting neighboring pairs, calculating psychological and physical interactions, and updating velocities and positions via Verlet Integration. To avoid the time complexity $O(N^2)$ in computing distances for all agents and obstacles, LCM is applied to reduce complexity to $O(N)$. LCM classifies any two agents within a distance shorter than a parameterized factor χ of their combined size as neighbors. Psychological interactions are based on strategic movement directions, aiming to align each agent's movement with their intended path, considering obstacles and neighboring agents.

The psychological interaction is controlled by a strategy function $Q(s)$, where s represents the state, providing the agent's target direction and neighbor list. Given the state s, $Q(s)$ returns an intended direction a. The resulting velocity vector is determined by adding a force proportional to the difference between the agent's current and target velocities, representing the agent's motivation. Figure 9 illustrates this approach, with f_m as a proportional parameter in the model.

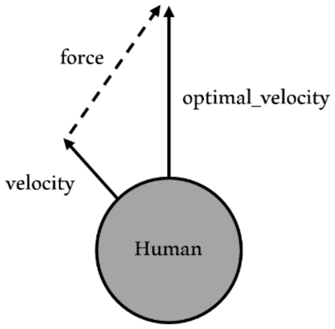


Figure 9. Implementation of Psychological Interactions for Crowd Agents

Physical interactions involve calculating inelastic collisions between agents or between agents and obstacles, with force magnitudes proportional to overlap depth as in the Soft Sphere DEM. The proportionality factor f_p and elasticity coefficient k serve as model parameters. Each simulation step concludes by updating agent positions and velocities, resetting accelerations and forces.

Incorporation of Strategy

To simulate realistic human behavior, we defined a plausible crowd movement strategy that manages psychological interactions between crowd agents and obstacles. Agents treat other agents or obstacles within their line of sight as impediments. If no obstacles are nearby, agents move toward their target direction. When an obstacle is detected, agents attempt to avoid it by rotating toward an alternate path, adjusting their movement more if the obstacle is closer or lies directly ahead. This movement is modeled by the strategy function and consists of two phases:

1. **Evaluation of Obstacle Impact**: Each obstacle's relative distance and direction determine how much it impedes the agent's target movement.

2. **Determination of Movement Direction**: The total obstacle impact is used to finalize the agent's directional movement.

Each agent's neighbors influence its direction proportionally to their distance d and angle θ. The rotation degree r correlates negatively with both d and θ, with any neighbor outside a $|\theta| \leq \pi/2$ range being ignored. The rotation degree r is computed as:

$$r = f(d, \theta) = \frac{\text{sign}(\theta)}{\alpha|\theta| + 1} \cdot d^{-2}$$

where α represents the sharpness of the function curve, as shown in Figure 10.

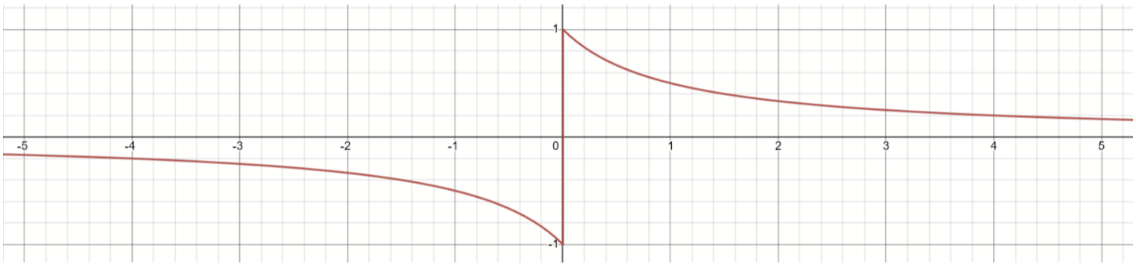


Figure 10. Functional Graph for Determining Agent Rotation Degree

With neighbors (d_i, θ_i), the total rotation is given by:

$$r = \sum_{i=1}^{N} f(d_i, \theta_i)$$

Since agents' actual rotation is constrained within $(-\pi/2, \pi/2)$, an activation function normalizes r:

$$\text{activation}(r) = \text{sign}(r) \cdot \left(\frac{\pi}{2} - \frac{1}{\beta|r| + 2/\pi}\right)$$

where β adjusts the function steepness, as illustrated in Figure 11.

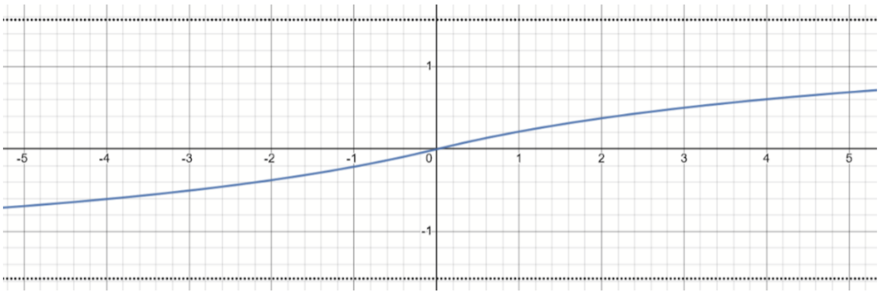


Figure 11. Activation Function Graph

The strategy function outputs a modified direction by rotating the initial target direction by $\text{activation}(r)$.

