Techniques and Applications of Urban Data Analysis

Dissertation by

Sawsan Nabeel AlHalawani

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

King Abdullah University of Science and Technology, Thuwal,

Kingdom of Saudi Arabia

The dissertation of Sawsan Nabeel AlHalawani is approved by the examination committee.

Committee Chairperson: Niloy J. Mitra

Committee Member: Bedrich Benes

Committee Member: Bernard Ghanem

Committee Member: Peter Wonka

ABSTRACT

Techniques and Applications of Urban Data Analysis

Sawsan Nabeel AlHalawani

Digitization and characterization of urban spaces are essential components as we move to an ever-growing 'always connected' world. Accurate analysis of such digital urban spaces has become more important as we continue to get spatial and social context-aware feedback and recommendations in our daily activities. Modeling and reconstruction of urban environments have thus gained unprecedented importance in the last few years. Such analysis typically spans multiple disciplines, such as computer graphics, and computer vision as well as architecture, geoscience, and remote sensing.

Reconstructing an urban environment usually requires an entire pipeline consisting of different tasks. In such a pipeline, data analysis plays a strong role in acquiring meaningful insights from the raw data. This dissertation primarily focuses on the analysis of various forms of urban data and proposes a set of techniques to extract useful information, which is then used for different applications.

The first part of this dissertation presents a semi-automatic framework to analyze facade images to recover individual windows along with their functional configurations such as open or (partially) closed states. The main advantage of recovering both the repetition patterns of windows and their individual deformation parameters is to produce a factored facade representation. Such a factored representation enables a range of applications including interactive facade images, improved multi-view stereo

reconstruction, facade-level change detection, and novel image editing possibilities.

The second part of this dissertation demonstrates the importance of a layout configuration on its performance. As a specific application scenario, I investigate the interior layout of warehouses wherein the goal is to assign items to their storage locations while reducing flow congestion and enhancing the speed of order picking processes.

The third part of the dissertation proposes a method to classify cities based on their functional behavior. Commonly used computational approaches concentrate on geometric descriptors, for both images and laser scans. Instead, I analyze street networks, both their topology (i.e., connectivity) and geometry (i.e., layout), in an attempt to understand the factors that play dominant roles in determining the characteristic of cities. A set of street network descriptors is proposed to capture the essence of city layouts and used, in a supervised setting, to classify and categorize various cities across the world.

Each part of the dissertation shows the utility of the proposed methods through describing a variety of applications on different examples.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere gratitude to all the people who were beside me throughout the coursework of this dissertation.

First and foremost, my utmost gratitude to my supervisor, Niloy Mitra, for all the support and opportunities he has provided. He has always been willing to help with his invaluable guidance, feedback, advice, and encouragement. His passion and enthusiasm have inspired me to enjoy researching in this field.

My special thanks extend to Yongliang Yang who has never hesitated to answer my queries or discuss new ideas. My appreciation also goes to Dong-Ming Yan, Han Liu, Youyi Zheng, Mohamed Ibrahim and Duygu Ceylan for their support, helpful discussions and excellent suggestions.

I am thankful for having the privilege to learn from a professional group of faculty members, researchers, and students at the Visual Computing Center (VCC) at KAUST. Many thanks go to Liliana Rivera and Tina Smith for their administrative help and cheerful smiles. Special thanks to my dearest friends: Sara Shaheen, Amal Abolhalasn, Imene boudellioua, Doha Hamza, Huda Ibied, among many others, for being there at the easy and hard times and making life at KAUST an enjoyable experience.

My deepest gratitude goes to my parents, who were beside me all the times. I thank them for their prayers and faith, for tirelessly supporting me throughout my studies, and for encouraging me to pursue my dreams. I am also grateful to my brothers: Abdullah and Muhammad and my sisters: Abeer and Yasmeen for their unconditional love and support. My sincere thanks go to my parents-in-law for their faithful prayers and support. My gratitude also extends to many other friends and family members who have supported me even with their kind wishes.

This dissertation would have never been possible without my source of strength, my husband, Ali. Thank you for helping me keep faith in myself and motivating me to achieve my dreams. Special thanks to my son Omar who brightened my life in the middle of this journey. His smiles were the best encouragement to move forward. I thank Ali and Omar for bearing with me during my studies. Words would never be enough to express my gratitude to them.

Thank you all. Nothing would have been possible without your great help and support.

dedicated to

my parents,

Nabeel and Jalilah

my husband and son,

Ali and Omar

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The word *urban* has evolved from the Latin word *urbãnus* describing the characteristics and lifestyle of a town or a city [13]. Typically, urban areas are identified by the high density of human population in many built environments, i.e., man-made surroundings which comprise of many residential buildings, workplaces, hospitals, parks, parkings, etc. In contrast, the low population density and large amounts of undeveloped land characterize rural areas such as villages. Historically, people have moved from rural to urban areas that are created from scratch and slowly evolve to meet evolving space and resource demands. According to the 2014 Revision of the World Urbanization Prospects published by the United Nations [14], more people now live in urban areas than in rural areas. Specifically, the world's urban population increased from 30% of the world's population in 1950 to 54% in 2014, and is expected to reach 66% by 2050.

The accelerated increase in the population and economy forces urban systems to change and grow leading to further expansions. Cities and towns are regularly reconfigured to meet both national and regional goals of the society. Hence, new responsibilities arise for stakeholders and decision makers to effectively lead the process of urban development and maintain intensive policy coordination and investment choices, which in turn increases the need for valuable tools to support the decision-making process [15]. A good example of urban growth is the construction of the educational campus for King Abdullah University of Science and Technology - KAUST on the western coast of Saudi Arabia. The campus was created from scratch, and the government had determined to perform a comprehensive development of the villages near KAUST such as Thuwal [16]. Figure 1.1 gives a picture of the campus before and after construction.

Figure 1.1: An example of urban development; (left) KAUST under construction in 2008, (middle) the 3D model of KAUST, and (right) the built campus [1].

Various resources contribute largely to the shaping of urban areas. A common goal is to move these resources across different scales of an urban environment. For example, having a well-designed infrastructure plays a strong role in the efficiency of the city to support the water, electrical and telecommunication supply and services. In addition, building an efficient transport network is vital to move people across or within cities to reach their living and work places. Moreover, managing the resources inside urban spaces improves the quality of the place. For instance, building interiors should be connected to facilitate people's movement in offices or apartments which require efficient functioning of spaces. Another important aspect is designing the place to meet the personal preferences such as having well-let rooms, good ventilation, good views from windows, and close-to a preferred location. Understanding the various effects would facilitate the understanding of the whole environment and provide a better sense of continuity between the different scales.

The continuous demand for cities to grow and develop increases the need for having efficient tools to support the planning and preparing for this growth which, in turn, maximizes the benefits of urbanization. Such tools could be useful to model, analyze, understand, simulate and improve the urban designs. Using these tools would provide new perspectives in understanding the growth patterns over space and time. Hence, they could allow the creation of virtual models and provide useful applications for urban planners.

In the context of computer graphics, procedural modeling is commonly used to create realistic 3D models of urban environments which makes it inevitable for the usage in many applications. The resultant virtual environments are used for simple applications and computer games and play an ever-increasing role in people's daily lives especially with the growing interaction with machines, robots, and surroundings using visual inputs and outputs [17]. Examples of applications that benefit from urban models (see also [18]) include emergency control, security training, civil protection, driving guidance, and urban planning, to name but a few. Figure 1.2 shows some example applications.

It is now possible to have a virtual tour and visit new places using the advanced 'cyber-tourism' technologies with lower costs and less effort. Applications of cyber tourism can be achieved through various forms such as virtual reality worlds, 3D models, panorama images, GIS tools or pictures. For example, a user can have a virtual tour of the Dolmabahce Palace in Turkey, Istanbul [19] (see Figure 1.2a). Moreover, navigation applications are widely used especially with their high availability and rapid advancement. In modern cars, users can connect their smartphones to the car navigation system as shown in Figure 1.2b. Further, virtual cities and urban models have been an essential feature of recent computer games and movies (Figure 1.2c). Finally, urban modeling has its usage in archeological research. The data can be acquired directly from building remains and combined with the knowledge from a variety of sources such as maps, books, and paintings. The generated models can be of high value for education and planning purposes as shown in Figure 1.2d.

(a) a cyber tour in the Dolmabahce Palace in Istanbul [19].

(b) an application connecting a smartphone with the car navigation system [20].

(c) an entertainment example [21].

(d) a digital documentation of cultural heritage sites on the West African coast [22].

Figure 1.2: Examples of various applications for the use of urban data.

One of the most important uses of urban modeling is in the multidisciplinary process of urban planning involving economic, sociological, transportation, and environmental considerations. The adoption of virtual prototypes helps to demonstrate and evaluate various aspects of urban planning at an early stage which aids decision makers and stakeholders to exchange ideas, assess alternatives and reach consensus (refer to Figure 1.1). A successful urban plan, which is simulated and evaluated thoroughly at an early stage, would improve the urban regeneration project concerning cost, efficiency, and productivity [23]. There are many software applications which facilitate the generation of urban models, for example, CityEngine [2]. Following the procedural modeling approach, CityEngine is used to create detailed large-scale 3D models of cities and efficiently design and model virtual urban environments. It generates the models through a predefined rule set enabling the creation of complex parametric models. Figure 1.3 shows an example of a virtual model created of a city using CityEngine which exhibits the realistic entities of the environment.

Figure 1.3: CityEngine software enables the creation of virtual 3D urban models in their realistic context. Figure courtesy [2].

Having a virtual model of a city involves having a model of its transport network such as roads, streets, or railways. Besides considering the design and features of the network, it is also vital to model the behavior of its transportation and traffic flow. Recently, interesting applications have addressed the challenging problem of optimizing traffic behavior and providing traffic flow simulation. Traffic patterns involve many parameters such as road occupancy, travel time and travel distances. This provides a better understanding of the functional and behavioral style of the transportation in an urban space of the moving objects such as people, vehicles, taxis and many others. Traffic modeling has various applications such as in emergency situations to design evacuation and action plans.

Rapid advances in acquisition technologies and portable scanning setups have vastly simplified data acquisition at various scales: maps are obtained through careful measurements to capture city layouts (OpenStreetMap, Google maps); scans for neighborhoods, street-level acquisition, and facade images are possible using LiDAR scans, aerial scans, and air-drone scans; while Kinect and multi-view stereo reconstruction methods are used for obtaining floor plans and interiors of buildings. Generating highly detailed 3D models requires the coordination between the resources and utilizing the advanced algorithms. For example, FalconViz is a 3D aerial scanning and modeling system that generates high-quality 3D models. The process starts by gathering high-resolution images using a specially configured autonomous multirotor copter navigating a pre-specified flight path. Then, the images, orientations, and positioning data are processed in real time using Aerial SFM software to generate an interactive 3D model offering documentation of the scanned area [3].

Figure 1.4: FalconViz hexacopter for 3D aerial scanning and modeling. Figure courtesy [3].

Scope of this dissertation The access to vast volumes of data and the availability of advanced processing and modeling techniques provide a unique opportunity to analyze, understand, and predict behaviors in urban spaces at different scales. In this dissertation, the primary objective is to provide a computer graphics contribution to the field of urban reconstruction by proposing various techniques for the analysis of urban data. The primary focus is studying three types of data which are facade images, interior layout configurations, and street networks. The proposed methods provide analysis techniques for each type of the data with a demonstration of some possible applications.

1.1 Problem Statement

This dissertation addresses the problem of urban data analysis to facilitate its use for further applications. Urban reconstruction usually goes through various steps along a pipeline which could use different types of data. Urban spaces exist in a range of scales which could also be referred to as the size or proportion from one element to the other. Studying, analyzing and processing the components of an urban environment on a variety of scale levels convey different types of information and provide insights into a variety of urban aspects. This plays a vital role in the analysis of the whole environment and provides a better sense of continuity between the different scales. With this consideration, a set of techniques were proposed to analyze and understand urban data at three different levels. Figure 1.5 gives an overview of the different scales that are addressed in this dissertation.

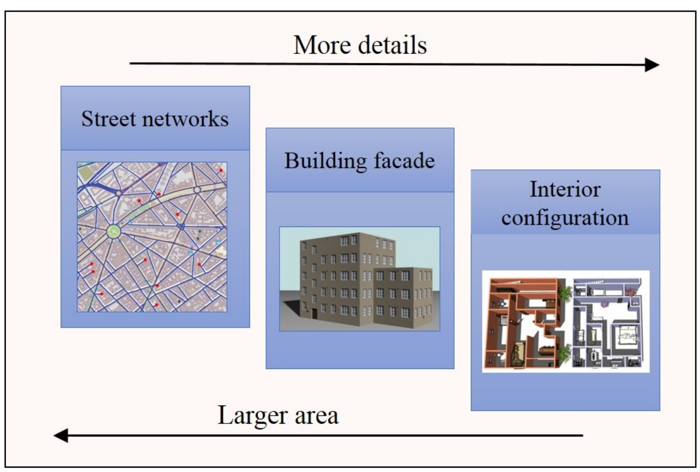


Figure 1.5: A variety of urban scales. When going from left to right, the urban area decreases but the level of detail increases. Figures courtesy of [4, 5, 6].

The first target of this dissertation is the exterior part of an urban model, commonly known as facade, which defines its nature and largely influences the public experience. Facade images are integral components of the reconstruction process as they are used to generate the external appearance of a model, either for texturing or approximating the 3D geometry of facade elements. Understanding the aspects affecting a facade would improve the reconstruction process of places and allow the generation of variations which exhibit similar patterns. Correctly detecting the features within a facade improves the quality of the modeling results. However, facades are often changed over time by adding or removing items such as flower pots or by opening or closing them. These frequent changes pose many challenges to the detection methods. In this dissertation, the given facade images, either manually captured using a camera or from on-line sources, are analyzed using the proposed semi-automatic framework for investigating such deformations and detecting the repeated elements. The detected structure and the deformation parameter of each element enable novel applications and synthesis possibilities (chapter 3).

The second part of the dissertation considers the interior layout configuration which is an essential part of the reconstruction as it has an impact on the environment functionality. Such configurations could represent the placement of furniture elements, work equipment, material or any items. The interior layout setup has a direct impact on the environment performance, comfort, and ease of navigation. Therefore, its analysis is necessary to understand and evaluate the configuration efficiency. In this dissertation, the primary focus is on the interior configuration which represents storage assignment of stock items in a warehouse. The proposed method optimizes the configuration of items allocation with the goal of improving the order picking processes performance and reducing congestion (chapter 4).

Going on a larger scale gives a better understanding of the urban scene as it would reveal useful information which could be related to the interaction between different elements of the environment, or the perception of the city design. The transportation network (e.g., street network or train network) forms the lifeline of a city and largely defines how the space works and functions. Many factors play dominant roles in determining the efficiency of people's transportation around a city. The last part of this dissertation addresses street networks of cities which are obtained from OpenStreetMap (OSM) [7]. A set of features are proposed to capture the topological and geometric essence of each city. The presented method investigates street networks in a supervised setting with the goal of understanding their behavior and examining which factors have an impact on the city functionality (chapter 5).

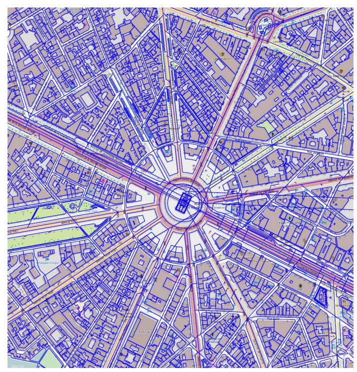


Figure 1.6: A sample street network extported from OpenStreetMap [7].

In summary, this dissertation mainly addresses the following research questions based on the given data type:

Facade images:

- How to analyze a single facade image to identify its structural grid such that the repeated windows are determined?
- How to detect the deformations within the individual repeated windows?
- How to generate new facade structures with simple, smart and intuitive user interaction operations?

Warehouse interior layout:

- What features are suitable to find a storage assignment policy that reduces traffic congestion?

- How to use the proposed measures to evaluate warehouse performance and improve its layout accordingly?

Street networks:

- What are the good descriptors for classifying street networks?

- What are the possible applications using the proposed feature descriptors?

1.2 Contributions

The main contributions of the work performed in this dissertation provide answers to the research questions presented in the previous section. Figure 1.7 shows the contributions to various aspects of urban data analysis. Following is a brief description of the problems addressed in this dissertation.

