

Realistic Interactive Pedestrian Simulation and Visualization for Virtual 3D Environments

N. Brändle, T. Matyus

arsenal research

Vienna, Austria

Email: norbert.braendle@arsenal.ac.at

M. Brunnhuber, G. Hesina

VRVis Research Center

Vienna, Austria

Email: brunn@vrvis.at

H. Neuschmied, M. Rosner

Joanneum Research

Graz, Austria

Email: helmut.neuschmied@joanneum.at

Abstract—We present a visual computing effort to realistically and interactively simulate and visualize aspects of human motion behavior in virtual 3D environments. It allows virtually changing the infrastructure of a layout and assessing the consequences in terms of motion paths and visibility (where will people look at?). We first create a virtual 3D model of an infrastructure with photogrammetric reconstruction of images and obtain high-resolution video footage of real-world scenarios of the infrastructure. We calibrate video information with the 3D model in order to optimize an automatic human detection and tracking algorithm and to obtain real-world trajectories of people in world coordinates of major ground plane. These real world trajectories provide input to strategic and tactical movement aspects of pedestrian simulations and help validating the human motion simulations. The simulated pedestrian trajectories are the base for path visualizations and visibility analyses in the virtual environment. Our case study comprises an entry hall of a train station, but the approach is applicable to other environment such as museums.

I. INTRODUCTION

Virtual 3D Environments have many important applications in different fields such as computer game simulation, history and arts, archaeology and transportation. For example, Fig. 1a shows an image of the entrance hall of a main train station. Operators of such an infrastructure want to assess 1) what impact adaptations in infrastructure layouts would have on human motion behavior (e.g. placing additional ticket machines), and 2) which parts of the infrastructure will be easily perceived by pedestrians and which areas will be ignored, thus supporting analyzing guidance systems or placing billboards.

Such questions can be answered with the help of a 3D model, a realistic pedestrian simulation, visibility analysis based on simulated human paths, and an interactive application allowing to navigate through the 3D world and placing additional 3D objects. Figure 1b shows a snapshot of our realistic 3D model, including simulated human paths. This paper describes the major steps of our approach, which is applicable to any infrastructure.

One contribution concerns realistic pedestrian simulation incorporating clues of real world trajectory data obtained with an automatic state-of-the-art automatic pedestrian detection and tracking algorithm. Pedestrian simulations in gaming industry have a clear focus on real-time performance and nice-looking crowd visualizations, e.g. [1], and are usually not calibrated against real motion data. Our focus is not on



Fig. 1. Main station entrance hall in Graz, Austria, with interior wall designed by Peter Kogler; (a) Photograph of infrastructure (b) Our 3D model with visualized pedestrian paths.

real-time performance, but on valid simulations. A completely data driven simulation approach is presented in [2]. Their simulation, however uses a top view camera (with very limited coverage) and experimental setup with volunteers acting as ‘normal’ as possible. The work in [3] includes real scenarios from a top view camera and a simple head-based tracker looking for dark areas. Our rationale is to capture motion behavior of people in real-world scenarios with high-resolution images of an oblique camera view, such that extended trajectories can be obtained with an automatic people detection and tracking algorithm to gain insights into origin-destination flows. While automatic pedestrian detection and tracking with computer vision has matured considerably over the past 10 years, there still remain many challenges, especially for complex scenarios with many people [4]. We employ the popular Histogram-of-Gradient (HOG) based detection method [5] and improve it in terms of speed and classification performance.

Visual surveillance can provide information about the real motion behavior, so the question arises why pedestrian simulation is necessary at all. There are two major reasons to simulate motion behavior: Firstly, many built environments have an architecture which does not necessarily allow full video coverage. Too many cameras might be required to cover an entire building. Secondly, many experiments would be too costly or impossible to be performed in real world (e.g. placing ticket machines at different locations). Simulations are therefore well justified.