Simulation Testing in Various Scenarios

To verify that the simulation accurately depicts crowd movement as intended, various scenarios were established and tested, including:

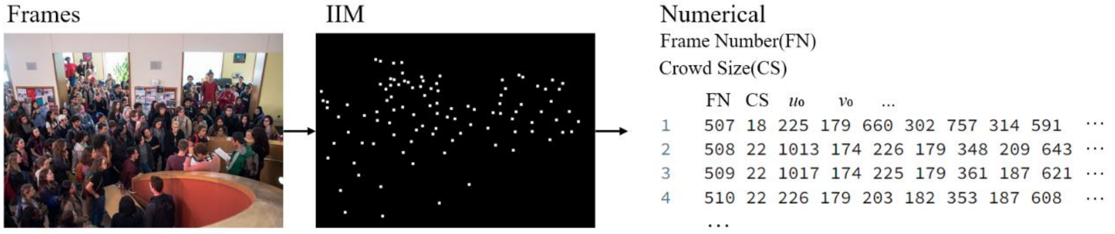
- **Straight Path**: A single-lane scenario with a uniform road layout.
- **Narrowed Straight Path**: A single-lane path that narrows mid-way, increasing crowd turbulence potential.
- **Intersection and Crosswalks**: Scenarios with multidirectional crowd flows simulating real-world intersections.

These scenarios enabled evaluation of the framework's capacity to simulate realistic crowd dynamics.

3.2.2 Transformation of Real-World Crowd Data

Application of the IIM Model and Crowd Localization

To convert real-world crowd data into Bird's Eye View (BEV) format, the initial step involves applying the Independent Instance Map (IIM) model to obtain the position of each individual within the crowd from image data. As this study uses video data, the model is applied to each frame, converting the crowd count and position data from the IIM output into numerical format. Figure 12 illustrates this process and the format of the numerical data obtained. It is important to note that the location data is presented as a simple set of points without information to individually distinguish each agent, as the IIM model does not support such differentiation.



```
     FN  CS  u₀   v₀  ...
1    507 18 225 179 660 302 757 314 591  ···
2    508 22 1013 174 226 179 348 209 643 ···
3    509 22 1017 174 225 179 361 187 621 ···
4    510 22 226 179 203 182 353 187 608  ···
     ...
```

Figure 12. IIM Model Application Process and Data Format

Homography Matrix Calculation and IPM Application

To compute the Homography Matrix H required for Inverse Perspective Mapping (IPM), Direct Linear Transformation (DLT) was used. Corresponding point pairs for DLT were manually collected from images, with world coordinates determined using actual measurements. Approximately 10 point pairs were selected for each dataset to calculate H, assuming a pinhole camera model to disregard lens distortion. To ensure independence, datasets included at least one point with $Z \neq 0$. The computed Homography Matrix H was applied to each frame's crowd position data to transform it into BEV format, yielding world-coordinate positions for each individual in the BEV data.

Agent Tracking and Kalman Filter Application

To properly separate individual agents from the BEV data, a tracking algorithm was implemented to maintain continuity in each detected point across frames. This assignment problem is similar to linking individual agents between frames based on spatial proximity, with Euclidean distance serving as the cost metric for matching points. Although algorithms such as the Hungarian Algorithm minimize the total cost sum, a Greedy Algorithm was chosen here to prioritize natural movement continuity over purely minimizing total cost. The algorithm follows these steps:

1. **Initialization**: Each point in the first frame is assigned a unique ID, creating a new track.
2. **Cost Matrix Calculation**: For each frame, a Cost Matrix is generated by computing the Euclidean distance between points in the current and previous frames.
3. **Minimum Cost Matching**: Points are matched in order of increasing cost. Matches with costs exceeding a set threshold are disregarded, and unmatched points are treated as new agents.
4. **Track Updates**: Matched points are added to existing tracks. Unmatched points close to boundaries initiate new tracks, while tracks without matches over a defined frame threshold are finalized.

Additionally, Missing Frames are managed to handle instances where an agent is briefly undetected, reducing false negatives. Linear interpolation fills in missing data for these frames, while short-lived tracks are removed to prevent false positives. Figure 13 provides a schematic of the tracking algorithm.

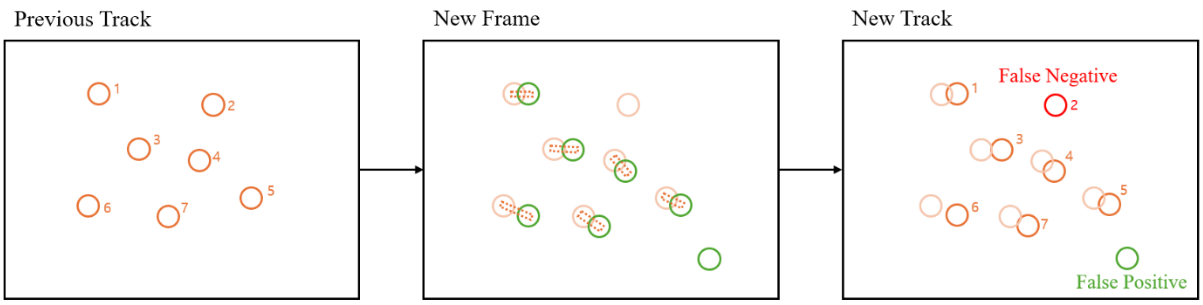


Figure 13. Agent Tracking Algorithm Schematic

To minimize noise introduced during tracking, a Kalman Filter was applied to smooth the resulting trajectories, as it is commonly used in dynamic systems for noise correction.

Testing on Various Real-World Crowd Data

To verify the suitability of the framework's transformation process for real-world crowd data, the "Indoor and Outdoor Crowd Characteristics Data" introduced earlier was used as a test case. For DLT, outdoor videos where real-world measurements could be collected were selected. The videos used for testing are listed in Table 2.