Chapter 3 proposes a method to analyze facade images and detect the repeated elements (i.e., windows) along with their variations. Facade analysis methods (such as [24, 25, 26, 27]) usually address facades with (near) identical repeated facade elements and may fail in the existence of structured variations. Besides, editing facade images could help to improve their quality and, hence, enhance the modeling efficiency such as removing the occlusions from facade images [28] or generating variations from the original facades [5, 29]. However, existing methods do not consider the intrinsic variations of the elements such as their varying blind positions on windows of being open or (partially) closed which are addressed in this dissertation (as shown in Figure 1.8). The proposed method considers such variations with an attempt to improve the quality of structure detection. After detecting the structure and the elements deformation parameters, the output is used to represent the facade using

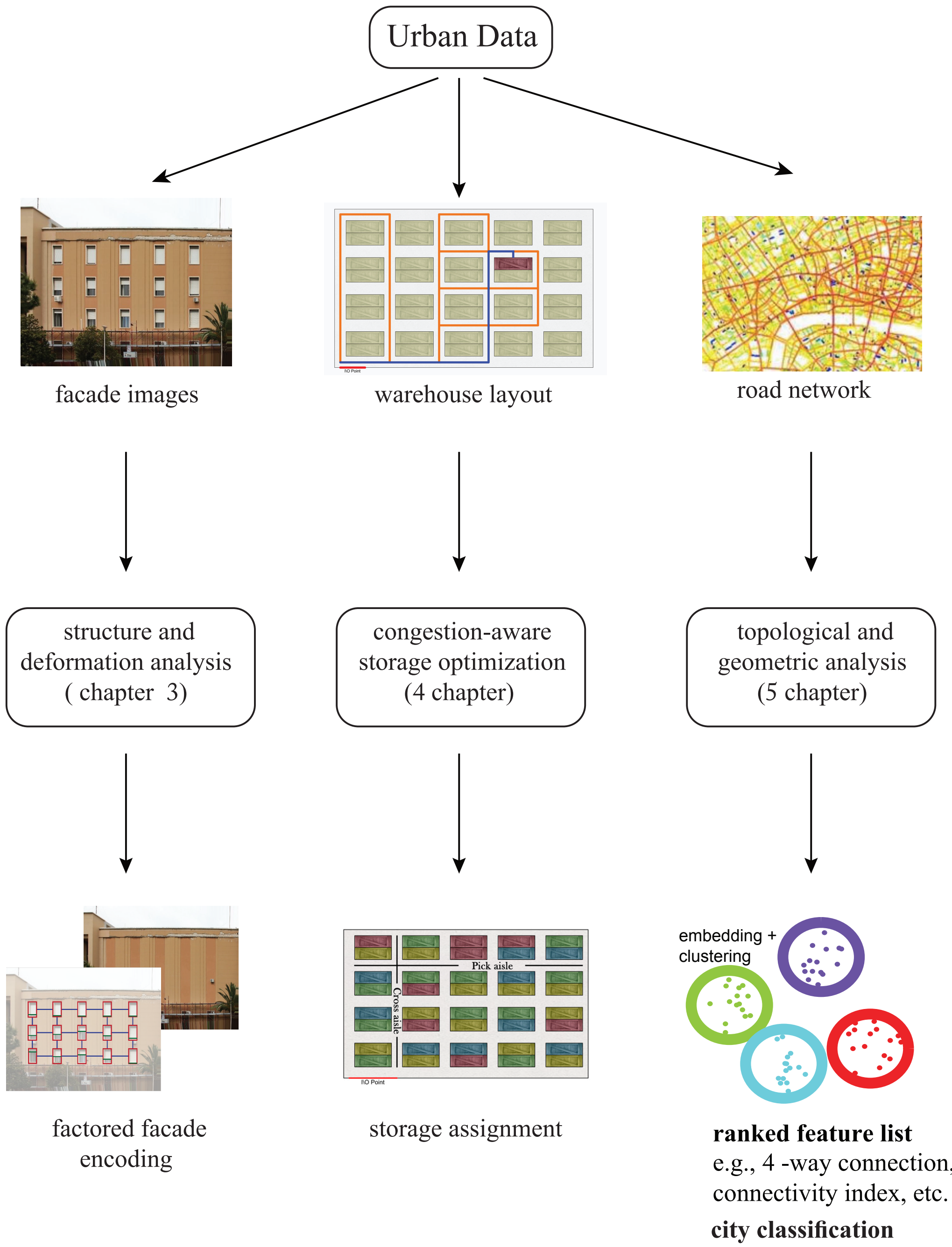


Figure 1.7: *dissertation Outline.* Urban data are analyzed at three different scales. Structure and deformation of facade windows are detected to generate the *factored facade encoding*. Congestion-aware warehouse layout analysis is performed to assign stock to storage locations. The topology and Geometry of road networks are analyzed to understand the behavior of cities.

(a) facade analysis [27] can fail due to element variations.

(b) the proposed algorithm correctly extracts both the underlying repetition pattern and the individual blind positions.

Figure 1.8: Analysis of facade images is often complicated by the variations of the repeated elements (e.g., windows) from the canonical configurations.

different layers in a *Factored Facade Encoding* to encode specifically the window-level low degree of freedom movements. The factored facade representation enables novel interactions to bring static (single) facade images to life. Recently, another layered approach has been proposed to interactively model building facades using multiple overlapping layers of the complete layout [8] as shown in Figure 1.9.

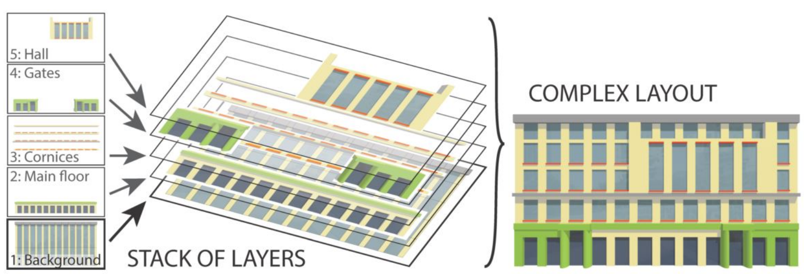


Figure 1.9: Modeling urban facades by decomposing the layout into multiple layers. Figure courtesy [8].

Chapter 4 targets the interior layout configurations and their effect on the efficiency of an urban space. There are many applications for interior layout configurations such as floor plans [30, 31, 32, 6], furniture setup [33, 34] and game level design [35]. This dissertation targets the interior configuration of a warehouse. It proposes a method to find the optimal assignment of stock items to their storage locations in warehouses. The main goal of the optimal assignment is to reduce traffic congestion of moving people and vehicles through considering the alternative paths. The presented algorithm is inspired by the routing algorithm [36] for the on-chip networks which minimizes congestion between each pair of source and destination switches. Having a congestion-aware storage assignment policy greatly enhances the speed of order picking processes, and hence improves the business efficiency. A recent attempt was made by Peng et al. [37] who proposed to create a layout network by specifying how the network should behave using functional specifications and used it in the context of urban street layouts, floor planning, and game level design.

Finally, chapter 5 provides an approach to characterize cities at the street network level based on a set of proposed features. An earlier work addressed the problem from a visual perspective as to distinguish cities based on the visual features representing architectural elements of different places [38]. However, the work presented in this dissertation captures the behavior and functionality of the transportation network in a city rather than its visual features. A similar attempt was done later by Yang et al. [39] who extracts the intersections from a paper map and augments them with GIS information regardless of the original map texture. In procedural modeling, designing an urban plan for street networks requires tuning a set of parameters and optimizing the quality of the network as was addressed by many research works [40, 41, 42]. The work presented in this dissertation proposes a set of descriptors representing both the topological and geometric essence of network layouts. It also discusses the importance of these features as well as their usage in a supervised setting to classify street networks. Moreover, a set of possible applications utilizing these features are considered.

1.3 Scientific Publications

While working on the research of this dissertation, the following scientific papers were published [43, 4, 44]:

- Sawsan AlHalawani, Yong-Liang Yang, Han Liu, and Niloy J. Mitra: **Interactive Facades Analysis and Synthesis of Semi-Regular Facades**. In Eurographics, Computer Graphics Forum, 2013.
- Sawsan AlHalawani and Niloy J. Mitra: **Congestion-Aware Warehouse Flow Analysis and Optimization** Advances in Visual Computing, 2015. (Proceedings of ISVC)
- Sawsan AlHalawani, Yong-Liang Yang, Peter Wonka, and Niloy J. Mitra: **What Makes London Work Like London?** Computer Graphics Forum, 2014. (Proceedings of SGP)

Furthermore, during the research for this dissertation, I contributed to another publication [6]:

- Han Liu, Yong-Liang Yang, Sawsan AlHalawani, and Niloy J. Mitra: **Constraint-aware Interior Layout Exploration for Pre-cast Concrete-based Buildings**. Computer Graphics International, 2013.

Chapter 2

Related Work

With the growing popularity of virtual worlds, a significant amount of work has been conducted demonstrating various methods for modeling real worlds and generating their configurations. Smelik et al. [45] presented a survey of procedural modeling methods that are useful to generate features of virtual worlds. This chapter of the dissertation discusses the works most relevant to the problem of the proposed research.

2.1 Facade Segmentation

Many algorithms have been proposed for detecting the structure of facade data whether images or laser scans. These methods involve local segmentation which is if used alone, not reliable for the detection of general facade structure. Therefore, many new methods involve global consideration of the facade structure such as symmetry and repetitive patterns detection. Moreover, other methods tend to combine low-level processing techniques with high-level techniques such as unsupervised clustering [46] or machine learning [47]. Some of the work that follows these approaches is discussed next.

Symmetry and Repetitive Patterns Detection.

An early work on detecting symmetry in images was done by Liu et al. [48] who proposed a symmetry based method that performs paper cut patterns on images and then creates newly synthesized images with different symmetry groups. Another method for detecting affine symmetry groups in 2D images was introduced by Loy and Elklundh [49] who group feature points to identify and characterize their underlying symmetries. Similarly, there is a good amount of work for detecting symmetries in 3D data such as the work proposed by Mitra et al. [50] who processes geometric models and extracts a compact representation of their Euclidean symmetries. This approach was followed by an algorithm [51] that symmetrizes the geometry of 2D and 3D shapes as well as methods to discover regularities in 3D scans [52, 53]. Another algorithm in this class of work was proposed by Wu et al. [54] which detects the repeated elements under perspective skew at which they rectify the image by finding the elements' salient boundaries. In their follow-up work [55], they used this detected grid of symmetry to reconstruct dense 3D structure.

Nan et al. [56] proposed an interactive tool to discover the underlying architectural structures such as repetitive patterns of a moderate user interaction over a noisy point cloud. The user initializes the boxes which are smart as their sizes and locations are automatically adjusted based on an optimization between a data-fitting term and a contextual term. Aliaga et al. [57] presented a fully automatic approach to segment and detect similarities within an existing 3D architectural model using a weighted minimum set cover over an input triangle soup, and maximizes repetition similarities.

Ceylan et al. [58] introduced a method for image-based 3D buildings reconstruction that relies on symmetry priors. Their extracted symmetry patterns can be used for interactive and intuitive model manipulations. Going a step further beyond the detection of the repeated patterns, Alsisan and Mitra [59] introduced a method to encode different configurations of the repeated facade structural elements by using a Markov Random Field (MRF) optimization to label these elements.

Recently, Shen et al. [25] presented an automatic method to discover high-level facade structures in images and point clouds. Their main observation about these structures is that they are formed by concatenated and/or interlaced grids. This method has the advantage of the adaptive specification of the parameters such as the splitting direction and the number and location of the splitting planes.

Higher-order knowledge-based segmentation.

This section discusses some methods driven by hierarchical rule based segmentation techniques. They subdivide the facade into smaller parts which are arranged based on hierarchical context-free grammar rules.

This approach was followed by Muller et al. [24] who subdivided a simple regular facade into floors and tiles in a synchronized manner. This method reduces the facade into a set of irreducible elements. The result is then turned into procedural rules by using a template grammar to describe the floor pattern. Another example in this category was proposed by Teboul et al. [60] who creates a perceptual interpretation of buildings by performing supervised learning using shape grammar priors and random walks on these learned models. Later, they [27] extended their work to use recursive binary split grammar and reinforcement learning to annotate the facades into elements such as windows, doors, roofs, etc. using training data.

2.2 Facade Modeling

In the seminal work of Debevec et al. [61], they introduced an interactive method that combines image-based modeling with geometry-based modeling. They used line features in images and 3D polyhedral primitives to reconstruct the model with a view-dependent texture mapping. Recovering urban facades from unorganized photo collections using structure-from-motion (SfM) has been also studied (see [62, 63]) using photogrammetric reconstruction and image-based modeling techniques. Xiao et al. [64] proposed a semi-automatic method that uses images captured along the streets and relies on the structure-from-motion to obtain camera positions and initial point cloud for modeling. They proposed an automatic decomposition approach for both analysis and reconstruction. Later in their follow-up work, they provided an automatic system that uses a multi-view image-based approach to generate 3D models [65]. Steinberger et al. [66] presented an approach for the parallel evaluation of procedural shape grammars to generate architectural models on the graphics processing unit (GPU).

Automatic creation of 3D models of high visual quality from a single rectified facade image was targeted by Muller et al. [24]. They presented an algorithm that combines procedural modeling of shape grammars with image analysis to derive a meaningful hierarchical facade subdivision. Detecting repetitive patterns along the axes in orthogonal facade images was proposed by Musialski et al. [67]. Later, Musialski et al. [26] proposed an interactive framework for detailed modeling of building facades from images which is based on coherent based operations. It exploits the partial symmetries across the facade and mixes manual interaction with automatic splitting and grouping operations. Later, this work was utilized by Bao et al. [32] to segment a facade layout and allow to generate procedural design variations which are automatically computed by a combination of heuristic search and quadratic programming. Similarly, editing a 2D building facade was investigated by Dang et al. [29] who proposed a structure preserving editing method which allows the exploration of structural ambiguities.

Using the automatic reconstruction methods does not usually enable generating high quality and accurate models. Improving the quality requires refining the model to get the desired level of detail which can be a tedious task. Therefore, giving the user some control at an early stage can be of a great advantage. An example of an interactive modeling system was presented by Sinha et al. [68] which combines the user drawn outlines on the image with the multi-view information to reconstruct automatically the 3D model. Dang et al. [29] proposed a semi-automatic method to incrementally edit a facade while preserving the detected relations and groupings of the facade elements. In the automatic context, Ning et al. [69] proposed a hierarchical facade reconstruction framework by combining facade structures, detailed windows prorogation, hierarchical model consolidation and contextual semantic representations. Moreover, combining different types of data such as images and 3D scans can be very useful as they are complementary to each other. Li et al. [70] decomposed the facade images into planar fragments and diffused depth information from the corresponding 3D data. Moreover, generating good building layouts was addressed by Bao et al. [32] who encoded the spaces and the transitions of good layouts in a portal graph and allowed the user to explore the plausible layouts.

2.3 Interior Layout Configuration

The configuration of the interior layouts has a direct effect on the urban place efficiency. Researchers approached such arrangements from various angles. For instance, the arrangement of furniture should obey some well-known design guidelines as well as ensure people's comfort. Merrell et al. [33] proposed to arrange furniture based on some interior design guidelines using a hardware-accelerated Monte Carlo sampler for the density function. Yu et al. [34] presented an automatic system to arrange a variety of furniture objects and generate realistic indoor scenes.

Similarly, storage assignment policies in warehouses are vital to determine which product is to be positioned at which location. This was initially addressed by Hausman et al. [71] who proposed a detailed taxonomy of storage location assignment policies. Many rules exist for storage assignment policies including random storage, dedicated storage and class-based storage as explained in Gu et al. [72]. A comparison between these storage methods was conducted by Hausman et al. [71], Graves et al.[73] and Schwarz et al. [74] using both analytical models and simulations. They showed that significant reductions in travel time are obtainable from dedicated storage compared with random storage, and also, that class-based storage with relatively few classes yields travel time reductions that are close to those obtained by dedicated storage.

Layout structure.

Designing interior building layouts was initially treated as packing problem. Galle et al. [30] implemented an exhaustive algorithm to select the rectangular arrangements satisfying constraints among all the possible generations. Wong et al. [31] searched feasible solutions by simulated annealing. Searching among layout possibilities has been treated using crossovers and mutations operators [75, 76, 77]. In these approaches, an evolutionary strategy was used to fit rooms into target envelopes, while improving designed fitness functions. Another approach utilized image processing and procedural generation techniques to automatically generate floor plans from hand-drawn building sketches as was proposed by Camozzato et al. [78]. Moreover, efficient and cost effective construction is important to consider as was addressed by Liu et al. [6] who proposed an exploration method for the interior layouts of precast concrete-based buildings. Moreover, it is not only important to generate a model of the world but to control its details. Vanegas et al. [41] provided an interactive mechanism to edit an urban model using inverse modeling to vary and control the parameters during the modeling process. Another interesting application of interior layout structures is the design of game layouts for various levels with the goal of engaging the user in a visual experience through a sequence of designer-controlled steps. Ma et al. [35] proposed an automatic algorithm to lay out game levels and control the flow through a set of user-specified blocks.

Warehouse layout is a major factor affecting the order picking processes. There are many ways to design a layout. Among the most standard layouts are the ones with parallel storage aisles. Some of them add cross aisles in between the parallel aisles (see Figure 2.1) which provides considerable improvement. Recently, Gue and Meller [10] described the non-traditional aisle designs in which the rules requiring parallel picking aisles and orthogonal cross aisles were relaxed. They developed two designs Flying-V and Fishbone (see Figure 2.2).