Passenger flows can be visualized by depicting individual people or by accumulating streams which represent large

numbers of people. Nakamura et al. [6] provides a combined approach, visualizing overall behavior with accumulated color coded lines, and marking individuals of interest (stationary people or collision avoidance) with dots. Katabira et al. [7] visualize people trajectories by probability density maps, which can be interpreted as crowd density. Very high densities can represent areas that are prone to congestion. However, their work does not represent walking speeds of passengers – a significant parameter for our proposed application.

People trajectories allow computing the visibility of different parts of an infrastructure. Our contribution adapts the concept of shadow maps introduced in [8], where shadow colors are set according to illumination. Instead of shadow maps we create *visibility maps*, in which colors indicate which areas are ‘seen’ by many pedestrians (the definition of the word differs from the one used by Goradia et al. [9]). A similar concept has been presented in [10] for motion analysis without resorting to individual people tracking.

Our overall contribution is a combined visual computing approach with realistic pedestrian simulations derived from an oblique camera angle covering an extended region, path and visibility map visualization, and real-time rendering and interaction with objects. The paper is organized as follows: Section II describes aspects of data acquisition, including 3D model creation, video recording and calibration and human motion tracking. Section III introduces the concept of pedestrian simulation, incorporating real world data. Section IV describes visualization aspects, including visibility map creation and path visualization. Section V describes the real-time interaction in 3D.

II. DATA ACQUISITION

This section deals with aspects of acquiring data which are necessary for both the simulation and the visualization.

A. 3D Model

The 3D models are built with photogrammetric reconstruction from images created with an off-the-shelf SLR camera. The reconstruction method – city dense modeling – is described in [11]. To improve the details of the model, the created point clouds are combined with CAD plans of the infrastructure. The resulting 3D data has been manually completed with the above images to create textures and further details of the indoor environments.

B. Video Recording and Calibration

We have mounted an IP surveillance camera (Axis 223M) with a resolution of 1600×1200 pixels capturing images at a rate of approximately 10 frames per second (IP cameras do not provide fixed frame rates). The high image resolution allows capturing the most interesting parts of the entrance hall and automatic people tracking with a single video, thus avoiding tracking across multiple cameras. Figure 2a shows the field of view of the camera. The wide viewing angle leads to severe lens distortions – straight lines of the build environment should lead to straight lines in the image, which

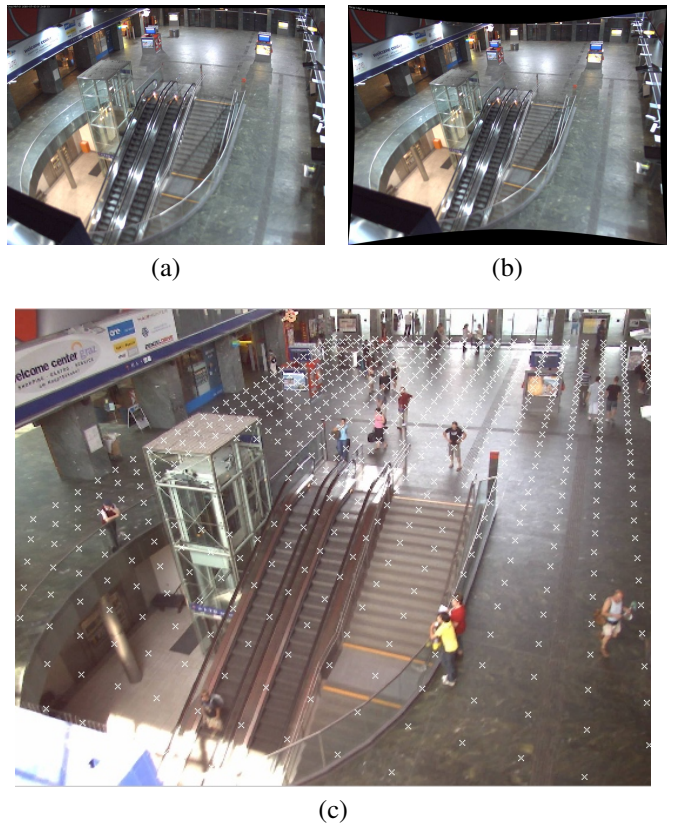


Fig. 2. Video recording and calibration. (a) camera field of view (b) undistorted camera view assuming radial lens distortion (c) virtual calibration grid at one meter above the ground floor, obtained with the help of 3D model.