Table 2. Test Video List for Real-World Crowd Data Transformation

Location	Video Number
Shinsan Park	006, 007, 009, 010, 011, 012, 014, 015, 016
Oedo Elementary	110, 111, 112, 114, 086, 088
Ildo Sports Park	117, 139, 140, 141, 142, 143, 145, 146, 148, 149, 150, 151, 153, 155, 156, 157, 158, 083
Jeju Tapdong Plaza	056, 063, 064

Performance evaluation was conducted qualitatively by visualizing results from the listed videos and quantitatively by recording the precision, recall, and computation time per frame of the IIM model during Crowd Localization.

3.2.3 Strategy Optimization Through Data Comparison

Designing the Movement Similarity Measurement Method

A key aspect of this study is optimizing the simulation strategy by comparing real crowd data with simulated data. To achieve this, a movement similarity measurement method was designed based on Dynamic Time Warping (DTW). As mentioned in prior research, DTW is a robust tool for comparing time-series data by aligning sequences with varying time scales. In this study, both real and simulated crowd data were converted into time-series data, then DTW was applied to measure the similarity between them. The frames were divided into segments, and time variations were analyzed within each segment based on initial conditions, allowing for a quantitative comparison between real and simulated data.

Optimization of Strategy Parameters

The similarity scores obtained from DTW were then used to optimize the parameters of the simulation strategy, with Bayesian Optimization applied to find parameter settings that minimized DTW distance. By iteratively adjusting parameters to achieve minimal DTW distance, this optimization process enhanced the realism of the simulation, enabling it to more accurately reproduce real crowd movement. This approach not only improved the validity of the simulation when based on real-world data but also established a foundation for optimizing response strategies across various crowd scenarios.

3.2.4 Application of an Example Scenario: Shibuya Scramble Crossing, Tokyo

As a test scenario for the framework, we selected Tokyo's Shibuya Scramble Crossing, an iconic location known for its complex, multidirectional crowd flow, making it an ideal environment for validating the methods developed in this study. In this scenario, each individual's movement direction and speed were calculated and applied to the simulation strategy. The movement patterns obtained from real data were compared with those in the simulation, verifying the effectiveness of the developed crowd flow simulation framework in complex crowd dynamics. This process also demonstrated the improvement in simulation validity based on real-world crowd data.

The step-by-step application of the simulation framework to the Shibuya crossing scenario is as follows:

1. **Data Collection**: Real crowd data was collected in video form from recordings at the Shibuya Scramble Crossing.

2. **Application of the IIM Model and Crowd Localization**: The IIM model was applied to the collected data to identify individual locations, converting positional information from the images into numerical data.

3. **Application of IPM and Individual Agent Separation**: Homography Matrix H was calculated using specific points within the images, and IPM was applied. The agent separation algorithm then transformed the real crowd data to match the simulation environment.

4. **Application of the Strategy-Based CFD Simulation**: Based on the transformed data, basic directional and velocity information was generated for each agent, enabling a simulation that closely replicated the real environment.

5. **Data Comparison and Optimization Using DTW**: Each frame segment was divided, and DTW was used to compare simulation results with real crowd data within each segment. Bayesian Optimization was applied to maximize similarity by fine-tuning the simulation strategy's parameters.

Through these steps, the framework successfully simulated the complex crowd dynamics of the Shibuya Scramble Crossing, providing strong evidence of the simulation framework's effectiveness and validity. Figure 14 depicts the bustling scene at the Shibuya crossing.

Figure 14. Crowded Scene at Shibuya Scramble Crossing, Tokyo

4. Results

4.1 Strategy-Based Simulation with DEM

To verify that the simulation accurately depicted crowd movement, various scenarios were tested. Figure 15 shows the simulation result for a basic straight-road scenario. Figure 16 presents results for a narrow road scenario, which has a high potential for crowd turbulence. Figure 17 shows a simulation of a crossroads with multi-directional movement, while Figure 18 demonstrates a one-way, multi-path scenario. The test conditions were designed to reflect key factors identified in the Itaewon disaster, including large crowd sizes, diverse movement directions, and obstacles within narrow alleys.