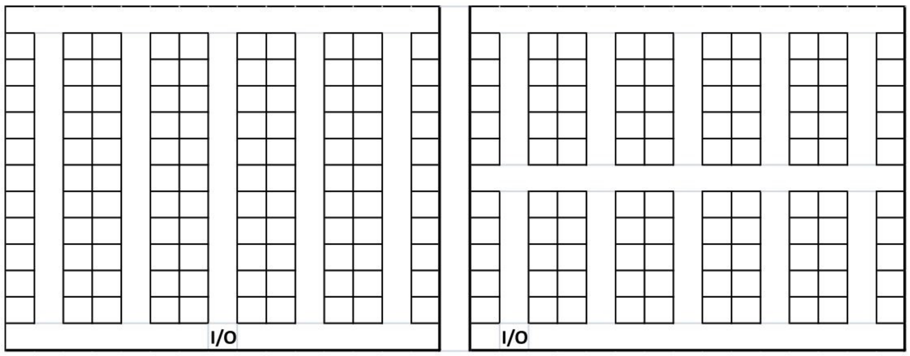


Figure 2.1: An example showing the standard layouts with parallel aisles, figure courtesy [9]

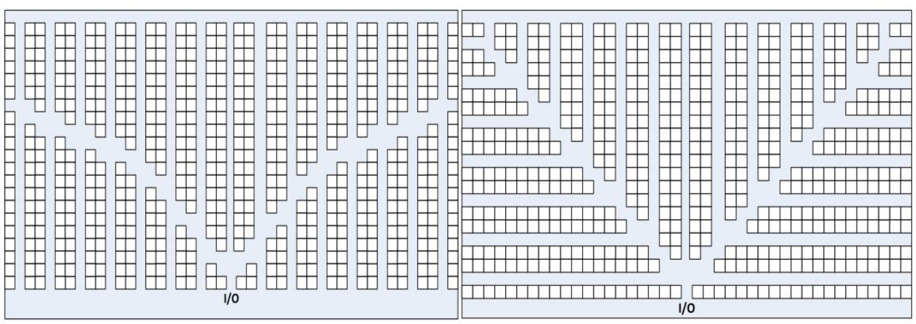


Figure 2.2: An example showing The Flying-V and Fishbone designs introduced by Gue and Meller [10]

Layout analysis.

There are many competing factors that have a great impact on the network performance as was discussed by Gallager [79] who proposed a routing algorithm which achieves the minimum total delay in the network. Aslam et al. [80] learn a congestion model based on real data and develop a congestion aware traffic planning system. When shortest paths are congested, passing through them can deteriorate the network efficiency considerably. A particular example was presented by Ebrahimi

[36] for the on on-chip networks who uses an adaptive routing algorithm to select a less congested path from a broad range of alternatives between each pair of source and destination switches. In chapter 4, a similar concept was adopted to evaluate the reliability of traffic within a warehouse by considering possible redundant routes between the I/O point and the pick locations.

The analysis and optimization of warehouse design have been addressed as in Meller and Gau [81], Meller [82] and Tompkins et al. [83]. Hall [84] analyzed different routing strategies and their impact on order picking efficiency through analytical estimates for travel distance in one block layouts assuming random, uniformly distributed pick locations. Roodbergen and Vis [85] presented an optimization approach for identifying the number and length of aisles and the depot location of the best layout structure with the goal of minimizing the expected average travel distance for the order picking tour. However, the results were limited as they only optimized for layouts consisting of one block. Later, they [86] extended their work to cover arrangements with any number of blocks and any number of aisles. Similarly, Petersen [87] and [88] studied the impact of different routing policies on the layout with the help of simulation. However, he did not provide a general design methodology. Vaughan and Petersen [89] investigated the effects of adding cross aisles to the layout of a warehouse with respect to order picking efficiency.

2.4 Street Network Modeling

This section presents a review of the related work in urban street network modeling and street network analysis in urban planning.

Street network modeling.

Urban networks can be generated by iteratively adding streets to an existing street network [90, 91]. The algorithm can be controlled by setting parameters such as a distribution of angles at an intersection, the street length between two intersections, and snapping distances. Procedural modeling can be combined with interactive techniques to have more control over the result [92, 93]. Yang et al. [42] proposed a framework for street network design that jointly considers the quality of streets and parcels. Another approach to network modeling is data driven. The work of Aliaga et al. [94] investigated the problem of synthesizing urban layouts by example by combining aerial view images of urban areas with vector-based data describing the street and parcel network. Another work of Aliaga et al. [95] supported interactively editing an urban layout by translating, rotating, copying, cutting, and pasting (a group of) tiles to edit or create a new city arrangement. Most related to my dissertation work is the idea to tune the parameters of a procedural model to optimize for particular functional or behavioral characteristics [40, 41, 96]. The modeling of large-scale road networks has been tackled by Galin et al. [97]. At a lower level, Maréchal et al. [98] studied how a single road adapts to the terrain.

Street network analysis and urban planning.

There is a considerable amount of urban planning literature that addresses different aspects of street network planning and analysis. Most literature is concerned with discussing goals and relevant factors on a high level and the descriptions are abstract [99, 100, 101, 102, 103]. The urban planning literature itself rarely includes concrete details on computation for generating new street networks or analyzing existing street networks. The most specific results are typically design standards [104] and some simple statistics used to describe road networks (e.g., the number of city blocks per area). Similar statistics are used in the presented dissertation work. A typical strategy for the analysis in these books is to argue by example. For example, multiple authors categorize street patterns into multiple categories. Each category is explained by one or multiple examples and a textural description. Unfortunately, a lot more information is necessary to perform an actual classification by automatic computation. This problem is also recognized in the literature, e.g., Marshall [103].

Several researchers in urban planning attempted a more formal analysis of street networks. A lot of these papers stem from the seminal work in space syntax [105]. Two popular ideas in this work are to consider visibility and turning angles. For example, researchers conjecture that a person is more likely to walk straight rather than taking turns, and therefore, they augment distance measurements by considering the number of turns a person has to take. The work of this dissertation also integrates an adapted version of reach and directional reach from a recent paper [106]. The study of smaller scale aspects typically falls in the realm of engineering. The study of geometric road details (e.g., highway on-ramps, intersections, turning path of trucks) and the study of local traffic flow (e.g., one vehicle following another in a single lane) have been well investigated and there are a larger number of mathematical models available [107].

Chapter 3

Facade Analysis

Navigating around a city and looking at its components grasp the experience of the city. The street dominates the public experience in almost all modern cities, and it is generally shaped and defined by its surrounding, especially the exterior parts of the urban elements such as building facades. Building facades are integral to most urban environments both real-world and virtual. Different parts define the facade structure such as doors, windows or ornaments. These elements are usually repeated either in regular (e.g., along 2D grids) or semi-regular arrangements. Continuous modifications in these components often occur over time as opening or closing windows, adding secondary elements like flower pots or curtains, etc. Such variations contribute to large appearance changes, often effectively complicating the underlying simplicity of the facades.

The discussion of a facade tackles the exterior part among the scales of an urban environment as shown in Figure 1.5. In order to analyze such information, it is important to collect representative data. A simplified and accurate acquisition of facades has been recently addressed in a variety of directions such as using image-based methods or directly using 3D scanning setups (e.g., LiDAR scans). The analysis of such raw data reveals the underlying repetition patterns and provides new possibilities for compact storage, non-local data completion, or inverse procedural modeling (see recent surveys [108, 109] and references therein). These methods, however, are designed to detect (near) identical repeated facade elements arranged on regular grids and can fail in the scenes with systematic variations like varying blind positions on windows or semi-regular grids (see Figure 1.8).

This part of the dissertation analyzes facade images (on the exterior scale) with the goal of jointly extracting element-level repetitions while factoring out allowed variations (e.g., opening or closing of windows). This analysis enables the decomposition of facade elements and facilitates the extraction of a *factored representation*. This representation enables the encoding for the allowed variation modes which could further facilitate a variety of applications, especially the support of novel editing and synthesis possibilities. The key observation is that element segmentation, repetition (and structure) detection, and solving for the individual element configurations when performed jointly lead to increased accuracy and robustness. Although a full joint optimization is complicated, a simpler iterative solution is proposed to utilize the presence of simple user hints (e.g., marked rectangle in Figure 1.8-b) and simple priors about how windows can typically be manipulated (e.g., window blinds can slide up/down, window shutters can open outwards). Beyond new analysis possibilities leading to significantly compact facade encodings (typically 5-10x compression ratio), the factored representation (see Figure 3.8) reveals the functional workings of the facades and allows facade-syntax preserving manipulations leading to interactive facade images.

The proposed framework is demonstrated on a range of synthetic and real-world images with a varying amount of element variations and clutter while using the obtained factored representations to make the input facades *interactive*. In addition, novel synthesis and manipulation possibilities, both on the input images and on synthesized geometries, are presented towards the goal of meaningful facade manipulations.

In summary, the main contributions are:

- Introducing factored facade representation to specifically encode window-level low degree of freedom movements
- Extracting such encodings from input facade images
- Using the encodings to enable a range of novel interaction to bring static (single) facade images to life

3.1 Overview

The primary goal of this part of the dissertation is to decouple repetitions (regular or semi-regular) from element-level variations in the context of facade images. Figure 3.1 gives an overview of the proposed algorithm. Starting from single input facade images, the analysis phase extracts the repeated facade elements (i.e., windows) along with their variations. Image-space variations among the repeated elements arise from a range of factors such as geometric variations of windows being open/closed; customization due to the addition of flower pots, blinds; appearance changes due to shadow, reflection, and many other factors. In this dissertation, the primary focus is on extracting the geometric changes, which are necessary to enable realistic synthesis. There are two key observations: (i) errors in the local analysis for repetition detection can be rectified using a non-local repetition maximizing optimization, and (ii) having simple deformation priors for 3D window elements leads to robust extraction of the corresponding deformation parameters. The user marks a single window element, and the proposed framework extracts the repetition pattern, per window deformation parameters, and corresponding structured image completion results for the window elements (see Section 3.3). Subsequently, the extracted factored representation is used to enable facade-syntax preserving interaction and synthesis possibilities.

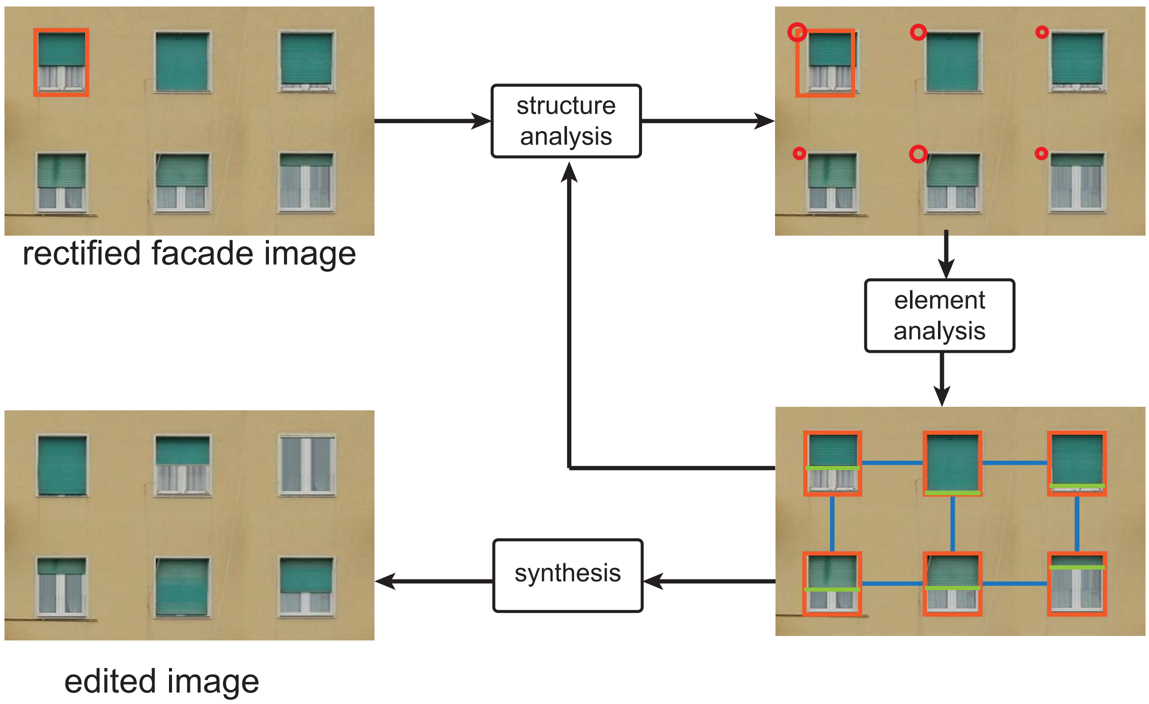


Figure 3.1: An overview of the proposed algorithm which starts from a single rectified facade image with a single marked window (in orange) and then solves an optimization to maximize the regularity of the repetition structure (red circles show local estimates of window locations before global coupling) while recovering the individual deformation parameters of the repeated window elements. The resultant factorization enables novel synthesis possibilities while preserving the extracted facade-syntax.

3.2 Data Collection

The following analysis is performed on a set of images taken from publicly available benchmark facade images, captured from various locations or some synthetic examples generated using a computer software. The first step is to prepare the input image by rectifying it using the vanishing lines which exist in the image using the method proposed by Wu et al. [11]. Then, I extract the Canny edges and discard the small edges which are 2% of the element size. Note that these edges arise from room interiors, window frames, facade textures, etc.

3.3 Repetition Structure Analysis

Current methods detect patterns with the assumption that the repeated elements are similar, and they fail with the existence of window-level variations such as opening or closing of windows (see Figure 3.2). To tackle such variations, the user is allowed to mark a single element of interest W_s (to implicitly identify the scale of interest) and the analysis framework recovers a repetition maximizing regularity, as described next.

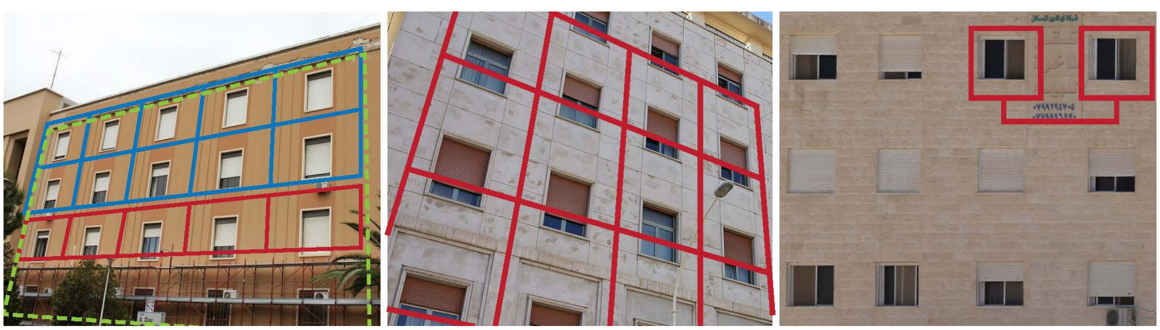


Figure 3.2: The presence of varying configurations of the repeated elements may result in failure to detect facade structure using state-of-the-art algorithms (e.g., [11]), for example, regular grids may get split into multiple grids; detected repetitions may have offset ambiguity, or only small partial grids may be detected. In the interactive framework, the user is expected to indicate one window, while the framework extracts the (semi-regular) repetition pattern for window locations.

3.3.1 Repetition Detection

Using the selected window frame W_s, a search for the candidate repeated windows in the rectified input image is performed using a consistency score that combines intensity and edges similarity. The intensity score $w_{intensity}$ is computed using the Normalized Cross Correlation (NCC) [110] as it is resilient to intensity variations due to lighting and shadow fluctuations.

The edge score for a candidate patch W_t is computed by matching its edges with the edges from the user provided template W_s. I mark a pair of line segments (l_i, l_j) such that $l_i \in W_s$, $l_j \in W_t$ to be matching (i.e., $\text{match}(l_i, l_j) = 1$) if their directions agree within a $\pm 20°$ margin AND the distance between the midpoints of the line segments is smaller than a threshold δ (10% of element size in the implementation). The total edge score is defined as the sum of such matches:

$$w_{edge} := \sum_{l_i \in W_s, l_j \in W_t} \text{match}(l_i, l_j)/|W_s||W_t|. \tag{3.1}$$

Finally, the combined score is defined as: $w_t(W_s, W_t) := w_{intensity}(W_s, W_t) + \lambda w_{edge}(W_s, W_t)$, with $\lambda = 0.5$ in the tests.

It is common in urban facades that windows are aligned horizontally and vertically. Therefore, the first step is to sweep the window vertically starting from the template and compute the confidence of w_t at each point on the line. Then, the algorithm extracts the locations where w_s locally has maximal similarity. Similarly, it obtains the local maxima in the correlation profile along the horizontal direction. Thus, it retrieves a set of $2D$ locations for the candidate repeated elements along with their confidence scores. However, they can overlap and are not yet non-locally coupled (see Figure 3.3).

3.3.2 Repetition Structure Optimization

In the previous step, it is possible to get incorrect candidate positions due to shadows or different interior elements which would affect the intensity similarity while different window configurations might lead to spurious edges similarity. Therefore, an optimization is used to prune the wrong matches and detect the semi-regular pattern of the facade. The main interest is to find a set of consistent splitting lines.

As a first step, the detected candidate window positions are projected along the x and y directions as each direction is handled independently. Let the set of projected candidate locations along one direction be denoted as $P := \{p_1, \dots p_n\}$ with their associated confidence weights $W := \{w_1, \dots w_n\}$. The goal is to select a subset of k splits that best explains the current observations while maximizing regularity. The simulated annealing approach is used to optimize for the set of good splits. A set S contains k points, i.e., the original template position W_s and $k-1$ other points from P. A set of points is considered to be *valid* only if its stencils at the corresponding positions do not overlap.