is clearly violated in Fig. 2a. Assuming radial lens distortion and the projection center coinciding with the image center, we calibrate the parameter of the first order model with the help of the optimization algorithm as described in [12]. The thus corrected image can be seen in Figure 2b.

After lens correction we identify a linear transformation between pixel coordinates of the (corrected) image plane and the coordinates in the three-dimensional infrastructure. We estimate a planar homography from a set of N image points \mathbf{x}_i and a corresponding set of 2D planar world points \mathbf{X}_i , $1 \leq i \leq N$ with the help of a normalized Direct Linear Transform [13]. The set of world points \mathbf{X}_i is chosen to lie on a planar grid one meter above the ground floor of the station hall, corresponding to the height of people’s waist: Automatic people detectors are based on classifying rectangular pixel sets. Associating the centroid of rectangles classified as human with the people’s waist turned out to be more stable than associating the lower center with people’s feet (see also Fig. 3). Our approach therefore places into the 3D model a virtual planar point grid at the average waist height of one meter, and renders a 1600×1200 image of the 3D model with approximately the same camera field of view as the real undistorted camera view. This synthetic image has designated pixel values at the grid locations. This approach avoids identifying and/or using markers one meter above the

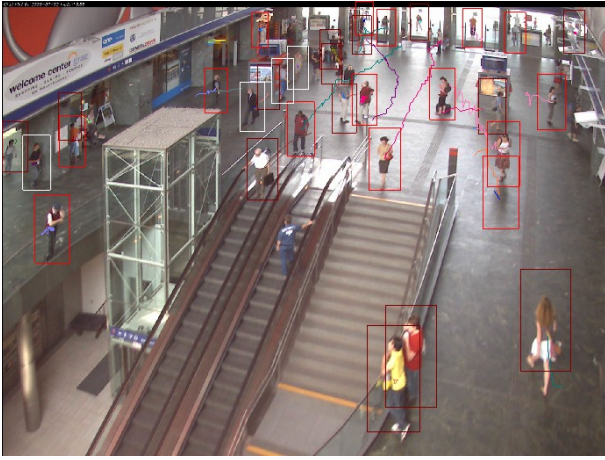


Fig. 3. Automatic people detection and tracking.



Fig. 4. All smoothed trajectories with more than 30 trajectory states, computed from a 15 minutes video.

ground plane in the real infrastructure. Figure 2c illustrates the identified point set superimposed as white crosses on a lens-corrected image. While the single planar assumption is clearly violated for the escalator and the stair region, the floor of the hall covers all interesting entry and exit regions with respect to origin-destination flows.

C. Human Motion Tracking

Automatic people detection for the recorded video footage is based on an improved Histogram of Oriented Gradients (HOG) Cascade [5]. The static camera view allows fixing the possible scales (i.e. pedestrian sizes) with respect to the ground plane (cf Fig. 2c), thus reducing the number of necessary classification tasks and false detections in wrong scales. A second improvement concerns replacing the AdaBoost classifier by a Support Vector Machine (SVM) classifier. All features in a single cascade step are handled as one input vector instead of several smaller ones as in [5].

Tracking of detected people is based on a combination of the Kalman filter and a simple first order motion model. The Kalman filter prediction works well if a person cannot be detected for a while (e.g. due to an occlusion), and the first order motion model reacts very fast to abrupt motion changes. When the HOG people detector indicates multiple detections for one person, the tracking process selects the best one depending on position prediction, detection confidence, and the visual appearance matching. Data association is based on a Global Nearest Neighbor matching implemented by the auction algorithm [14].