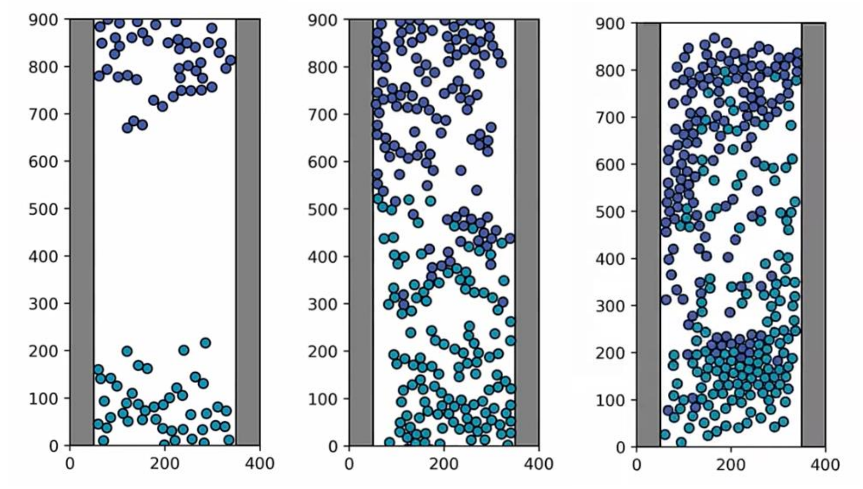


Figure 15. Basic Straight Road Scenario

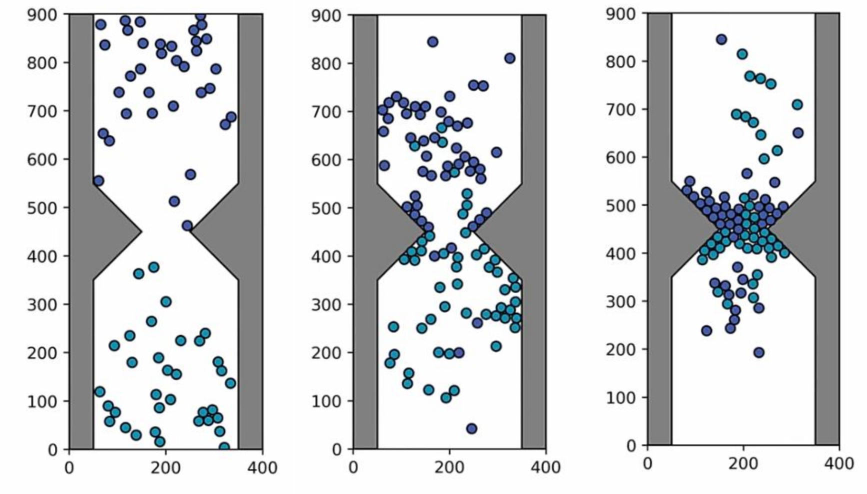


Figure 16. Narrowed Straight Road Scenario

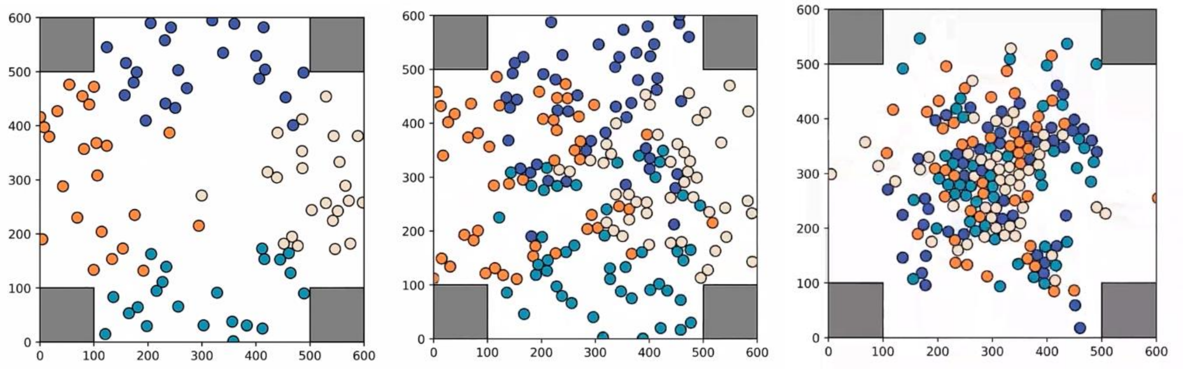


Figure 17. Crossroads Scenario

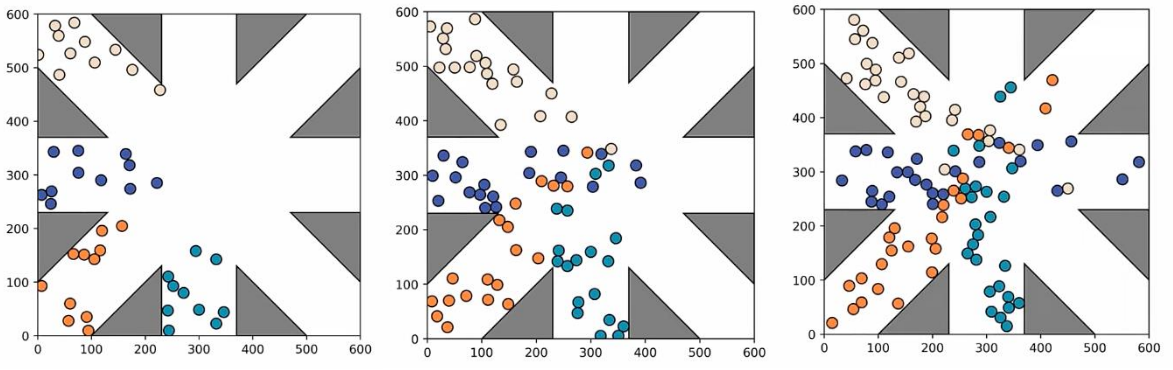


Figure 18. Multi-Path One-Way Scenario

4.2 Transformation of Real-World Crowd Data

4.2.1 Application of IIM Model and Crowd Localization

Table 3 summarizes the average processing time per frame, precision, and recall achieved by the IIM model when applied to the test videos. Each image was processed at the pixel level, with processing times averaging 6 to 8 seconds per frame, precision ranging from 0.8 to 0.9, and recall averaging around 0.8.

Table 3. IIM Model Performance by Location

Location	Average Processing Time (s)	Average Precision	Average Recall
Shinsan Park	6.89	0.922	0.874
Oedo Elementary	7.33	0.863	0.827
Ildo Sports Park	8.56	0.842	0.832
Jeju Tapdong Plaza	7.96	0.913	0.844

4.2.2 Testing Transformation Process Across Real-World Data

The transformation process was tested on various datasets, with qualitative analysis conducted by comparing Worst Case and Best Case scenarios.

a) Worst Case

The visualization results using the “Ildo Sports Park 143” dataset are shown below. Figure 19 displays the IIM model applied to Frame 352, while Figure 20 shows the final transformation results with IPM and individual object separation applied to the original image, and Figure 21 shows the data in BEV format. Red-shaded areas in the BEV format indicate regions that were not observable in the original image.