In order to find the splitting lines that specify the grid containing the windows, a minimization of an energy function is computed as a combination of a data term and

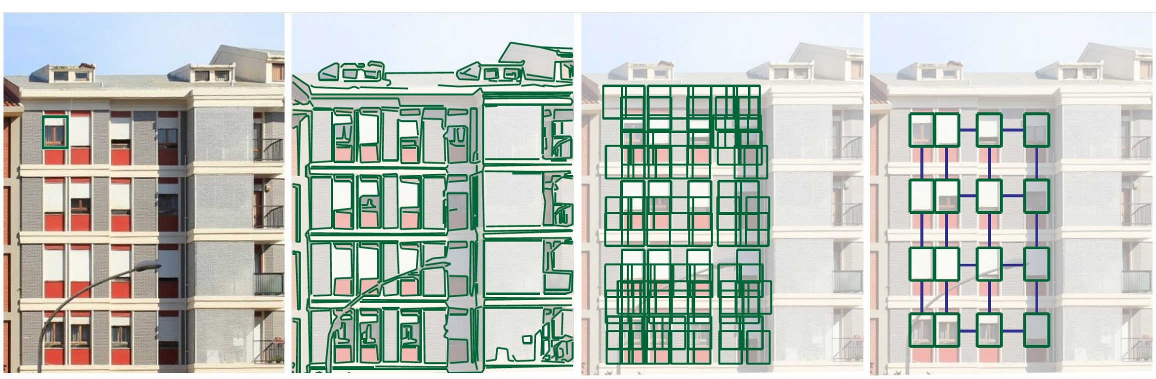


Figure 3.3: (left) Starting from an input (rectified) image and an element proxy W_s (shown in green), the method extracts image edges (middle-left) and identifies potential matches using a combined NCC and edge similarity score (middle-right). Subsequently, it selects a subset of matches in a repetition maximizing optimization (right).

a non-local regularity term defined as follows:

$$\min_{S} E(S) := E_{data}(S) + \lambda E_{reg}(S) \tag{3.2}$$

where λ determines the relative contributions among the data and regularity terms ($\lambda = 0.3$ in the tests). The two terms are defined as follows.

Data term (E_{data}): This term measures the confidence with which a set S covers the candidate set of points P as:

$$E_{data}(S) := \sum_{i=1}^{k} 1/(w_i \cdot g_i) \tag{3.3}$$

where, w_i denotes $w_s(q_i, W_s)$ and g_i measures how well the point $q_i \in S$ agrees with the other points $p_j \in P$ using a Gaussian falloff as:

$$g_i(S_i) := \sum_{p_j \in P} \exp(-\|q_i - p_j\|^2 / 2\sigma^2) \tag{3.4}$$

where σ represents the width or height of the template W_s depending on the dimension of the optimization.

Regularity term (E_{reg}): A typical facade has a near-uniform distribution of distances among its neighboring windows. Therefore, this term is introduced to measure the regularity of the points q_i among the set S. The proposed method finds the spacing distribution among $q_i \in S$ as $\{d_1, d_2, d_3, ..., d_{k-1}\}$ after sorting them by x (or y) values. Then, it measures the regularity of the selection as follows:

$$E_{reg}(S) := \sum_{i=1}^{k-1} (d_i - \mu)^2 + \sum_{i=1}^{k-2} (d_{i+1} - d_i)^2, \tag{3.5}$$

where μ denotes the average spacing. Note that the first term captures deviation from mean separation, while the second term penalizes variations in neighboring deviations.

The optimization starts with $k = n$ and solves for decreasing values of k and

Figure 3.4: Extracted repetition grid in (rectified) facade images under varying regularity, occlusion, and difficult illumination conditions. In each example, the user prescribed one template rectangle W_s (top-left windows in each example).

keeps the first local minima for $E(S)/k$. This identifies a set $S^\star$ of the points that specify the positions of the splitting lines, solved for x and y independently. Note that this approach allows irregularly-spaced partially-filled grid of windows, unlike other methods that solve for full 2-parameter grids [111]. The active windows are marked by looking for points $p_i \in P$ located nearby and have a confidence score larger than 50% of $\max_{W_t} w_t(W_s, W_t)$. Figure 3.4 shows results under varying grid spacing, different occlusion, and challenging illumination conditions.

3.4 Windows Analysis

For each of the detected repeated windows, the proposed method recovers their deformation parameters as described in this section. It makes two central assumptions as follows:

- Some windows have blinds that slide horizontally or vertically to open/close. This type of windows can be analyzed directly on the facade image.
- Some windows have shutters that rotate inward or outward to open/close. Therefore, they need to be analyzed using rough $3D$ information and back-projected on the image to verify the solution.

3.4.1 Blind Window Analysis

The ratio between the area of the blind and the area of the window defines the opening parameter for a sliding window. In the work of this dissertation, the up/down sliding shutters parameters are extracted, while left/right sliding shutters can be analyzed similarly (see Figures 3.5 and 3.6). Let these configurations be denoted as t which determines the status of the blind as follows:

- $t = 0$ denotes open window
- $t = 1$ denotes closed window
- $t \in (0, 1)$ denotes partial cover

As shown in Figure 3.5, for each window element, the first step is to detect edges using Canny edge detector and enumerate the edge pixels per row. The proposed algorithm looks for the rows with the maximum number of pixels to identify a set of candidate horizontal splitting lines. It removes nearby candidates using 3 pixels non-maximal suppression. This step divides each window into vertically aligned sub-regions. The next step is to compute the average color for each of the sub-regions and use it to color the sub-regions. Then, it obtains a mask image composed of the vertical subregions for each window.

Each splitting line in the mask image is associated with two colors; top color c_{top} which represents blind color and bottom color c_{bottom} representing window interior color. In order to select the best color for the blinds c_b and windows interiors c_i, I compute a sum of fitness value over all windows and choose the pair of colors with the best fitness.

The fitness value is calculated for one mask as follows: for each possible splitting line, find the line where the average color of the upper part of the window is as close to the top color and also the bottom part is as close to the bottom color, the lowest difference is used as the fitness value. This exploits the fact that the difference in

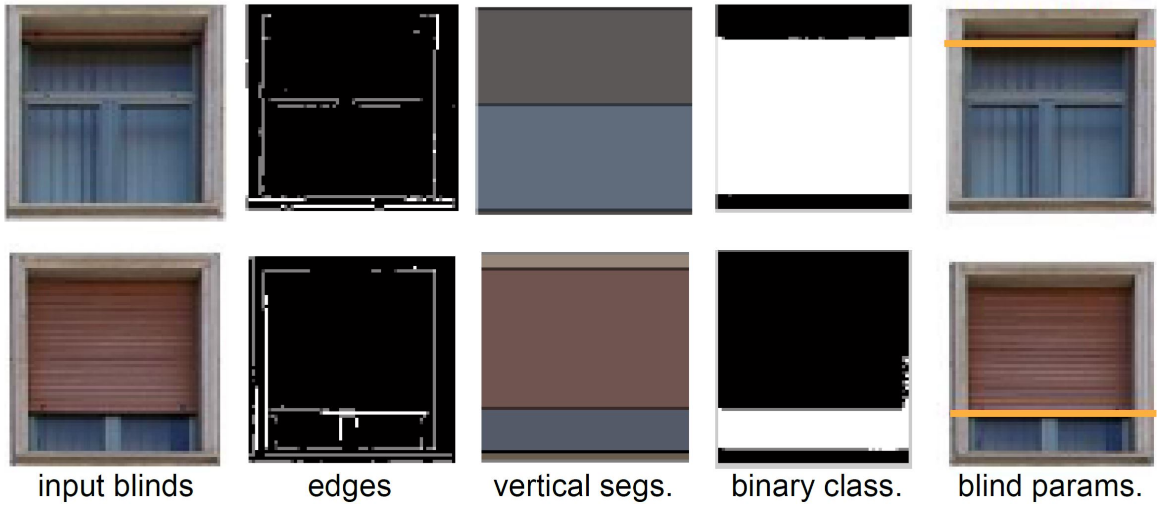


Figure 3.5: Stages of sliding (top-down) window parameter estimation. Starting from window images (first column), Canny edge detection is used to extract dominant horizontal edges (second column) and vertically divide the windows into top and bottom segments (third column). The consistent blind (and interior) colors are determined by comparing the extracted top (and bottom) segments across multiple windows and are used to reclassify the regions for each window (fourth column). This leads to obtaining the respective blind parameter for each window (last column) as a single parameter $t \in [0, 1]$ capturing the extent to which the corresponding window is open.

colors across opaque blinds among the repeated windows is small compared to the difference across the windows interiors which is distinctive for the different windows.

After that, each pixel color c_p is classified as a blind if $|c_b - c_p| \leq |c_i - c_p|$, else as an interior. This forms a binary image where blind parts are black and interiors are white. When the sum of black pixels above and white pixels below reach the maxima, the horizontal splitting line is identified which defines the parameter t (see Figures 3.5 and 3.6).

3.4.2 Shutter Window Analysis

Due to the nature of the rotating shutter windows, their opening parameter is approximated as the rotation angle θ between the shutter and the background of the window frame. The rotation angles θ_l and θ_r are used for the left and the right shutters, respectively. These angles are approximated as follows:

- Half of the window is fully covered (i.e., closed) if $\theta = 0$
- Fully exposed (i.e., open) if $\theta = \pi$
- Partly shaded if $\theta \in (0, \pi)$

Since the shutters rotate off the facade plane, it is important to obtain rough $3D$ information which is approximated using the method proposed by Wu et al. [11]. After having the rough 3D point set, the dominant plane is extracted using the RANSAC-based plane fitting. Then, the corresponding $3D$ placement of the detected $2D$ grid of windows from the previous step are lifted into 3D.

First, it is assumed that the user labeled a fully opened window as the reference window W_s. Then, this window is abstracted in $3D$ by estimating its size and its two shutters in world coordinates as three rectangles (i.e. one rectangle for the window frame and two for its shutters) by projecting the user labeled frame into the approximated facade plane. The abstracted window is denoted as $W_a^{(3)}$.

The abstracted window $W_a^{(3)}$ is translated to align its center with the center of W_t's 3D counterpart. Then, the left shutter is rotated by an angle $\hat{\theta}$ and the $3D$ rectangle is projected onto the $2D$ image which maps the $3D$ shutter rectangle into a quad region $\mathcal{R}$ on the image.

Figure 3.6: An example showing the synthesis of a new window parameter position by appropriately treating the background (i.e., window specific layer) and the blind, which are repeated across other windows. In the case of sliding transparent (e.g., glass) window frames, the user is expected to roughly indicate a mask for the frame.

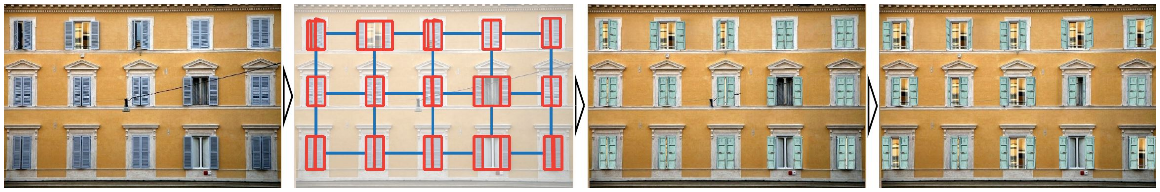


Figure 3.7: Starting from an input image (left), the repetition windows along with their rotation parameters (middle-left) are detected. This allows synthesis possibilites such as replacing the window textures (middle-right) and subsequently virtually changing the window rotation parameters (right).

In order to detect the opening angle of each shutter, the algorithm estimates edges and color consistency as the shutter is rotated in $3D$. First, it performs a line-search (in the space of angles) based $2D - 3D$ matching to find the corresponding θ_l and θ_r. It is important to compare the colors of $\mathcal{R}$ and the left shutter in W_s to evaluate color consistency. This is achieved by estimating the average color of $\mathcal{R}$ as $c_{\mathcal{R}}$ and the average color of the left shutter in W_s as c_{W_s}. Then, the matching score is defined as $area(\mathcal{R})/(|c_{\mathcal{R}} - c_{W_s}| + 1)$. Figure 3.7 shows the opening parameter estimation results of the rotating shutters.

In order to improve the results, the detected positions of the repeated elements along with their configuration estimates are utilized to put all the repeated elements to a canonical configuration and repeat the facade structure analysis and iterate. Typically, performing $2-3$ iterations in the tests could help in improving the results of estimating the positions of the repeated elements as well as their configurations.

3.5 Applications

The analysis of the exterior parts of the buildings (i.e. facades) has a vast set of applications which demonstrates the importance of the proposed techniques for improving the quality of the generated urban models. The framework is evaluated on a large number of images including publicly available benchmark facade images. Following is a discussion of a set of possible applications and evaluation results for the proposed analysis.

3.5.1 Factored Facade Representation

Utilizing the detected windows positions and their configurations allows to efficiently synthesize new facade images, hence, enables a novel encoding for facade images which is called in this dissertation the ”Factored Facade Representation”. An important part of this encoding is to fill in the missing information for the detected facade elements. This step is divided into three categories i.e. the facade background, the blind windows and the shutter windows. These three layers form the factored facade representation which is used to encode the input facade image and later facilitates many edit operations. Following is a more detailed description of each layer.

Background layer

In order to define the background layer for the exterior of a building, I first remove the blinds, shutters or window frames from the image. Then, I use Patch-Match based image completion [112] using structuring guidelines, where vertical information propagation is preferred since windows have a reflective symmetry.

Blind window synthesis

Having the deformation parameters t_i for each of the repeated windows I_i provides an implicit correspondence among all windows. This correspondence is vital to fill in the missing information and complete the blind windows layer. There are two regions with this type of windows which are the blind and the window background areas. The blind regions are repeated; therefore, missing information for one blind can be filled using the information from the other repeated more closed windows. The windows background captures the building interior and differs among the repeated windows. I exploit the implicit correspondence to solve a graph-cut based formulation for image completion using the information across the repeated windows.

Shutter window synthesis

Given the nature of the rotating shutters, the synthesis of the intermediate positions of shutters is approached by sampling a set of textures for the repeated windows for a discrete set of angles, i.e., $\theta_1, \theta_2, \ldots$. Since each window is represented using a window rectangle in 3D, I directly bring the adjacent windows (in space of rotations) to the target configuration θ and interpolate its texture accordingly.

Therefore, each input facade image can be encoded based on the previous synthesis results using the following information (see Figure 3.8):

- the rectifying transform
- the repetition grid of the repeated elements
- the opening parameters for each window along with synthesized open/closed positions
- the background layer

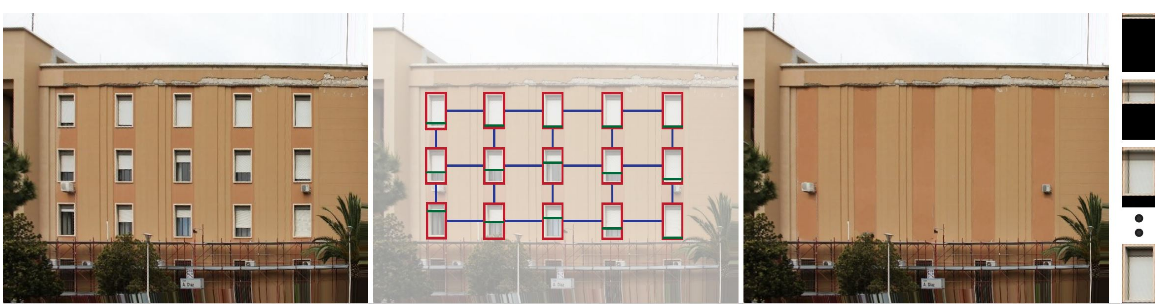


Figure 3.8: Factored facade representation extracted from an input image. (Left-to-right) The input image, the extracted repetition pattern with the individual window positions, the synthesized background layer, and the intermediate blind positions (without synthesized interiors).

3.5.2 Interactive Facade Synthesis

Having the deformation parameter for each window element in the facade allows novel interactive operations. The proposed framework provides an interface to support image synthesis using the factored facade representation. It enables the following operations:

- *Restructuring the facade elements.* The user is allowed to interactively select and drag an element to change its position while maintaining the alignment of other elements in the same row and column. Moreover, the user can change the number of rows or columns of elements to introduce or remove some elements.

- *Opening blind windows.* Since it is known how much each window is open, the user can select a blind and drag its bottom edge up or down to open it or close it.

- *Opening shutter windows.* Similarly, the orientation of shutters can be altered interactively. Note that the previous edits could be done directly on the facade image while changing the shutters' orientation needs to utilize the 3D information. The shutters are rotated in 3D and projected back to the image.

Having these interactive possibilities enables the users to create novel facade images while maintaining the facade structure. For example, a user could synthesize new facade images by swapping their window elements as shown in Figure 3.9.

3.5.3 Multi-View Stereo

The images generated using the interactive framework can be used to improve the performance of other applications. For example, structure-from-motion (SfM) and multi-view stereo (MVS) are popular techniques to reconstruct 3D data from a series of 2D images and have been widely used in the reconstruction of building facades. The

Figure 3.9: Synthetic example generated by utilizing the detected structure and having the factored facade representation. Here, the repetition structure is first identified, and the window elements are then swapped.

quality of the generated models using these methods highly depends on the detection of feature correspondences across the images. Considering facade images, most of the details are found in facade elements such as windows, rather than wall regions which lack sufficient features. Therefore, windows are considered to be suitable candidates for feature correspondence detection. As a result, the variations caused by changes in element parameters pose challenges for SfM based reconstructions.