Figure 3 shows a snapshot of people detection (rectangles) and tracking (colored lines). The obtained trajectories are smoothed in a postprocessing step by cubic spline approximation in order to get rid of the ragged tracks Figure 4 shows all smoothed trajectories with a minimum length of 30 states, computed from 15 minutes video. These data provide valuable input for the pedestrian simulation.

III. PEDESTRIAN SIMULATION

Pedestrian simulation can provide motion behavior for regions not covered by cameras and allow virtual experiments. Pedestrian motion behavior is often described in three different levels [15]. The *strategic level* determines the arrival time of the pedestrian at the infrastructure, its entry position and the pedestrian's goal (e.g. going to the train). The *tactical level* describes the route a pedestrian will choose to move from the current position to the target. The *operational level* calculates the actual movements, including collision avoidance.

In order to obtain more realistic scenarios, different pedestrian types can be introduced with different operational and tactical parameter sets. Our simulation approach obtains clues from real people trajectories on a tactical and strategic level. It also uses people trajectory data to validate the realism of the simulation.

A. Operational Level

We model human motion on an operational level based on an adapted social force model [16]. Force models provide explicit equations for the movement of pedestrians. They are defined in continuous time and space by providing differential equations for the acceleration of a pedestrian. Acceleration is then integrated once in order to obtain velocity, and integrated twice in order to obtain the positions of the pedestrian. Using an analogy from physics, acceleration is identified with force. Different forces act on each pedestrian i as follows:

$$F_i = F_i^a + F_i^p + F_i^w, \quad (1)$$

where F_i^a denotes the attractive force directed towards the pedestrian's goal, F_i^p denotes the repulsive forces directed away from other pedestrians and F_i^w prevents the pedestrian from colliding with walls or other obstacles. The attractive force depends on the desired speed of the pedestrian. This parameter can be varied in order to define different pedestrian types.



Fig. 5. Simulated trajectories, with original destination flow obtained from data visualized in Fig. 4.

B. Special Areas

The mere goal directed operational behavior will not lead to realistic motion in special areas like escalators and in front of ticket machines. It is therefore necessary to model escalators and ticket machines separately. Every *ticket machine* is modeled as an attraction point with a given service time, and a queuing line in front of it. A pedestrian intending to take an *escalator* first aims at a point in an area in front of the escalator. After having reached the beginning, the pedestrian chooses the right or left lane, depending on the size of the crowd in front of the right lane. On the right lane, pedestrians are supposed to stand still on the escalator. The velocity on this lane is therefore lower than on the left lane.

C. Tactical Level

The routing algorithm is based on two assumptions: First, all pedestrians choose the shortest path to reach their goal and second, each pedestrian knows the infrastructure completely, i.e., at any point in the building the pedestrian knows the shortest path to the target. In order to navigate through the infrastructure, the next visible corner lying on the shortest path to the target is set as an intermediate goal. Having reached this point, the procedure is repeated until the pedestrian reaches its final goal.

If a pedestrian wants to buy a ticket he chooses the nearest ticket machine independently of the length of the queue in front of it. The need to buy a ticket is one of the tactical parameters determined by the pedestrian type.

D. Incorporating Real World Data

Figure 5 shows the simulated trajectories, where the strategic and tactical input (people counts, entry times, origin-destination flows) has been determined by the postprocessed real trajectory data (Fig. 4). Note that simulated 3D world coordinates have been projected back into the image by the planar homography, leading to distorted results for the



Fig. 6. Simulation validation. In a set of real trajectories, one real trajectory (in red) is replaced by a simulated trajectory (light blue and dark blue, see text).

escalator and stair area. The full 3D visualization of people paths is described in Sect. IV.