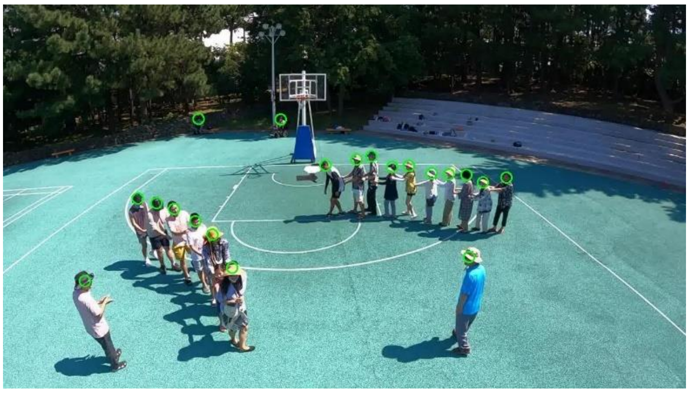


Figure 19. “Ildo Sports Park 143” Dataset, IIM Model Application

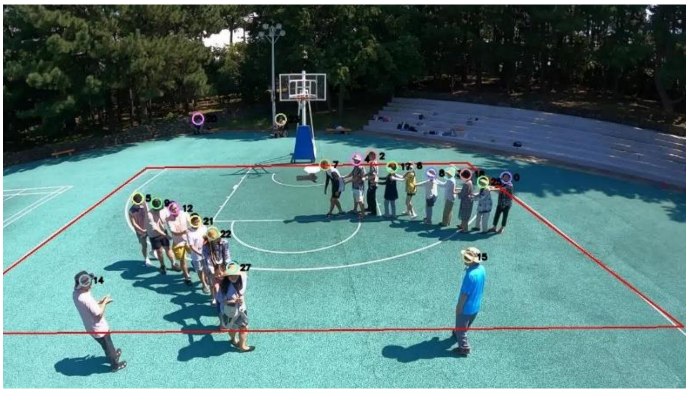


Figure 20. “Ildo Sports Park 143” Dataset, Individual Object Tracking

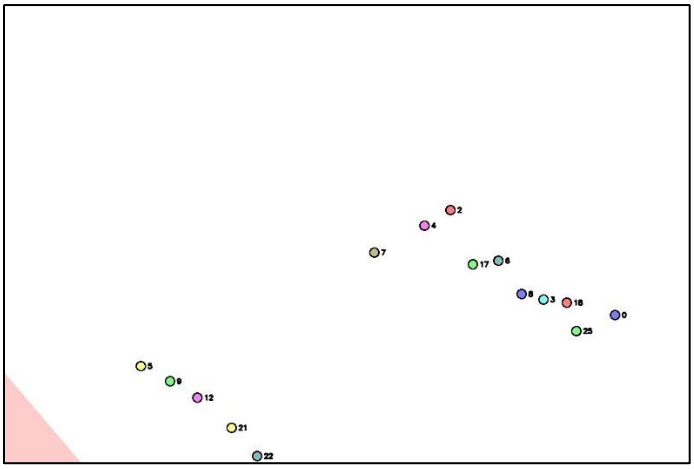


Figure 21. "Ildo Sports Park 143" Dataset, BEV Transformation Result

b) Best Case

The "Jeju Tapdong Plaza 056" dataset was used for the Best Case scenario. Figure 22 shows the IIM model applied to Frame 512, Figure 23 presents the transformed data with IPM and object separation in the original image, and Figure 24 shows the BEV transformation result.