For example, applying the patch-based MVS algorithm of Furukawa et al. [113] on synthetic scenes with facade element variations results in poor results (see Figure 3.10). A good experiment is to start with 10 input images of a synthetic scene and set the windows configuration randomly. Then, I use the MVS algorithm to extract the dense reconstruction from the images using the known (since synthetic) camera positions. The results show that the $3D$ reconstructed model is weak as image-level correspondence is inaccurate. Instead, I use the interactive framework after detecting the repetition grid and estimating the rotating window parameters to bring all the input images into similar canonical configuration and use them as input to the MSV algorithm to run the dense reconstruction. It is clear that the reconstruction results are significantly better as shown in Figure 3.10. It is expected to achieve similar improvements for the SfM algorithm, which is essential for recovering the

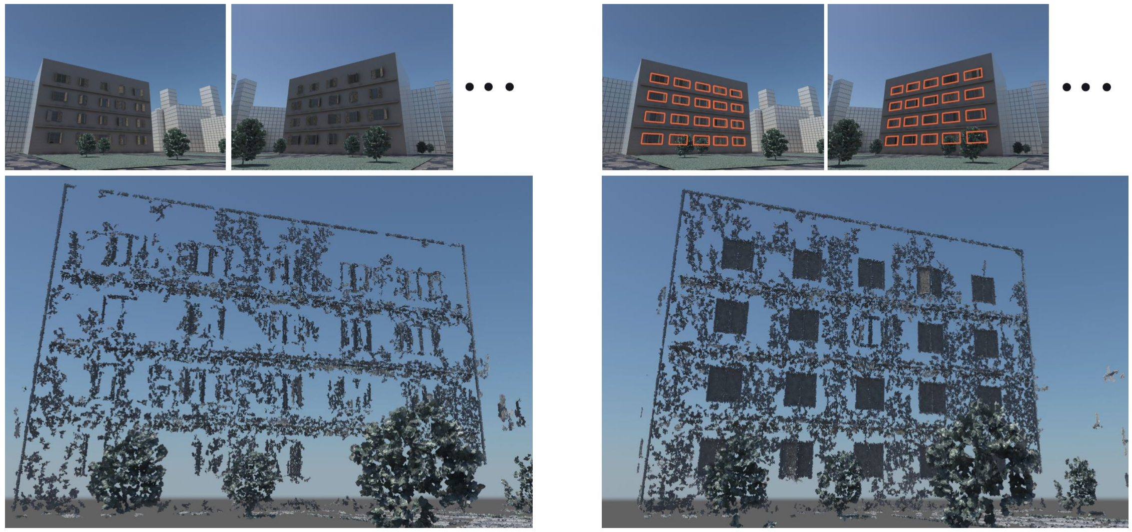


Figure 3.10: This example shows the improvement in the reconstruction quality after bringing all the windows to a canonical positioning using the interactive images. (Left) Using ten input images of a synthetic scene but each window is randomly positioned, to extract dense reconstruction from the images, it is clear that the result is poor as image-level correspondence is imperfect. Instead, the repetition grids are detected in each image, and the rough 3D points from the previous stage are used to help estimate the rotating windows parameters. Subsequently, all the windows are synthetically positioned to a canonical position (as in the right image) and dense reconstruction is performed. As a result, the recovered 3D scene has a much-improved resolution.

transformations between the images in a multi-view setting.

3.6 Conclusion

This chapter presented the first part of the urban data analysis investigated in this dissertation which considers the exterior part of a building i.e. its facade. It proposed a semi-automatic algorithm to analyze input facade images to extract their repeated windows regularity pattern while also identifying the corresponding windows deformations (i.e., the displacement parameters for sliding windows and the angle parameters for rotating windows). It also presented the factored facade representation which is very useful to support novel interactive edit operations. The proposed algorithm is evaluated on various challenging facade images to demonstrate its improvement over some of the well-known facade analysis algorithms. The evaluation showed a range of novel facade manipulation possibilities, both on the input images and on synthesized 3D geometry while preserving the extracted facade-syntax.

3.6.1 Future Work

It is possible that the initial repetition detection algorithm could fail due to the presence of large-scale window element variations which prevents the algorithm to proceed. Moreover, significant light/shadow changes, or presence of outlier factors, can lead to failure in abstracting the window variations or even missing the repetition grids. Therefore, an interesting research avenue would be to investigate a more tightly coupled repetition detection and window variation estimation to address these issues. Moreover, the algorithm uses approximated 3D geometries which can be further utilized towards more certain appearance modeling effects under known light and camera settings (see also [114]). Finally, the factored facade representation can be used to reveal interesting image parts which cannot be explained directly by the extracted windows templates, leading to novel image understanding possibilities (see Figure 3.11 for an initial result).

Figure 3.11: Given the encoded factored facades for window elements (top) and based on the allowed frame movements, it is possible to prune out some edges from the original edge maps (middle). This extraction can potentially reveal interesting parts. For example, considering the bottom region of a window helps to identify regions of interest (flower pots) as unique features compared to the window template (bottom).

Chapter 4

Interior Layout Analysis

The continuous experience of any virtual environment application is necessary as the user desires to feel immersed in the scene. As the previous chapter addressed the analysis of the exterior facade elements towards a realistic modeling of buildings, it is also vital to analyze the interior configuration of urban spaces (refer to figure 1.5) towards layout setup evaluation and improvement. Examples of interior layout configurations could be floor plans, furniture setup, game levels, items assignment, etc. It is well-known how the placement of objects within an environment plays a vital role in its functionality. The arrangement of elements in any setting should provide comfort as well as enhance the effectiveness of the place. For instance, the arrangement of furniture in houses, offices or classrooms impacts the performance of people in these locations. Moreover, the arrangement of items and products in showrooms or warehouses directly influences the effectiveness of the work processes. Therefore, it is important to pay real consideration to the problem of positioning objects in environments.

Considering warehouses, the placement of objects within a warehouse has a critical impact on the customer service levels and logistics costs. Intuitively, warehouses are designed to be cost effective. There are many typical warehouse layout considerations such as its capacity, utilization of resources, throughput, and resource constraints. Managers usually aim to achieve the optimum layout which reduces material handling costs, minimizes space requirements, and lowers energy bills. In warehouse design, there are many techniques to approach the assignment of stock to storage locations. *Randomized storage* policy randomly assigns each item to an available position with equal probability. This method guarantees the highest minimization of the needed slots but does not consider material handling and requires a sound system to record the variable positions of products. *Dedicated storage* is to assign a precise number of slots for each product which ensures a smooth tracking but results in a wasted space with the existence of the seasonal demand goods as the assigned slots can not be reused. *Class-based (ABC) storage* policy assigns the most frequently requested items closest to the input and output point where the loading and unloading happens.

This part of the dissertation analyzes the layout configuration of objects within a warehouse with the aim of increasing its efficiency. Many competing factors need to be explored. One important factor that profoundly influences the costs in a warehouse is the speed of processing items. Objects should be placed close to the input and output point to achieve quick processing of requests. However, this can lead to congestion as the density of moving people and carriers near the I/O point will be high and they will start blocking each other. Hence, it is important to investigate a right balance between the processing speed and the resulted congestion. The work presented in this part of the dissertation achieved a better performance compared to the commonly used storage assignment solutions (in warehouse design). The main goal is to harvest the advantages of two well-known storage assignment policies. Starting with a random assignment prevents the wasted storage drawback of having seasonal items while using the class-based storage assignment provides fast processing of requests.

Given a warehouse layout with the number of elements to be stored, the primary goal is to find the best storage assignment of the items that improves the warehouse efficiency by lowering down the congestion of moving vehicles within a warehouse. Clearly, having a lower congestion level will have a great positive impact on the order

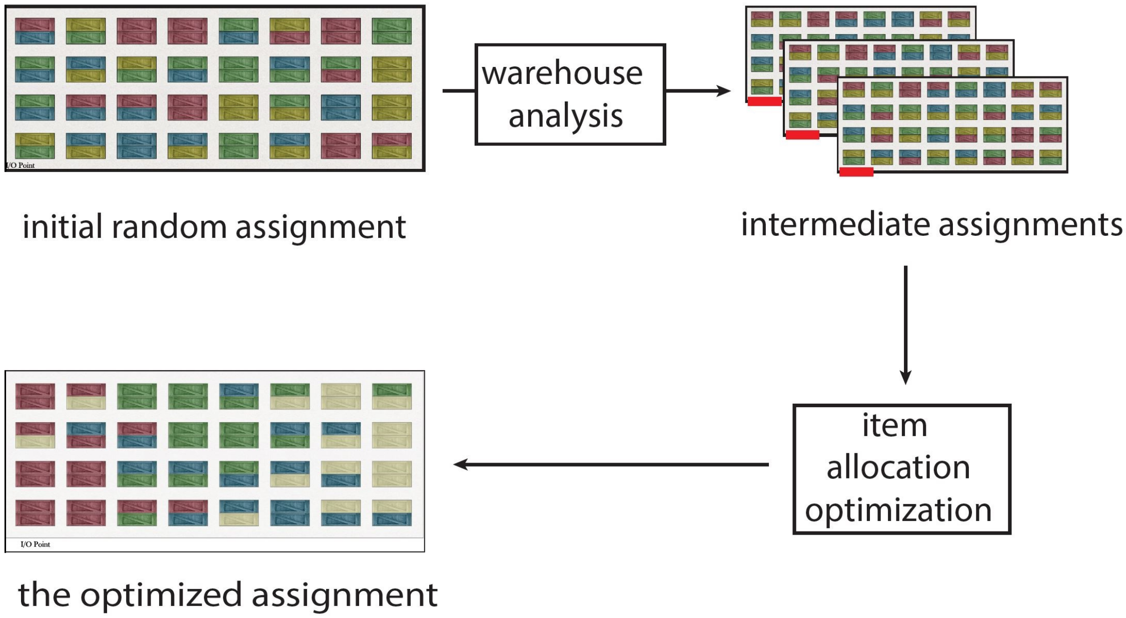


Figure 4.1: An overview of the analysis pipeline of the interior warehouse configurations. First, the storage assignment is randomly initialized. Then, the algorithm optimizes the storage assignment to minimize traffic congestion and ensure fast processing of orders. The proposed methodology enables novel warehouse analysis and synthesis possibilities.

picking processes within the warehouse. The first part of the proposed algorithm is to analyze traffic flow within a warehouse and then optimize to find the optimal storage assignment that enhances the traffic flow. There are two main questions of interest which are as follows:

- What are the suitable features to find a storage assignment policy that reduces traffic congestion in the warehouse aisles?
- What are the possible applications using the proposed method?

4.1 Overview

The primary target of this part of the dissertation is to investigate the possibility to configure a storage layout of a warehouse while keeping congestion as minimum as possible. Figure 4.1 gives an overview of the proposed work. First, the storage positions are initialized with a random assignment of different item types. Then, by considering the possible redundant routes, a probabilistic metric is evaluated to reflect the speed of processing requests. The algorithm also proposes a measure to approximate congestion level in the individual aisles as well as in the whole warehouse. Then, these metrics are used to perform congestion-aware optimization using Simulated Annealing (SA) based sampling to find the best items layout. Using the optimal elements arrangement improves the order picking processes speed and minimizes congestion. Subsequently, the proposed metrics are useful to evaluate the performance of different warehouse layouts and investigate the possibility of synthesizing better ones.

4.2 Warehouse Layout Structure

This section describes the structure of the order picking area in a warehouse. Typically, order pickers drive through warehouse aisles to retrieve products from their storage locations. Figure 4.2 shows various aspects of the layout of order picking area. Several pick aisles have racks on both sides in which to store products. Changing from one pick aisle to another is possible through the cross aisles, which are perpendicular to the pick aisles. At least, there should be two cross aisles at the borders of the warehouse. These cross aisles do not contain pick locations. Adding more cross aisles increases the number of possible routes within a warehouse. The main advantage of having extra cross aisles in a warehouse is the larger number of routing options, which may result in lower travel distances [89]. However, the size of the warehouse will inevitably increase when adding more cross aisles, because the total storage area must be kept constant to meet the predefined requirements.

Each warehouse layout is composed of a set of aisles and a set of storage shelves.

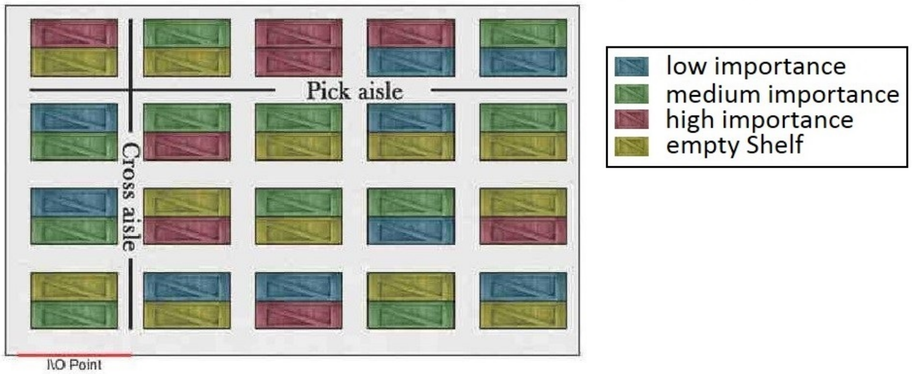


Figure 4.2: A schematic top view of a typical order picking area in a warehouse. There are five pick aisles and six cross aisles. Shelves are depicted in random colors based on the assigned item type which reflects their demand importance.

The aisles are defined using a set of nodes (i.e., intersections) $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ described by their 2D locations, together with the individual aisles connections $\{e_1, e_2, ..., e_M\}$. Based on the node set and the connectivity of individual aisles, I construct a graph $G = \{V, E\}$ for the entire warehouse layout. The vertex set is defined as $V := \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ and the edge set as $E := \{e_{ij}\}$ where $e_{ij} = \overline{\mathbf{v}_i \mathbf{v}_j}$ denotes an aisle segment.

Each side of a picking aisle contains storage shelves, and it is assumed only to have one rack of shelves. The set of storage shelves is defined as $S := \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_L\}$ where each node is assigned a type $t \in T$ where T is the set of item types to be stored in the warehouse. Moreover, based on the assigned type, it is associated with a demand $d \in D$ to approximate the importance of the item stored in the location. The algorithm assumes having three types; the most, medium and lowest demanded items. In order to access the storage shelves, each node is projected to the nearest aisle from the set E. Another assumption is having a single input/output (from now on I/O point) for the loading and unloading of items which is the bottom left corner in the provided examples.

Figure 4.2 shows a case of a warehouse layout that is a 5x6 grid of aisles. Storage shelves are shown as rectangles with random color assignment based on the stored item type. Red shelves represent the most demanded items with high demand d value, green for the medium demand and blue for the least demanded items. Yellow shelves represent the empty ones.

4.3 Congestion-Aware Optimization

This section describes the analysis of a warehouse to find the best items allocation that enhances the traffic flow within a warehouse. Since there are items that are more vital to the business than others, it is common and intuitive to place the most important items closer to the I/O point. However, this will lead to more congestion within the warehouse as more items are concentrated in one region. Therefore, the proposed congestion-aware analysis investigates an efficient method to improve the warehouse reliability by specifying storage locations and allocating their types.

In order for the proposed method to evaluate the traffic congestion, it needs to consider the possible redundant routes. Having a variety of routes indicates the increased possibility to change the path and, hence, reduce or avoid traffic. Therefore, the proposed method first defines the shortest paths from the I/O point to each of the storage shelves. Then, within twice the shortest distance, it enumerates their H possible redundant routes. It defines the set $R := \{R_1, R_2, ..., R_L\}$ where R_i denotes H possible redundant routes for each storage location $\mathbf{s}_i$ such that $R_i := \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_H\}$, where $\mathbf{r}_j$ is a single route composed of a sequence of edges that belong to the aisles edges E. Figure 4.3 shows a set of possible redundant routes between the I/O point and a selected storage shelf.

4.3.1 Warehouse Flow Analysis

The primary goal is to configure the items storage layout such that their flow meets the following features: (a) quick access and delivery of items relative to their importance,

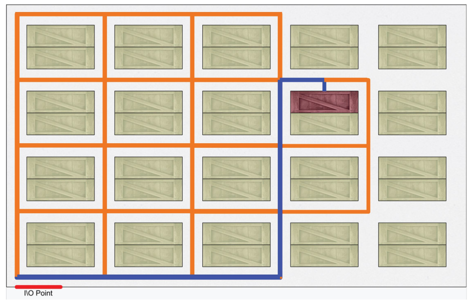


Figure 4.3: Having a variety of redundant routes reduces the probability of congestion. This figure shows H possible redundant routes between the I/O point and a selected storage shelf $\mathbf{s}_i$ (in red). The blue route represents the shortest path.

and to have (b) reliable traffic within the warehouse with the minimum congestion. The following terms are computed to evaluate these features.

Processing speed term. This metric measures the rate of processing items with respect to their demand rate. It favors the most demanded items to be closer to the I/O point. The processing speed energy term is computed as the sum product of the demand value d_j for type t_j and the average shortest distance to reach all storage shelves of type t_j which is expressed as follows:

$$E_{processing} = \sum_{j;t_j \in T} d_j * \frac{\sum_{i;t_i=t_j} shortest\ distance_i}{n\ elements\ of\ type\ t_j} \tag{4.1}$$

where d_j is the demand rate for type t_j.