The simulated trajectories of Fig. 5 include more straight lines than the real trajectories, and the question arises where the simulation is valid. One interesting validation approach is to use all real trajectories of a time interval, leave out one trajectory and simulate the omitted trajectory with the remaining trajectory configuration. Figure 6 shows an example of a person path entering the main hall and heading towards the escalator (red line). The light blue line indicates the simulated trajectory, which severely deviates from the real trajectory. The reason is that in the imaged scenario there are some stationary people (having a chat) which are not reliably tracked by automatic people detection and tracking, such that these people suddenly disappear and reappear. After the straight part of the light blue trajectory, the stationary person (wearing grey clothes) is suddenly detected and causes a sharp bend due to collision avoidance. Manually replacing the unreliable partial trajectories with a consistent stationary trajectory leads to a more plausible simulation, indicated as the dark blue line in Fig. 6.

IV. VISUALIZATION

While visualizing simulated trajectories in the original 2D camera view provides insights into the motion behavior, integrating human trajectories into the 3D models enables to compute visibility maps and more intuitive visualizations of direction and speed.

A. Creating Visibility Maps

Visibility maps are first initialized by placing white base textures on all surfaces of the 3D model. In general, it will be necessary to identify planar parts of the model and suitable texture coordinates due to the possibly varying quality of 3D models. Our approach groups the triangles of the 3D model into classes, based on position and normal vector information. (with a tolerance parameter). The triangles belonging to same class, i.e. to the same plane, are then assigned to texture coordinates based on their position within a 2D bounding box that encompasses all of them.

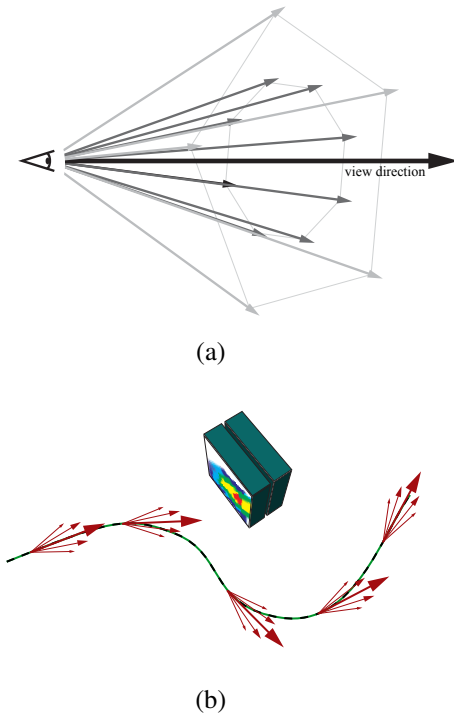


Fig. 7. (a) Ray bundles as circles around the view direction; (b) Face colored by ray bundle.

Every triangle class is associated with a single empty visibility map, which is updated based on human view simulation. Human vision is usually not punctual, but rather encompasses a certain field of view. We model such as field of view as bundles of N rays shot from the position of the current person path trajectory point, implicitly defining the eye level. The first ray of the bundle is shot exactly in the direction of the view vector calculated for the pedestrian path. The other $N - 1$ rays are shot in circles around the center view vector. The first circle uses eight rays and every further circle doubles the number of rays (i.e. 8, 16, 32 rays).

Figure 7a shows an example with 14 rays shot from one point. The first eight rays are shot in the middle of the set angle and the rest of the rays is shot regularly on the set angle away from the view direction.

The ray shooting process is optimized by computing a K-d tree for the 3D model as described in [17]. Every ray on a surface hit is reflected in the corresponding visibility map. The value at this point is increased depending on the priority of the ray. Rays closer to the exact view direction are assigned a higher weight than peripheral ones. The values are normalized to the total number of person path points and rays per point. This approach simulates the fact that peripheral vision receives less attention than a person's central field of view by using the different priority levels.