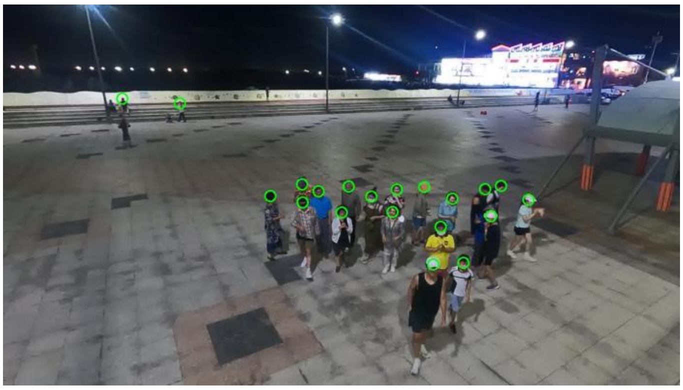


Figure 22. "Jeju Tapdong Plaza 056" Dataset, IIM Model Application

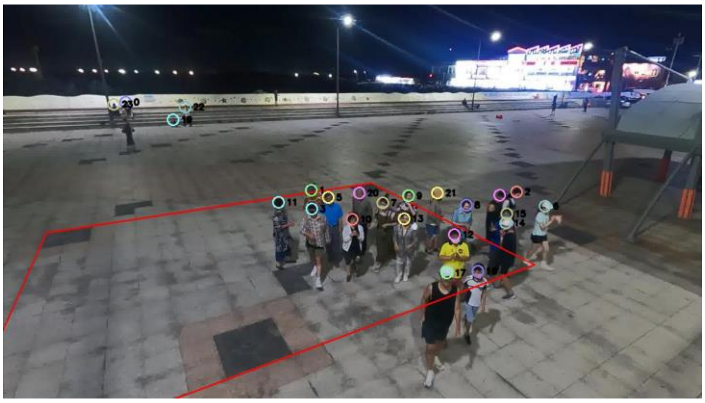


Figure 23. "Jeju Tapdong Plaza 056" Dataset, Individual Object Tracking

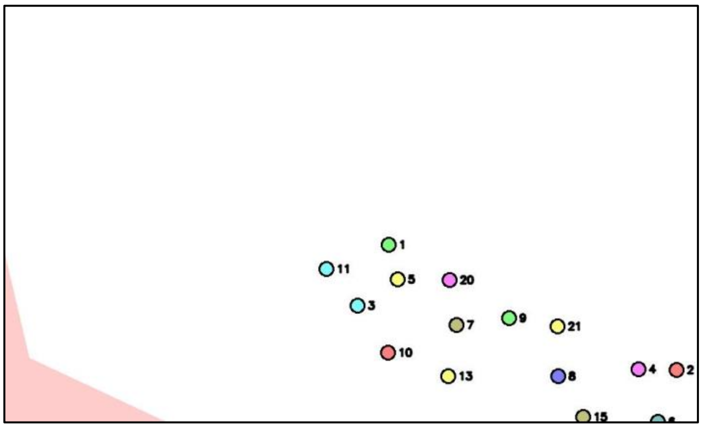


Figure 24. "Jeju Tapdong Plaza 056" Dataset, BEV Transformation Result

4.3 Application of Example Scenario: Shibuya Scramble Crossing, Tokyo

4.3.1 Transformation of Real Crowd Data

To test the framework, we applied it to video data collected from the Shibuya Scramble Crossing. The results are shown in Figures 25, 26, and 27, representing the Crowd Localization results from IIM, individual object separation, and BEV transformation, respectively.

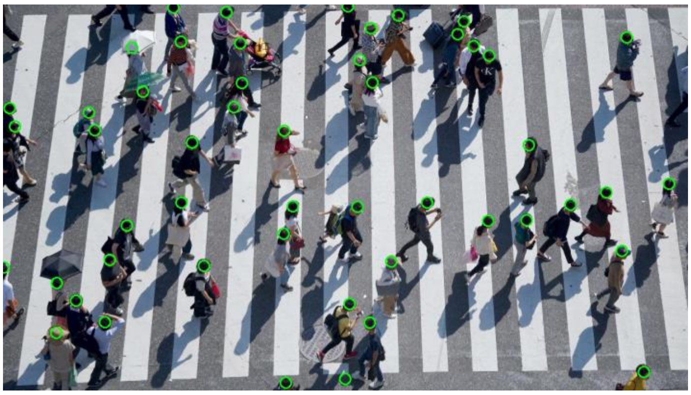


Figure 25. Example Scenario Data, IIM Model Application

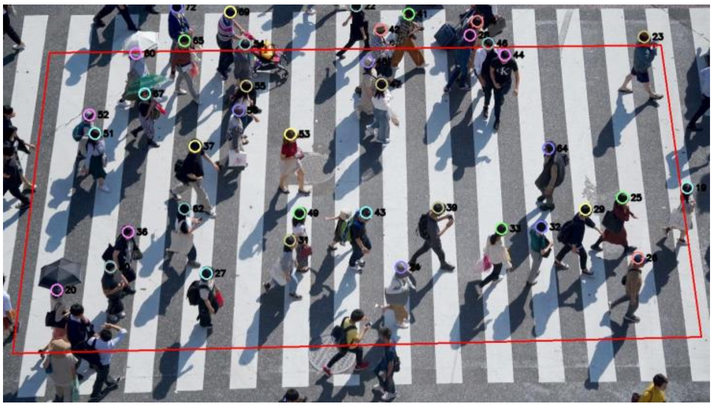


Figure 26. Example Scenario Data, Individual Object Separation

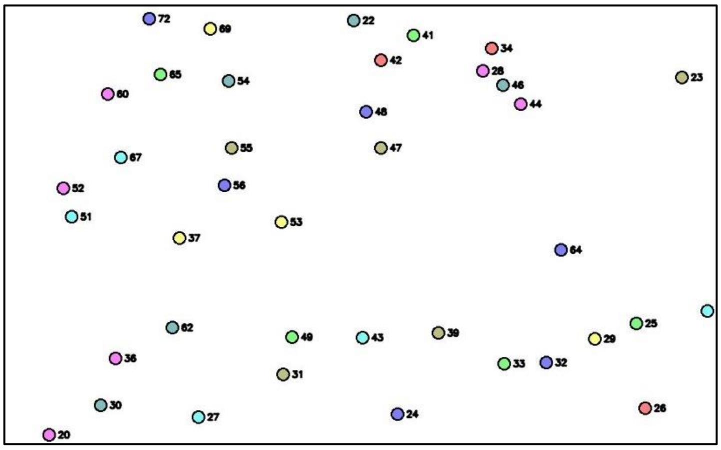


Figure 27. Example Scenario Data, BEV Transformation

4.3.2 Simulation Optimization

Using Bayesian Optimization, we optimized the model parameters to minimize the DTW distance between real crowd data and simulation data under identical initial conditions. The final DTW distance was approximately 36 cm, indicating that real and simulated crowd movements aligned within an average error of 36 cm. Table 4 lists the optimized model parameters, while Figure 28 illustrates the performance changes across various parameter pairs.