Dispersion term. It is important to consider the distribution of items, especially items of the same type. In order to maximize the convenience and improve the speed of processing items, it is better to distribute similar items around the same region. This distribution is especially important when handling requests of large quantities as the carrier does not need to travel a greater distance around the warehouse to complete the task. In order to achieve this feature, dispersion term is computed based on the distribution of items of each type as follows:

$$E_{dispersion} = \frac{\sum_{j=1}^{n\ types} \frac{1}{n} \sum_{i=1}^{n\ elements\ of\ type\ t_j} ||z_i - c_j||}{n\ types} \tag{4.2}$$

where c_j is the centroid of z_i points computed for each type t_j and $||z_i - c_j|| = \sqrt{(x_i - c_1)^2 + (y_i - c_2)^2}$

Congestion term. This term addresses the core of the proposed algorithm which is to ensure having the minimum congestion within the warehouse. Approximating congestion is approached by evaluating the reliability of traffic. The algorithm considers the layout graph as a compound system of parallel and serial components as described in [115]. Each route along the aisles between the I/O point and the storage shelf s_i is composed of a series of edges while the multiple redundant routes are the parallel components. The first step is to compute edge probability for each edge $e_k \in E$ in the graph based on the number of all redundant paths K in R that use the edge such that:

$$p\_edge_k = \frac{1}{K} \tag{4.3}$$

Then, using the series component probability of each aisle being composed of many edges connected in series, the probability of each redundant route is evaluated as follows:

$$p\_route_j = \prod_{k \in r_j} p_k \tag{4.4}$$

By considering the redundant routes $R_i := \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_H\}$ for each storage shelf s_i as a parallel system, it is possible to find congestion rate as follows:

$$E_{congestion} = 1 - \prod_{r_j \in R_i} (1 - p\_route_j) \tag{4.5}$$

4.3.2 Storage Assignment Optimization

In order to evaluate the traffic flow efficiency of a warehouse, it is useful to combine the previously described evaluation metrics in an energy function as follows:

$$E = \lambda(E_{processing} + E_{dispersion}) + (1 - \lambda)E_{congestion} \tag{4.6}$$

where λ determines the relative contributions of the terms.

Minimizing the energy function given in equation 4.6 using a simulated annealing (SA) based sampling [116] approach helps to find the best items allocation. Initially, the optimization starts with a random allocation and a random type assignment. It sets $E \leftarrow \infty$ and $T = 500$. In each SA step, it randomly selects two items and swaps their types. It accepts the new solution with energy $E(S)$ if $E_{new} \leq E(S)$; else it accepts the new solution with a probability of $exp(-(E_{new} - E(S))/t)$, where t is the temperature; else it rejects the new layout and retains the old one. If it accepts a solution, it sets $E \leftarrow E_{new}$. In the annealing schedule, it reduces temperature t and continues with the iterations. It stops if either the maximum number of steps ($500 - 1000$ in the evaluation tests) has been reached, or when $E < threshold$. An example of the results with the intermediate steps is shown in figure 4.1.

4.4 Evaluation and Applications

In order to demonstrate the utility of the proposed method, various evaluations are demonstrated on warehouses of varying sizes and layouts. The results show that the framework is simple and yet useful for the analysis of warehouses and their order picking processes. The proposed analysis provides a better understanding of the warehouse layout and suggests an efficient generation of storage assignments that maximize warehouses reliability and productivity. The following part of the dissertation discusses different evaluation aspects and demonstrates some of the possible applications.

Processing speed analysis. As discussed earlier, it is common to store the most demanded items, which are commonly ordered more often, closer to the I/O point. The proposed processing speed term (refer to Equation 4.1) can be used to evaluate the speed of processing items relative to their demand rate. It decreases as the most demanded items are nearer to the I/O point while the least demanded items are further which implies a faster overall order picking process. Therefore, it is helpful to use this term to evaluate the processing speed of order picking processes for a given storage assignment policy.

Moreover, comparing the proposed assignment approach against some of the well-known storage assignment policies would give a good evaluation of its performance. Figure 4.4 shows an example of these assignments on an 8x8 warehouse layout. The used storage assignments are random, class-based, opposite class-based and the optimized assignment. Note that the opposite class-based assignment is not a standard method; however, it is used to emphasize the effect of items' positioning on the speed of processing. The evaluation addresses the proposed processing speed metric on these assignments and shows that the proposed method has the least value which implies the fastest processing of items with an improvement of around 20-25% as presented in figure 4.5.

Furthermore, it is important to evaluate the time and distance needed to process picking the items. Similarly, the proposed method is compared with the well-known methods for the warehouse in Figure 4.4 as shown in the tables 4.1 and 4.2. The optimized congestion-aware assignment provides a time improvement of around 20-30% and a distance improvement of around 15-25% (see Figures 4.6 and 4.7).

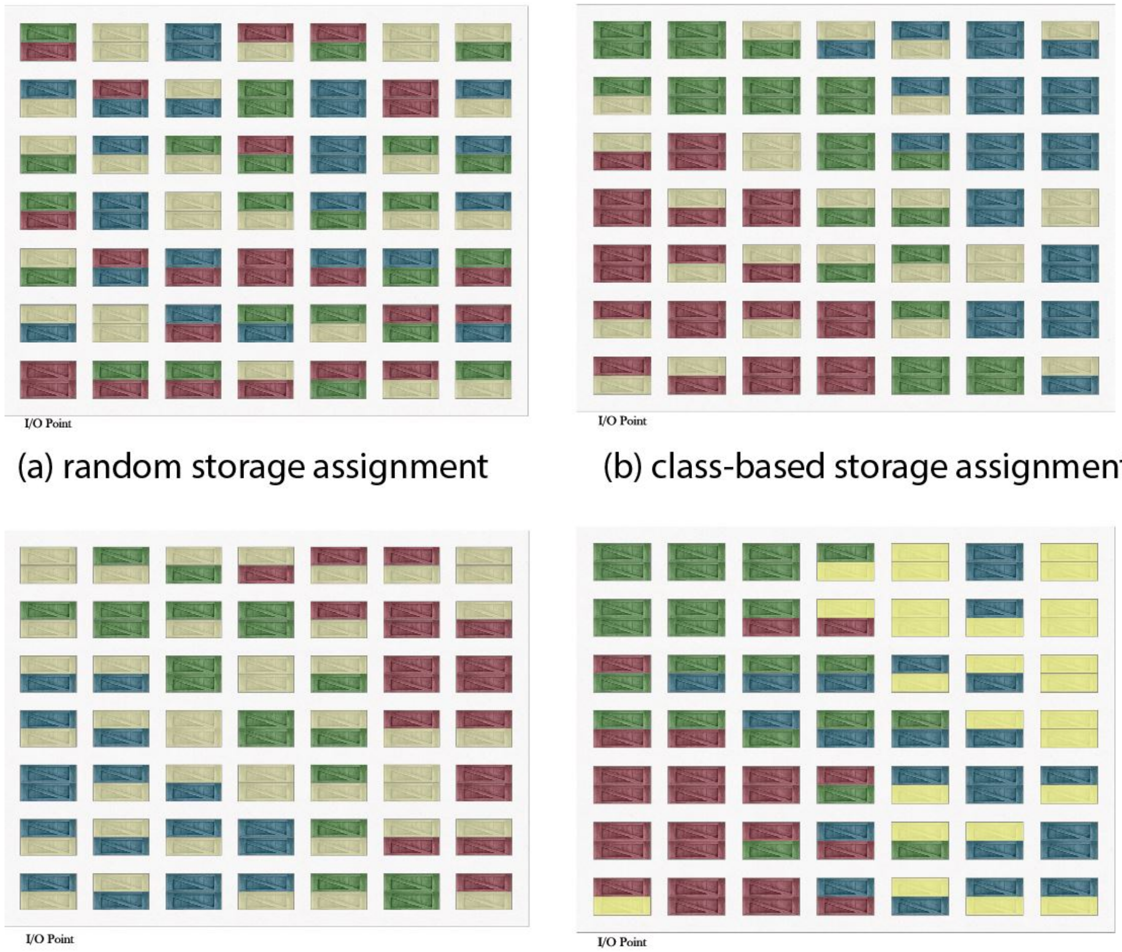
Congestion analysis. This study compares the proposed storage assignment with some of the well-known policies. It focuses on the random assignment and class-based assignment policies. The results in figure 4.8 show that the proposed assignment reduces congestion level in a warehouse. The random and class-based storage assignment methods have comparable congestion rates as they both depend on the randomness of their assignment without considering the traffic flow within a warehouse. The proposed method achieves a more reliable warehouse that has lower congestion as it harvests the benefits of class-based assignment using the processing

Table 4.1: A comparison of the travel *time* needed to reach the items of different item types in a warehouse of an 8x8 size using various assignment methods shown in Figure 4.4. Refer to Figure 4.6 for a visualization of this comparison.

Assigned type	**Random**	**Class-based**	**Opposite class-based**	**Congestion-aware**
t = 1	11.4	6.16	15.98	7.67
t = 2	12.09	10.61	10.47	7.55
t = 3	11.84	17.4	5.31	10.96
Average time for all types	11.78	11.39	10.59	8.73

Table 4.2: A comparison of the travel *distance* needed to reach the items of different item types in a warehouse of an 8x8 size using various assignment methods shown in Figure 4.4. See Figure 4.7 for a visualization of this comparison.

Assigned type	**Random**	**Class-based**	**Opposite class-based**	**Congestion-aware**
t = 1	516.84	295.8	680	312.63
t = 2	561.05	505.263	495	426.31
t = 3	527.36	746.31	244.21	573.68
Average distance for all types	535.08	515.79	473.07	437.54



(a) random storage assignment (b) class-based storage assignment

(c) opposite class-based storage assignment (b) congestion-aware storage assignment

Figure 4.4: Different storage assignment policies used to allocate items in an 8x8 warehouse. Class-based assignment (b) ensures having the most demanded items (in red) closest to the I/O point while the opposite class-based assignment keeps the least demanded items (in blue) closer to the I/O point. The proposed congestion-aware assignment optimizes the allocation of elements to achieve the fastest processing speed. As shown in Figure 4.5, the optimal assignment has the fastest processing speed value. Note: red, green and blue shelf colors represent the most, medium and lowest demanded items, respectively. Yellow locations represent empty shelves.

term.

Traffic flow analysis. The proposed framework can be used to analyze congestion levels based on the value computed using equation 4.3. The simulator renders the aisles with different colors based on the link probability at each aisle edge. Figure

Figure 4.5: An evaluation of the processing speed metric for different storage assignment policies. The proposed optimal assignment has the fastest processing speed value (around 20-25% improvement).

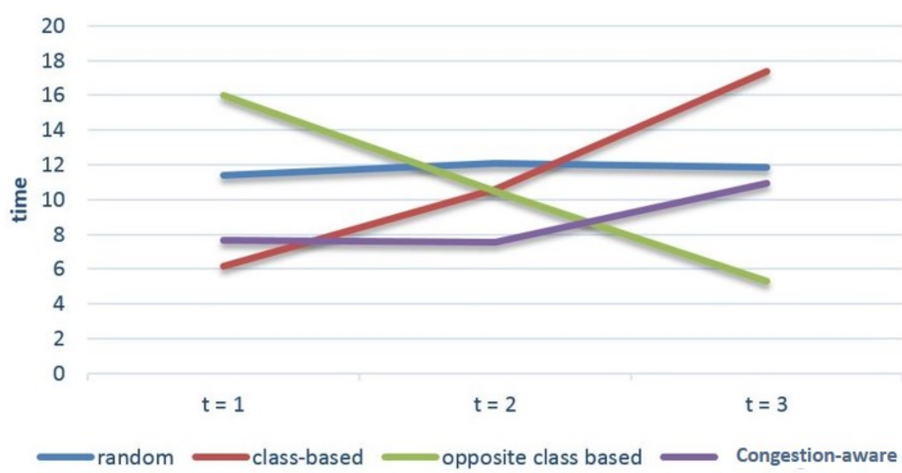


Figure 4.6: Different storage assignments imply varying time durations to perform the picking operations. The plot shows the time taken to reach items of different types i.e., $t = 1$, $t = 2$ and $t = 3$ which is a reflection of table 4.1. The optimal congestion-aware assignment shows a time improvement of around 20-30%.

4.9 shows that the optimized storage assignment reduces the number of congested aisle edges as layout (c) has the least number of red edges. Moreover, the overall congestion score is lower than the two common storage assignments.

Cross aisles evaluation. One factor affecting the efficiency of processes in a

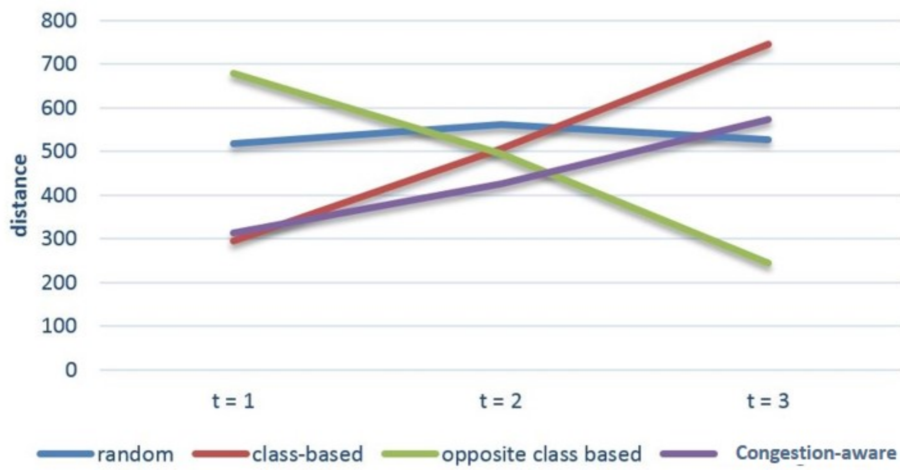
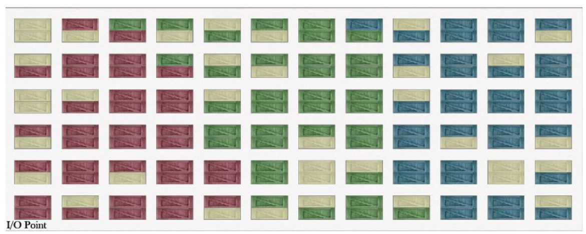


Figure 4.7: A comparison of distance performance of different storage assignment policies (shown in Figure 4.4) for three different item types i.e., $t = 1$, $t = 2$ and $t = 3$. The optimal congestion-aware assignment shows 15-25% distance improvement.

warehouse is the number of cross aisles in its layout structure. Adding more cross aisles increases the number of different possible routes in a warehouse which in turn reduces the possibility of having congested aisles. The congestion score (equation 4.5) evaluates the reliability of a warehouse. The result shows that increasing the number of cross aisles increases the number of different possible routes and reduces the probability of having congestion in the aisles which implies higher warehouse reliability and efficiency. Figure 4.10 shows a demonstration of these results. In this evaluation, the storage assignment was used with $\lambda = 0.2$.

Proposed metrics evaluation. The proposed method captures the essence of warehouses' efficiency using two terms i.e., the proximity of items to the I/O point leading to faster processing 4.1 and the congestion level within the warehouse aisles 4.5. In the proposed optimization 4.6, the relative contribution of each term was determined using the λ term. Increasing λ improves the time and distance needed to perform the order picking tasks as the contribution of the processing speed term increases (refer to Figure 4.11). Moreover, the performance improves by around $7-10\%$ over the class-based method which shows that the proposed method improves

(a) Random storage assignment
Congestion rate = 0.0098022



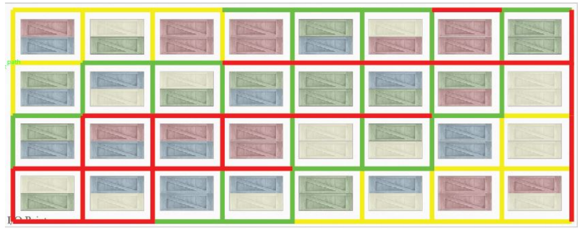
(b) Class-based storage assignment
Congestion rate = 0.0095065

(c) Congestion-aware storage assignment
Congestion rate = 0.00000018

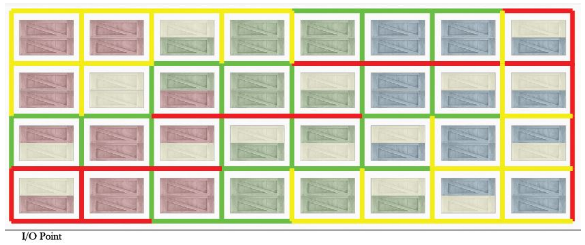
Figure 4.8: This example shows a comparison between the two standard storage methods and the proposed method. The used warehouse is composed of a 7x13 structure. Apparently, congestion rate is reduced using the optimized assignment method (layout c). Note: red, green and blue shelf colors represent the most, medium and lowest demanded items, respectively. Yellow locations represent empty shelves.

the performance even without considering the items proximity to the I/O point.

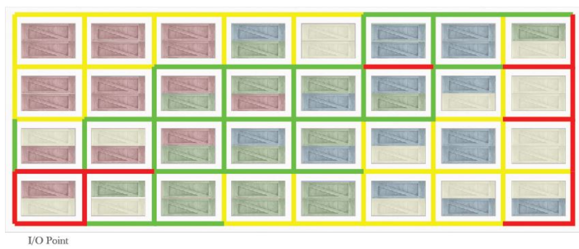
Congestion-aware layout synthesis. An important goal of procedural modeling is to provide solutions towards the generation of new layouts or improving the existing ones. In the context of warehouse evaluation, it would be useful to utilize the proposed metrics and investigate the possibilities of synthesizing better warehouses. Cross aisles have the greatest impact on warehouses reliability as they provide a larger variety of possible routes and, hence, reduce congestion. Therefore, the synthesis starts with a layout that has only cross aisles at the borders but many pick



(a) Random storage assignment
Congestion rate = 0.017328



(b) Class-based storage assignment
Congestion rate = 0.019672



(c) Congestion-aware storage assignment
Congestion rate = 0.0000324

Figure 4.9: Visualization of the traffic flow for different storage assignment policies. Different colors represent the congestion level at the respective edge; red, green and yellow edges denote high, medium and low congestion levels, respectively. The optimized assignment was generated by setting $\lambda = 0.2$ as to achieve the maximum reliability possible. The optimized result (layout c) has the least number of red (most congested) edges.

aisles. The items are allocated to their storage locations using the proposed method. The synthesis starts by randomly introducing new edges which represent parts of cross aisles. Then, congestion rate is evaluated and minimized to find the best layout which increases the warehouse reliability. Figure 4.12 shows the result of this synthesis.