Figure 7b illustrates the coloring of a plane with the ray shooting process. A ray bundle is shot from any defined position on the person path and the points hit on the base textures are colored. The face shown in Fig. 7b is marked

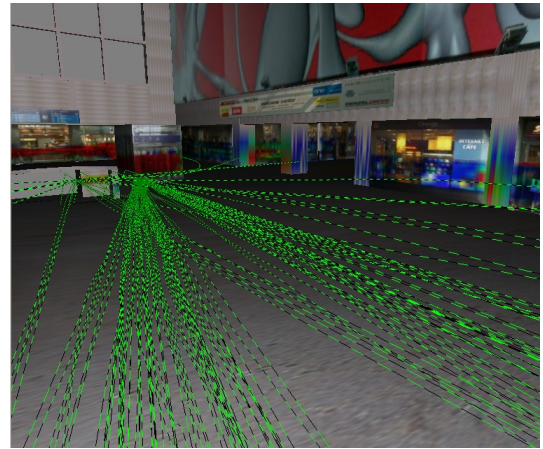


Fig. 8. Application screenshot with trajectories and visibility maps.

red on the position which is hit by more than one ray. The outer sections of the hits are colored in yellow green to blue depending on the priority of the rays as mentioned.

B. Rendering the Model with Visibility Maps

The 3D model is rendered with a simple headlight. The complexity of the railway station leads to a large number of textures (60). We therefore use texture atlases and compress them in order to ensure real-time rendering. Since the numerous individual visibility maps are relatively small, they can also be efficiently combined into larger maps, i.e. texture atlases, which greatly improves rendering performance by avoiding a large number of context switches.

The visibility maps are used like light maps. The base texture and visibility map of a surface are combined in a pixel shader. This approach allows changing the visibility map at runtime without affecting the basic textures. Furthermore, base textures can be reused throughout the model without consuming large amounts of texture memory.

Fig. 8 shows a part of the model of the Railway station in Graz in Austria. The visibility maps are easy to identify on the walls and the several shops in the railway station. Especially the elevator lies in the focus of the simulated persons. For this example 9 rays have been shot for every position on the trajectory in an angle of 5 degrees.

C. Visualizing Person Paths

The directions of moving persons cannot be discerned with simple solid lines. Hence the individual vertices are assigned alternating one-dimensional texture coordinates of zero and one. In the vertex shader, these texture coordinates are increased by a small value for every frame. A small two-color texture creates moving stripes along the pathways. Additionally, since vertices are located at regular time intervals, the length of individual line sections varies with the simulated person's walking speed, which can thus be directly discerned from the animation. Fig. 9 shows a part of the trajectory on an escalator. The longer subparts of the right trajectory show persons moving faster down than the ones on the left side.



Fig. 9. Different lengths of trajectory parts on the escalator.

This because people usually stand still on the right hand side of an escalator and walk down on the left hand side. Different trajectories on Fig. 8 can also be interpreted this way.

V. INTERACTION IN 3D

An important aspect is the interaction of the user with the virtual 3D environment. Our application supports interactive placement and removal of additional 3D models. The base textures of objects also can be changed at runtime, for example to explore the placement of signposts and advertisements. The newly imported objects are also integrated into the simulation of person paths and visibility map creation via an XML-interface.

The newly created paths are directly integrated in the application and replace the old person path trajectory. Subsequently, visibility maps are recalculated in a newly generated thread to keep the standard application running. Figure 10 illustrates person paths before and after integrating an obstacle object and a recalculating motions. The calculation of the updated visibility map was with only one ray per person path point, while the calculation before used ten rays per point. Animated examples are provided on <http://old.vrvis.at/rendering/downloads/OEBB/Graz/>.