Table 4. Optimized Model Parameters

Name	Description	Value
$\boldsymbol{\alpha}$	Sharpness of Strategy Function	32520
$\boldsymbol{\beta}$	Strategy Function Proportionality Constant	1
$\boldsymbol{\chi}$	Neighbor Detection Threshold	4.10
$\boldsymbol{f_m}$	Proportionality Constant for Psychological Interaction	985
$\boldsymbol{f_p}$	Proportionality Constant for Physical Interaction	124130
$\boldsymbol{k}$	Elasticity Coefficient	0.7

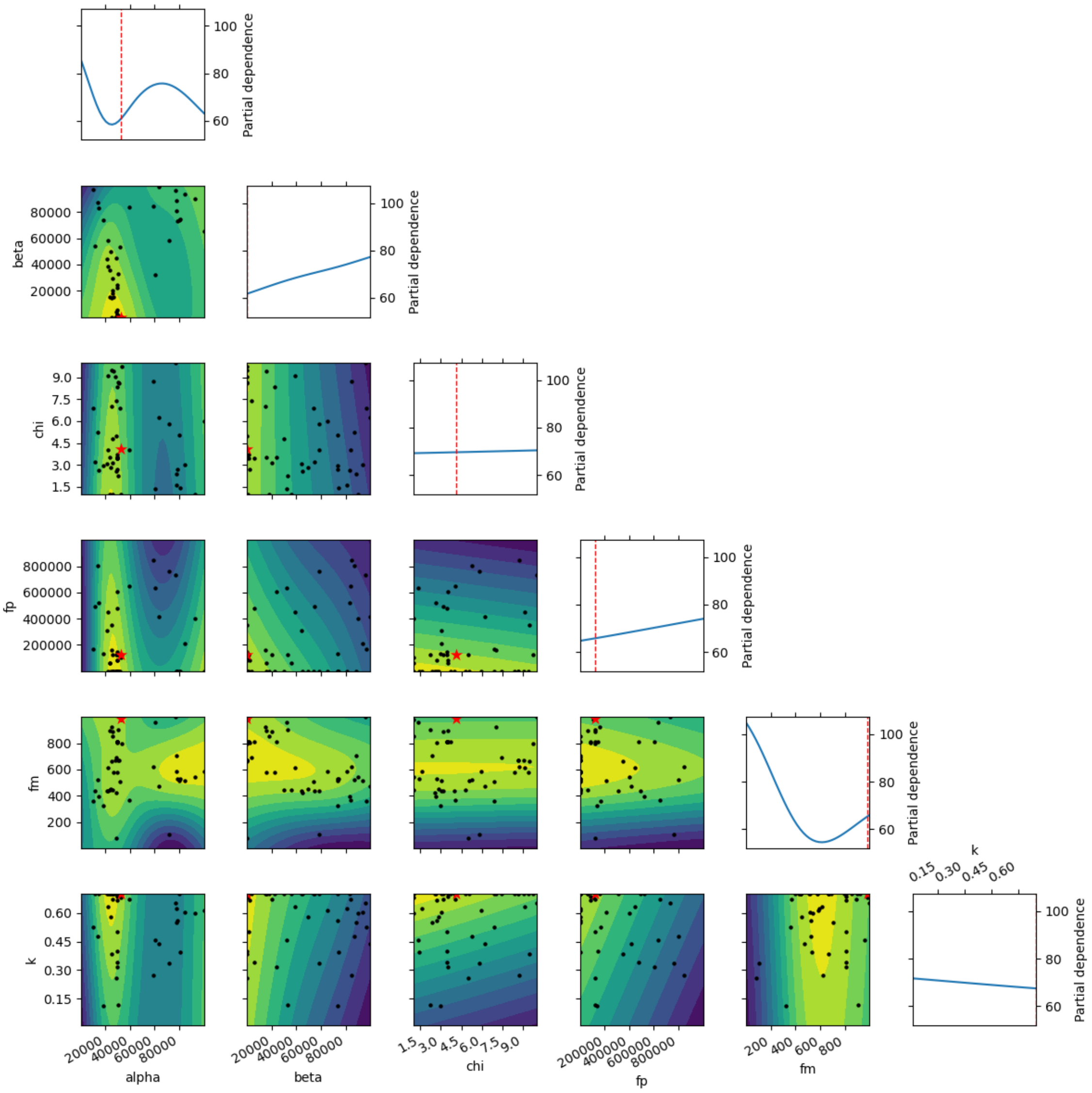


Figure 28. Performance Map for Arbitrary Parameter Pairs

5. Discussion and Conclusion

The primary objective of this study was to implement a new CFD simulation framework that more accurately replicates and reflects real crowd movement dynamics. This was achieved by incorporating three main stages in the framework: (1) development of a strategy-based CFD simulation, (2) transformation of real-world crowd data, and (3) optimization of strategy parameters through data comparison.

The results demonstrate that the strategy-based simulation effectively captured crowd dynamics in diverse scenarios, both qualitatively and quantitatively. In all four test scenarios, the model realistically represented crowd movements and allowed for flexible adjustments in crowd size and obstacle placement. However, as prior studies have noted, achieving intuitively human-like behavior was not the main focus of this study, highlighting the need for further optimization to increase the model's realism.

The real-world data transformation process proved successful across most test cases. However, certain limitations emerged, such as the fixed height assumption of 1.7 meters, which introduced significant errors in BEV data in scenarios involving diverse age groups, genders, or complex terrain. Additionally, in highly dense crowds where individuals partially obscure each other, Crowd Localization sometimes failed to detect all agents. These limitations, while not severely impacting most cases, could be addressed in future research.