(a) 5x5 layout
congestion rate = 8.6e-5

(b) 5x9 layout
congestion rate = 1.9e-7

Figure 4.10: Reducing the number of cross aisles raises the probability of having congestion in traffic flow within the warehouse. Evaluating congestion score for a warehouse (a) with five cross aisles and a warehouse (b) with nine cross aisles shows the decrease in congestion after adding more aisles. Note: red, green and blue shelf colors represent the most, medium and lowest demanded items, respectively. Yellow locations represent empty shelves.

4.5 Conclusion

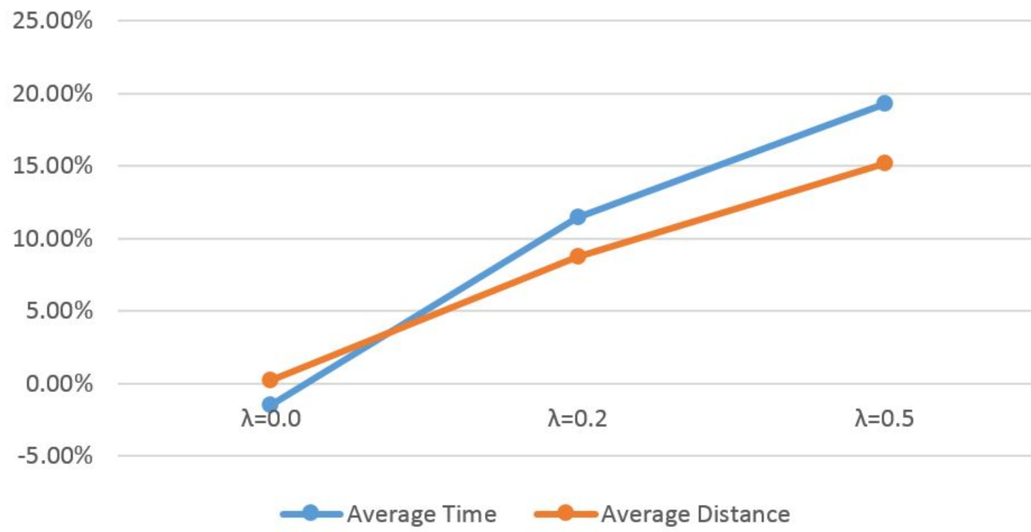
This chapter of the dissertation addressed the analysis of interior layouts configurations, more specifically, warehouse storage assignment. It is vital to consider the performance of an urban environment during the modeling process. Therefore, this chapter presented a method to evaluate the performance of a warehouse according to its items allocation. The first part of the proposed algorithm was to evaluate some features which are closely related to the warehouse efficiency. This evaluation resulted in metrics which could be further used to optimize the storage assignment and reduce congestion levels in a warehouse.

Table 4.3: Analyzing the effect of varying the contribution of the proposed features. The table shows the values for the standard methods (random and class-based) and the proposed congestion-aware method using a variety of λ values.

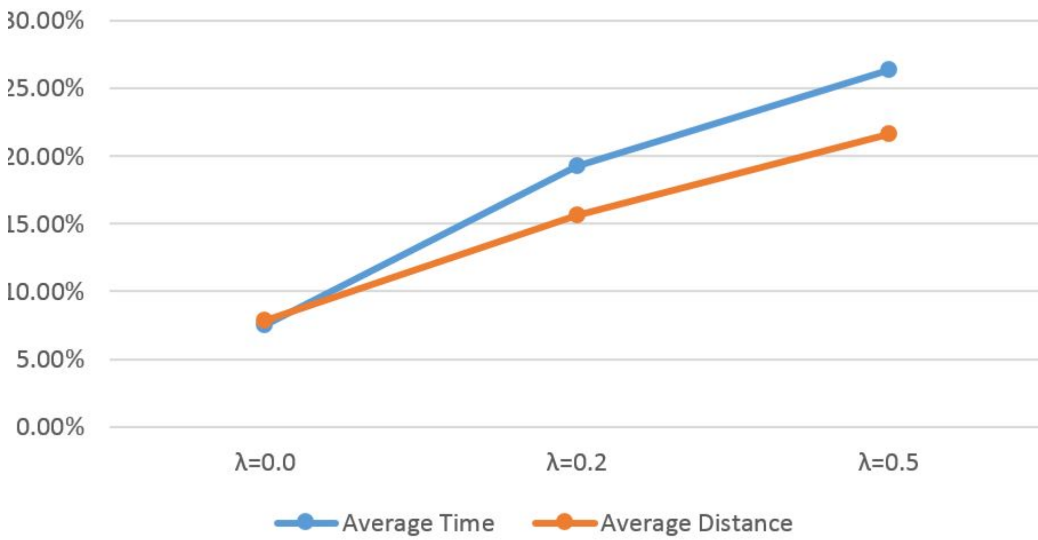
	Random	**Class-based**	$\boldsymbol{\lambda = 0}$	$\boldsymbol{\lambda = 0.2}$	$\boldsymbol{\lambda = 0.5}$
Average Time	11.15	12.23	11.317	9.87	9
Average Distance	452.78	490	451.67	413.3	384

In order to find the optimal storage assignment for an initial randomly assigned layout, the algorithm optimized an energy function that reflects two main features; (i) fast processing of the items and (ii) minimum congestion. Fast processing was achieved by placing the items according to their demand rate i.e., the most important items were placed close to the I/O point while minimum congestion was approximated by analyzing the redundant routes and their effect on traffic. While current storage assignment methods do not consider congestion in their policies, the proposed framework was evaluated on many warehouse layouts to generate a storage assignment that minimizes congestion. The presented analysis provided around 20% time and distance improvement over other existing methods.

In the case of a full warehouse that does not have any empty storage locations, the proposed method degenerates to a class-based storage assignment. This is because the reliability score would be constant in every optimization step resulting in an optimization that depends only on the processing speed (i.e., distance to I/O point). In the future, it would be interesting to investigate coupling reliability estimate with the demand value to overcome this issue. Another avenue of research would be to consider picking multiple items in a single trip which has a significant impact on the warehouse performance. The proposed metrics could be used to evaluate different warehouse layout structures as well as varying many warehouse parameters such as the number of cross aisles or the placement of the I/O point. These variations com-



(a) Congestion-aware vs. Random



(b) Congestion-aware vs.Class-based

Figure 4.11: Increasing λ improves the performance with regards to time and distance. Also, the proposed method (using $\lambda = 0.0$) has around $7\% - 10\%$ improvement over the class-based method. Table 4.3 shows the exact readings for these plots.

bined with the evaluation will lead to novel synthesis possibilities of warehouse layout structures. Moreover, the proposed algorithm could be generalized to be used on a variety of other graph data structures such as floor plans or street networks which leads to the discussion in the following chapter.

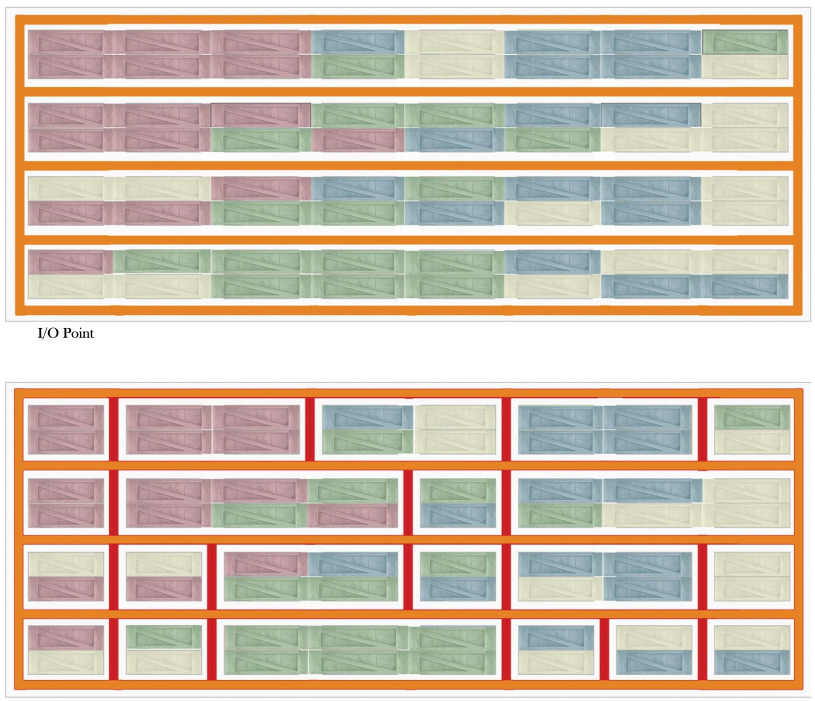


Figure 4.12: The initial (top) layout has a congestion score equals to 0.7092. It is possible to synthesize a new warehouse layout by randomly adding cross aisle edges to reduce the congestion score to 0.3139 (bottom layout). Original aisles are shown in orange while the added ones are shown in red.

Chapter 5

Street Network Analysis

Chapter 3 and 4 discussed the analysis of the exterior and interior configuration of an urban entity i.e., a building. However, people usually move around and navigate between locations which introduce the need to analyze the transportation networks (e.g., street network, train network) that form cities' lifeline, and largely define how the space works and functions. The analysis of transportation networks enables the evaluation of the overall environment performance on a larger scale to understand the interaction between different entities (see Figure 1.5).

The nature of street networks usually defines the style of a city and affects the behavior of the navigation. Many cities have unique characteristics, see Figure 5.1: some provide access to destinations within short walking distance while others are easier to travel by car. The street network patterns mainly influence the transportation styles, for example, grid networks encourage well-balanced traffic flow; while radial layouts provide easy access to the city centers but lead to irregularly shaped building blocks. The variety of navigation behaviors profoundly influences the city and defines its unique characteristics.

This chapter of the dissertation addresses the analysis of the global city by considering its street network pattern at a high level of the urban scale (see Figure 1.5). It asks if cities can be characterized and identified *directly* based on their street network patterns. The primary focus is on capturing the functionality of cities, rather than their visual appearance, i.e., how they *work* rather than how they *look*. The first step towards the answer to this question would be to extract the features that can capture the most important aspects of street networks. The analysis of images and geometry structures of street networks has been addressed through a direct application of existing features extraction and analysis frameworks. However, these forms of analysis fail to capture the essence of street networks for two main reasons. First, street networks are a specialization of geometric graphs, and it is important to adapt existing geometric features. Second, the analysis methods which are based on images and geometric models do not consider critical functional characteristics that are related to questions about paths and transport network efficiency.

Urban planning literature is rich with studies that discuss various factors affecting how a city functions. However, the analysis of the relative importance and effectiveness of the various features have not been qualitatively studied. This part of the dissertation investigates the possible descriptive features of street networks and provides an analysis framework for evaluating their effectiveness. The proposed technique performs topological and geometric analysis of street networks using a set of proposed descriptors. Then, the algorithm uses city-specific classifiers, in a supervised setting, to identify the relative importance of the proposed descriptors. Determining the critical descriptors provides interesting insights into city characteristics and allows to form an understanding of the navigation behavior. While in practice, various secondary factors influence network flow patterns (e.g., street width, peak traffic volume, speed

Figure 5.1: Cities vary according to their different characteristics. This chapter of the dissertation aims to understand the predominant features which affect the functionality of a city.

limits), this initial attempt focuses *only* on the topology and geometric layout of the street networks.

In order to analyze road networks, both inter-city and intra-city classification were used by training SVMs and discriminative analysis classifiers based on the set of proposed network descriptors. For inter-city classification, the density of 4-way crossings and connectivity index play dominant roles; while, for London intra-city classification, intersection, and street densities are the most discriminative. Based on the corresponding precision-recall curves, the descriptors perform significantly better than the state-of-the-art image descriptors working directly on the city (street) maps. Further analysis is conducted to investigate the effect of the scale as well as the relative importance of the different proposed features.

5.1 Overview

The primary goal of the work discussed in this chapter is to characterize and identify cities based on their street network patterns. Starting from multiple street networks given as OpenStreetMap (OSM) [7] data for various cities, the analysis proposes a set of descriptors, and later evaluates the importance of these proposed features in understanding cities' functionalities (see Figure 5.2).

Prior to the analysis, the given 'OSM' input map is preprocessed to parse the input GIS information, extract the underlying street network and represent it as a graph. The first step is to perform both topological and geometric analysis in order to compute the descriptors, and represent each map tile as a high dimensional feature vector. After that, the analysis uses supervised classification methods (SVMs and Discriminative Analysis) to determine the descriptive features and how to identify the uniqueness of cities. This classification provides insights into ranking the features based on their importance which can be used further to improve retrieval and

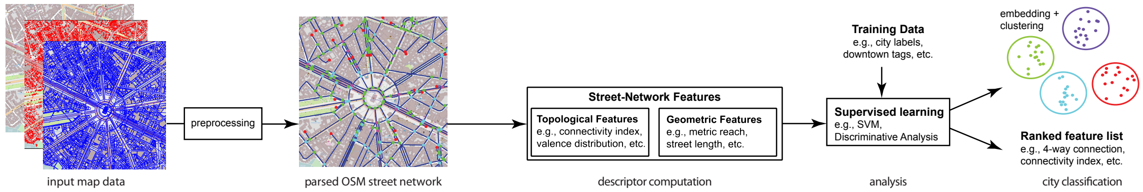


Figure 5.2: An overview of the proposed algorithm. Starting from OpenStreetMap (OSM) data for various cities, topological and geometric analysis is performed using a set of proposed street-level descriptors to characterize different cities. The algorithm learns the relative importance of the proposed street-level descriptors and uses them to retrieve, group, or measure the uniqueness of cities.

classification results.

5.2 Data Collection

In order to consider different styles of cities, I collected maps from ten cities which are (Beijing, Camp Durant, Cardiff, London, Los Angeles, Moscow, New York, Paris, Toronto, and Vienna) from OpenStreetMap [7]. These maps represent different urban patterns such as grid-like, radial, hierarchical, and curved patterns (see Figure 5.1). For each city, I collected 100 samples at three different scales (0.25 km^2, 1 km^2, and 4 km^2). The maps are represented using the standard XML format in the '.osm' files. Then, I parsed these maps to extract street information such as intersection nodes, street segments, street lengths, etc. using the open source library *readosm* [117]. The only focus is on using the primary and secondary roads while ignoring the other types such as stairs, footways, and bike paths.

For each layout, I obtain a set of nodes (i.e., intersections) $\{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$ described by their 2D location, together with the individual street connections $\{r_1, r_2, ..., r_M\}$. Each road $r_i (1 \leq i \leq M)$ is encoded by a sequence of nodes $\{\mathbf{v_{i_1}}, \mathbf{v_{i_2}}, ..., \mathbf{v_{i_{K^i}}}\}$ along the street, where $1 \leq i_j \leq N, 1 \leq j \leq K^i$. Effectively, r_i starts from $\mathbf{v_{i_1}}$ and ends at $\mathbf{v_{i_{K^i}}}$, and $\overline{\mathbf{v_{i_j}}\mathbf{v_{i_{j+1}}}}$ indicates a street segment.

Based on the node set and the connectivity of individual streets, I construct a graph $G = \{V, E\}$ for the entire street layout. The vertex set is defined as $V := \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_N}\}$ and the edge set as $E := \{e_{ij}\}$ where $e_{ij} = \overline{\mathbf{v_i}\mathbf{v_j}}$ denotes a street segment.

5.3 Street Network Analysis

This section discusses the extraction of city descriptive features that capture some of the most important aspects of street networks. First, the algorithm performs topological analysis to reflect the connectivity of the street network and then performs geometric analysis to evaluate the layout effectiveness.

5.3.1 Topological Analysis

The accessibility of a street network is highly affected by its connectivity. The roads and how they are connected affects people's style and ease of navigation within a city. This section describes a set of topological features which capture the connectivity style within a street network. Typically, street networks exhibit a graph structure which is utilized to analyze the following set of features.

Valence (t_{val}) This is the simplest connectivity measure capturing the number of incident roads to an intersection node. The analysis considers the following valence statistics:

- dead ends ($val = 1$) denoted by $t_{v=1}(G)$
- T-junctions ($val = 3$) denoted by $t_{v=3}(G)$
- 4-way crossings ($val = 4$) denoted by $t_{v=4}(G)$
- others ($val > 4$) denoted by $t_{v>4}(G)$

For simplicity, t_{val} represents the vector of four numbers. Figure 5.3 shows vertices with different valences in a sample layout.

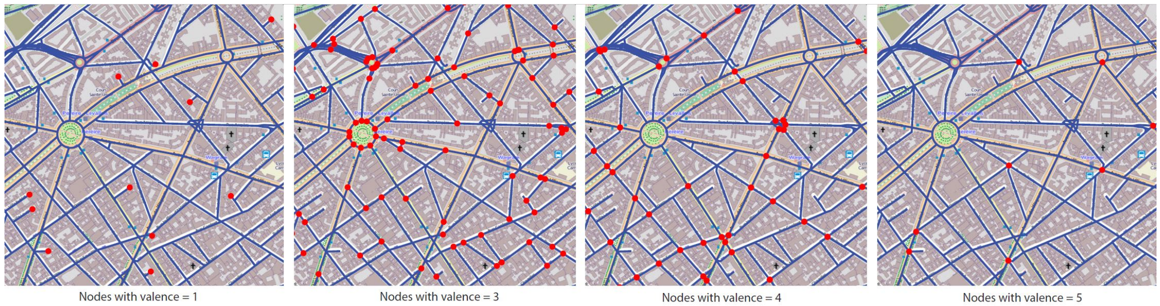


Figure 5.3: Different examples of valence statistics over the nodes of a layout.