VI. DISCUSSION AND FUTURE WORK

Incorporating real motion data from a general oblique camera view and automatic people tracking into a pedestrian simulation is feasible and can provide validation data. Not surprisingly, the real scenario in the entrance hall of the infrastructure with its incoherent motion could not be completely reliably analyzed by the computer vision components, leading

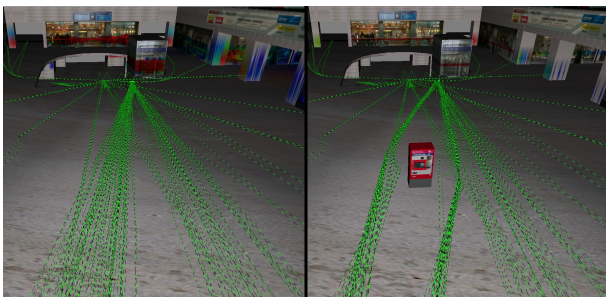


Fig. 10. Person paths before and after placing a 3d object (ticket machine) and recalculating trajectories

to false detections and broken trajectories. Future work will include clustering of both the real and simulated trajectory data into motion patterns.

The resolution of visibility maps is currently fixed. A possible improvement would be to adapt the resolution with respect to the size of the model's subpart. It is necessary to find a good trade-off between accuracy and performance to create every visibility map face. Sometimes the visibility maps are not easily identifiable on the textures. An improvement would be to create some contrast between the visibility maps and the textures.

ACKNOWLEDGMENTS

This work has been funded by the Austrian FIT-IT Visual Computing Program und Grant No. 813406/1245. The authors gratefully acknowledge the support of the companies of the Austrian Railways ÖBB.

REFERENCES

- [1] A. Treuille, S. Cooper, and Z. Popovic, "Continuum Crowds," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1160–1168, 2006.
- [2] K. Lee, M. Choi, Q. Hong, and J. Lee, "Group Behavior from Video: A Data-Driven Approach to Crowd Simulation," in *Proceedings ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2007.
- [3] A. Johansson, D. Helbing, and P. Shukla, "Specification of the Social Force Pedestrian Model by Evolutionary Adjustment to Video Tracking Data," *Advances in Complex Systems*, vol. 10, pp. 271–288, 2007.
- [4] N. Haering, P. Venetianer, and A. Lipton, "The evolution of video surveillance: an overview," *Machine Vision and Applications*, vol. 19, pp. 279–290, 2008.
- [5] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast Human Detection using a Cascade of Histogram of Oriented Gradients," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR2006)*, 2006.
- [6] K. Nakamura, H. Zhao, R. Shibasaki, K. Sakamoto, and N. Suzukawa, "Visualizing passenger flow in railway station using laser scanners," in *9th Computers in Urban Planning and Urban Management Conference (CUPUM)*, 2005.
- [7] K. Katabira, T. Suzuki, H. Zhao, and Y. Nakagawa, "An Analysis of Crowds Flow Characteristics by using Laser Range Scanners," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 37, Part B5*, 2008, pp. 953–956.
- [8] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering 3rd Edition*. Natick, MA, USA: A. K. Peters, Ltd., 2008.
- [9] R. Goradia, A. Kanakanti, S. Chandran, and A. Datta, "Visibility map for global illumination in point clouds," in *Proceedings 5th Intl. Conf. on Computer Graphics and Interactive techniques in Australia and Southeast Asia (GRAPHITE'07)*, New York, NY, USA, 2007, pp. 39–46.
- [10] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian Detection of Independent Motion in Crowds," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR2006)*, 2006.
- [11] A. Irschara, C. Zach, and H. Bischof, "Towards wiki-based dense city modeling," *Published at the Workshop on Virtual Representations and Modeling of Large-scale environments (VRML 2007)*, 2007.
- [12] F. Devernay and O. D. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, 2001.
- [13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [14] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [15] S. P. Hoogendoorn and P. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research, Part B: Methodological*, vol. 38, pp. 169–190, 2004.
- [16] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, pp. 4282–4286, 1995.
- [17] V. Havran, "On improving kd-trees for ray shooting," in *In Proc. of WSCG 2002 Conference*, 2002, pp. 209–217.