The framework's application to the Shibuya Scramble Crossing scenario confirmed its effectiveness in simulating complex crowd flow patterns. Using DTW and Bayesian Optimization, the simulation model accurately depicted crowd dynamics in challenging environments and achieved high similarity with real-world data. This validated the new framework's potential as a valuable tool for advancing the field of crowd flow dynamics.

In conclusion, the CFD simulation framework proposed in this study was shown to be effective in addressing crowd flow issues. Although a simple strategy was applied in this study's example scenario, the framework allows for flexible adaptation of more complex strategies. Expanding on this approach could offer practical applications in fields such as smart city development, traffic management, and safety planning, significantly improving the accuracy and adaptability of crowd flow analysis. This framework could become an essential tool for preventing and managing real-world crowd congestion incidents, like the Itaewon disaster.

6. Recommendations

This study presented an effective approach to analyzing and solving crowd flow issues through a new strategy-based CFD model. However, several limitations were identified, and recommendations are made for future research to address these.

1. **Enhanced Strategy Optimization Using AI**
   This study optimized a few parameters within manually set strategies. To achieve complete freedom in strategy design, it is recommended to implement a high-flexibility AI model, as seen in research utilizing Deep Reinforcement Learning (Deep RL). By using DTW distance as an error function for training, a more refined predictive capability could be achieved, allowing each agent to make dynamic decisions. While computational limitations prevented this study from applying such methods, future research with greater resources could overcome this constraint.

2. **Improvement of Real-World Data Transformation Using Multi-Camera Integration**
   The fixed-height assumption and issues of overlap between agents introduced transformation errors, largely due to the limitations of single-camera data collection. Integrating multi-angle data from multiple cameras could provide more accurate reconstructions of crowd movements. This would enhance the ability to analyze individual crowd behavior and simulate complex scenarios more precisely.

3. **Consideration of Diverse Human Shapes and Individual Differences**
   Future models should incorporate variations in individual body types and behaviors to improve predictions of crowd dynamics. This would allow for a deeper understanding of the complexities arising from individual movement differences and diverse human shapes, providing a more accurate representation of crowd behavior.

This study demonstrates the potential for enhanced accuracy in crowd flow analysis using a strategy-based CFD model. Future research should focus on overcoming these limitations and developing an even more refined model, which could set a new standard in crowd flow analysis with practical applications across multiple fields.

7. Acknowledgements

Thank you for the guidance of Jo Heon-guk from Dankook University and the coaching of Kibo Mun from Seoul Science High School in the development of this research paper.

8. References

1. S. Nam. Crowd-flow simulation using the potential pathline method. *WIT Transactions on Engineering Sciences.* **125**, 27–35 (2019).

2. S. Nam. Computational analysis of bidirectional crowd flow in railway stations. *Journal of the Korean Society of Mechanical Engineers.* **44**, 475–482 (2020).
3. P. A. Langston, R. Masling, B. N. Asmar. Crowd dynamics discrete element multi-circle model. *Safety Science.* **44**, 395–417 (2006).
4. M. L. Xu, H. Jiang, X. G. Jin, Z. Deng. Crowd simulation and its applications: Recent advances. *Journal of Computer Science and Technology* .**29**, 799–811 (2014).
5. I. Sagredo-Olivenza, M. Cárdenas-Bonett, J. J. Gómez-Sanz, J. Pavón. Using graphs of queues and genetic algorithms to fast approximate crowd simulations. *MDPI Proceedings.* **2**, 1216 (2018).
6. J. Lee, J. Won, J. Lee. Crowd simulation by deep reinforcement learning. *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 1–7 (2018).
7. K. Son, J. Park. Introduction and application areas of discrete element method. *Journal of Mechanical Engineering.* **60**, 31–36 (2020).
8. C. O'Sullivan. Particle-based discrete element modeling: Geomechanics perspective. *International Journal of Geomechanics.* **11**, 449–464 (2011).
9. W. Mattson, B. M. Rice. Near-neighbor calculations using a modified cell-linked list method. *Computer Physics Communications.* **119**, 135–148 (1999).
10. L. Verlet. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review.* **159**, 98 (1967).
11. J. Dummer. A simple time-corrected verlet integration method. *Game Developer.* (2004).
12. W. C. Swope, H. C. Andersen, P. H. Berens, K. R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *Journal of Chemical Physics.* **76**, 637−649 (1982).
13. M. A. Khan, H. Menouar, R. Hamila. Revisiting crowd counting: State-of-the-art, trends, and future perspectives. *Image and Vision Computing.* **129**, 104597 (2023).
14. M. A. Ahmad, A. M. Muad. Calibration of inverse perspective mapping from different road surface images. *2021 IEEE 11th International Conference on System Engineering and Technology (ICSET)*, 139–142 (2021).
15. J. Gao, T. Han, Q. Wang, Y. Yuan, X. Li. Learning independent instance maps for crowd localization. *arXiv:2012.04164* (2020).
16. R. Shapiro. Direct linear transformation method for three-dimensional cinematography. Research Quarterly. *American Alliance for Health, Physical Education and Recreation.* **49**, 197–205 (1978).
17. E. Keogh, C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems.* **7**, 358–386 (2005).
18. M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, E. Keogh. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery.* **31**, 1–31 (2017).
19. X. Wang, Y. Jin, S. Schmitt, M. Olhofer. Recent advances in Bayesian optimization. *ACM Computing Surveys.* **55**, 1–36 (2023).