Street density (t_{sd}) In the used data, a street is defined as a sequence of graph edges connecting two nodes that should have valence other than 2. This feature is measured as the number of streets within a given map layout area.

Connectivity index (t_{ci}) Connectivity Index is used to quantify how well a street network connects destinations. A higher index means that travelers have increased route choices which allow more direct connections for access between any two locations. The ratio of number of (connecting) links to the number of nodes defines this connectivity index, i.e.,

$$t_{ci}(G) = \sum_i \text{number of edges/number of nodes.} \tag{5.1}$$

According to urban design guidelines, connectivity index is recommended to be above 1.4 to have good accessibility.

Intersection density (t_{id}) It measures the number of intersection nodes within a given layout, i.e., the total number of graph nodes with valence higher than 2.

4-way crossing proportion (t_{cp}) It is the proportion of 4-way crossings with respect to all other intersections. For a Manhattan network, this ratio tends to 1.

Number of blocks (t_{nb}) A block is defined as a polygonal region bounded by streets. This measures how many blocks are in a patch layout (of a fixed area). Essentially, it is the dual of the street network graph.

5.3.2 Geometric Analysis

This section describes some geometric features that capture the behavior of a street network layout. This analysis is necessary to evaluate the ease of traveling between different locations in a city.

Total street length (g_{sl}) The total length of the streets in a network layout directly influences people's navigation around the area. People's population affects this factor, for example, crowded areas require more streets to support more transportation while rural regions require sparse roads.

Average street length (g_{al}) It is is the average street/link length between any two graph nodes whose valences are not 2.

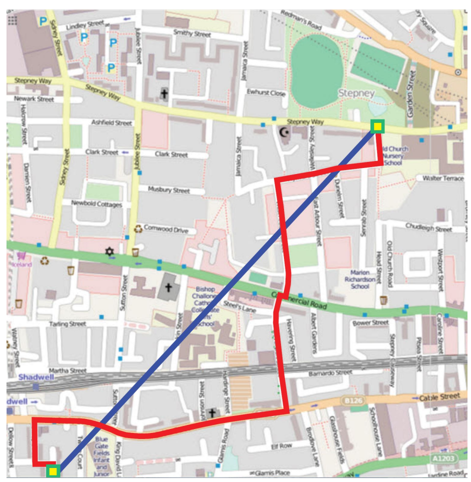


Figure 5.4: A comparison between the Euclidean shortest path (in blue) against the shortest path along the road network connecting the source and destination nodes (in red).

Transportation convenience (g_{tc}) Transportation convenience measures how easy it is to travel from source $\vec{s}$ to destination $\vec{d}$. Typically, the most convenient way of traveling between two locations is to go along a straight line connecting $\vec{s} \rightarrow \vec{d}$. However, this is not always possible if this direct path does not exist. Therefore, the proposed algorithm computes the shortest path using Dijkstra's algorithm as to travel along the network's (graph) nodes and links. Suppose the Euclidean distance between

(a) Valence (b) Redundancy (c) Metric reach (d) Travel distance histogram

Figure 5.5: Topological and geometric features for the same layout: (a) Valence t_{val}, red ($val = 1$), purple ($val = 3$), green ($val = 4$), and yellow ($val > 4$); (b) Redundancy g_r for two pairs of nodes; (c) Metric reach g_{mr} with $x = 0.5$ km; and (d) Travel distance histogram g_{th} for the 25 sampled nodes.

$\vec{s} \rightarrow \vec{d}$ is $d_E(\vec{s}, \vec{d})$ and the shortest graph distance is $d_D(\vec{s}, \vec{d})$, the transportation convenience between $\vec{s}$ and $\vec{d}$ is defined as:

$$g_{tc}(\vec{s}, \vec{d}) := d_E(\vec{s}, \vec{d})/d_D(\vec{s}, \vec{d}). \tag{5.2}$$

In order to evaluate the global transportation convenience $g_{tc}(G)$ for the entire layout, the algorithm samples random source and destination locations (1000 in these experiments) and computes their average. Each pair of nodes of origin and destination should have a minimal separating distance ($d_E(\vec{s}, \vec{d}) > 0.25$ km) to avoid local estimates.

Redundancy (g_r) The ease of rerouting is vital in any street network to occur without serious (time) penalty. *Redundancy* is useful to capture this factor which is defined as the number of different possible routes between two locations $\vec{s} \leftrightarrow \vec{d}$. In order to find the possible redundant routes, the algorithm first performs a breadth-first-search from the source node $\vec{s}$ to the destination node $\vec{d}$, and then backtracks all the paths from $\vec{d}$ to $\vec{s}$. It filters out the inefficient routes identified if their length is more than twice of the shortest graph distance. To estimate the redundancy of the entire layout, it considers the average of the top 100 pairs of nodes with the largest

Euclidean distance. Figure 5.5-b shows redundant paths between pairs of nodes.

Metric reach (g_{mr}) In urban planning, this measure is commonly discussed as the total length of streets if one travels a total of x km (0.5 km in this framework implementation) along all possible directions. Essentially, the total length of a street that one can access from any point as long as the total length is x km (without loops). See Figure 5.5-c.

Travel distance histogram (g_{th}) This descriptor captures the distribution of travel distances over the entire layout. The analysis algorithm uniformly samples 5×5 points and projects them to the nearest graph nodes. Then, it computes the shortest graph distance for each pair of the nodes. It builds the travel distance histogram based on the shortest distances over all the pairs. Finally, it quantizes the histogram into ten bins each of which has the same distance range across all the layouts. See Figure 5.5-d.

5.4 Features Representation

Each network layout L is represented as a feature vector $\vec{f}$ that reflects the set of features described in the previous section as follows:

$$\vec{f}(L) := [t_{val}, t_{sd}, t_{ci}, t_{id}, t_{cp}, t_{nb}, g_{sl}, g_{al}, g_{tc}, g_r, g_{mr}, g_{th}]. \tag{5.3}$$

The dimension of the feature vector is 24, given that t_{val} has four components and g_{th} has ten bins. In order to have the same range for each descriptor, the features are normalized based on the extent found in the corresponding training sets. Multidimensional scaling (MDS) is used to embed all the features for the ten cities in a 2D embedding as shown in Figure 5.6

Note that at the smaller scale (left figure), New York clearly stands out, while London and Paris are easy to confuse. The embedding with more cities (right figure) shows a hierarchical nature, where some cities are more different from others, but New

York remains quite distinctive. Note that the following analysis was performed in the $\vec{f}(L)$ space, rather than the 2D-embedded space, which is only used for visualization.

5.5 Layout Classification

One of the primary goals of this research is to investigate whether it is possible to characterize and identify cities using the proposed descriptors which capture the network patterns. In order to answer this question, the proposed analysis algorithm uses supervised learning using Support Vector Machine (SVM) [118] and Discriminative Analysis (DA) [119] linking the feature vectors and labels. Each classification method uses two kernel types. For SVM, the linear and radial basis functions (RBF) are used. The framework extracts the parameter values for RBF-SVM using 10-fold cross-validation which uses different partitions of the training data to determine the best parameter values. For DA, the framework uses Linear and Quadratic versions. The experiments used a variety of training set sizes i.e., 5%, 10%, 20%, and 50% of the input data. The classifiers have similar performance beyond 5%. Hence, as a default, the following experiments use 10% training data, unless mentioned. The

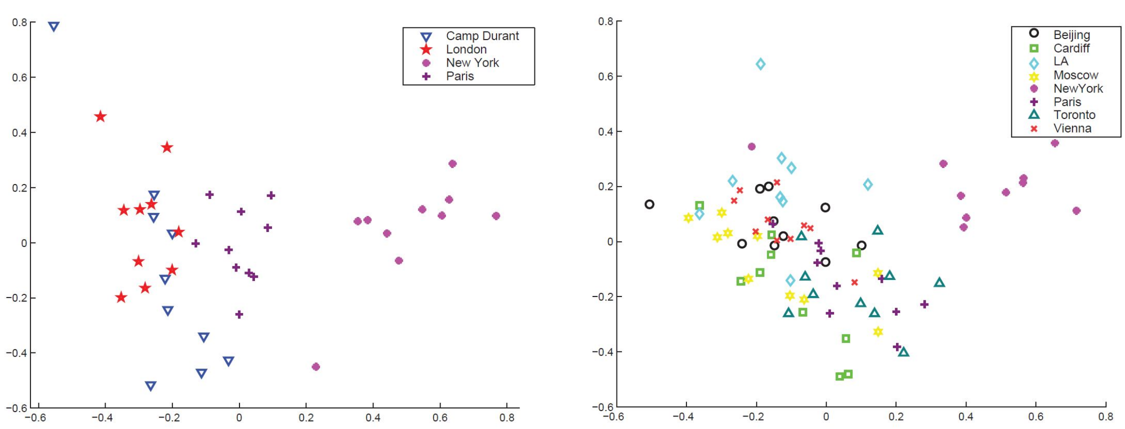


Figure 5.6: Visualizing cities based on a 2D-embedding of the corresponding features. A random sampling of cities is shown (left) although the embedding is generated using all the images from the respective cities.

next experiments perform the classifications with three different purposes as discussed next.

5.5.1 Intra-City Classification

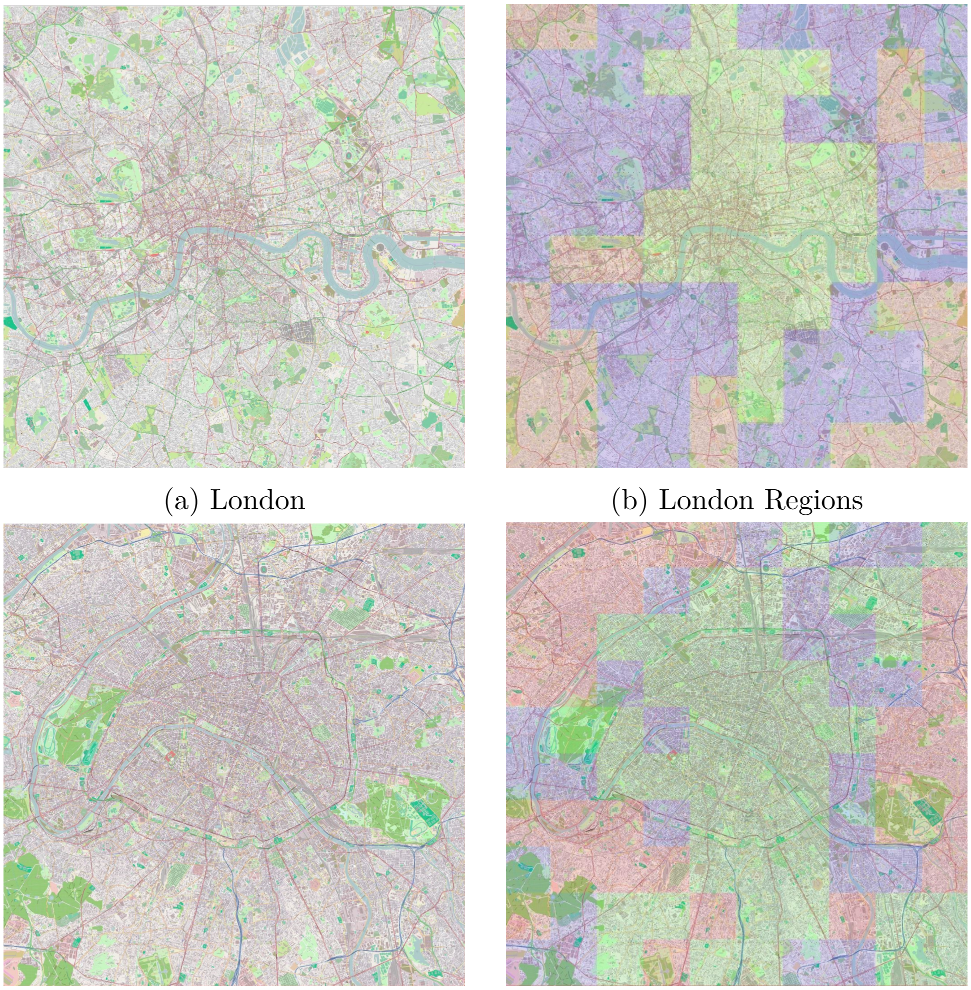
This type of classification distinguishes a city from other cities using the captured features. It presents multi-class classification with ten classes and class-specific classification with two classes (one for a city and one for all the remaining cities). Multi-class classifiers performed better compared to class-specific classifiers (see Figure 5.11), but the general accuracy was higher for binary classifiers (see Table 5.1).

classification type	SVML	SVMR	LDA	QDA
multi-class	35.8	41.7	47.3	53.4
Beijing and others	90.2	90.6	88.7	76.0
Camp Durant and others	90.2	91.1	88.8	89.9
Cardiff and others	91.1	91.3	83.8	84.1
LA and others	88.8	89.0	82.0	83.0
London and others	90.8	91.2	86.6	88.4
Moscow and others	89.3	89.1	80.3	83.7
NY and others	93.6	94.7	90.7	90.5
Paris and others	92.9	93.2	87.2	89.4
Toronto and others	89.0	89.1	67.9	66.2
Vienna and others	90.0	90.0	74.6	74.0

Table 5.1: Cross validation precision (%) using different classification methods with 10% data as a training size. SVML and SVMR refer to SVM linear and SVM RBF, respectively. LDA and QDA refer to linear discriminant analysis and quadratic discriminant analysis, respectively. Note that for binary classifiers, higher precision comes at the cost of lower recall (refer to Figure 5.11).

5.5.2 Inter-City Classification

This section asks how to distinguish different areas of London, working at three levels: downtown (zone 1-4), mid-zone (zone 5-8), and outskirts (zone 8+). I manually labeled the input map tiles in the training set into 3 zones: *downtown*, *mid-zone*, and *outskirts*. The classification of these three levels achieved high accuracy (around 70%) in the tests.



(a) London (b) London Regions

(c) Paris (d) Paris Regions

Figure 5.7: Using K-Means clustering [12] to identify city levels for London and Paris (left) into three regions (right) which are downtown (green), mid-zone (blue), and outskirts (red).

5.5.3 City Regions Clustering

Similar to inter-city classification, it is also interesting to cluster the regions within a single city into its major parts. Given the city patches on the scale of 4 km^2, we used *K-Means clustering* [12] to identify the regions which reflect the street network patterns of the three levels: downtown (zone 1-4), mid-zone (zone 5-8), and outskirts (zone 8+). Figure 5.7 shows the clustering results for London and Paris cities.

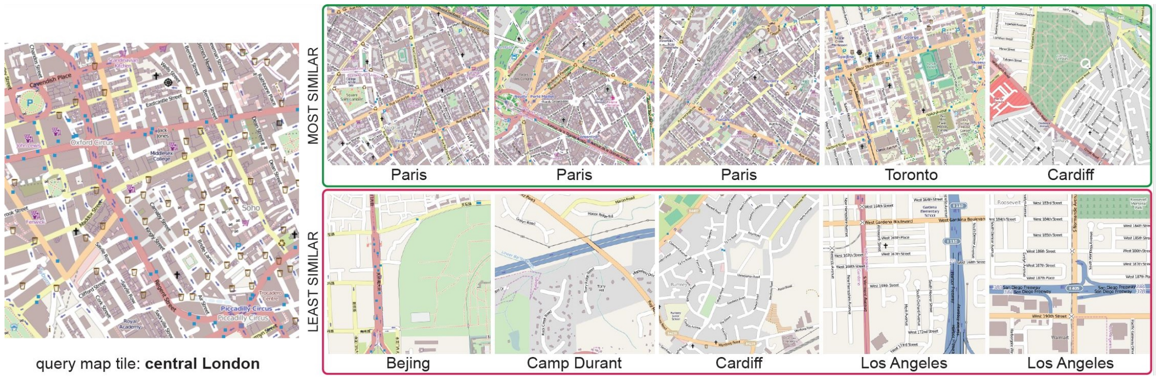


Figure 5.8: Finding 'sister' cities retrieves the similar and dissimilar layouts from other cities for a query patch of central London.

5.5.4 Downtown Classification

It is also interesting to look for retrieving similar street network using the proposed features. Essentially, this is the nearest neighbor query in the corresponding features space. For example, I take a sample patch (1 km^2) from London downtown and query for its *closest* and *farthest* layouts across all *other* city patches. The retrieved nearest patches exhibit similar style to London downtown while the furthest patches have different patterns (see Figure 5.8). Note that the match with Cardiff is an error as the network inside the park also is marked as a street in the input data.

5.6 Features Analysis

After discussing the proposed features, it would be interesting to look at the importance of these features and discuss some effects of some factors on using these features as discussed next.

5.6.1 Features Effectiveness - Intra-City Classification

The various proposed features are not equally effective in the classification process; hence, it would be interesting to evaluate their effectiveness and determine the most prominent ones. The first step in this evaluation is to learn the weights for the various descriptors using the SVM-linear classifier. The top five features which have a great impact on intra-city classification (refer to section 5.5.1) are:

- 4-way crossing proportion
- connectivity index
- # valence 4
- # valence 3
- number of blocks

5.6.2 Features Effectiveness - Inter-City Classification

Another interesting evaluation avenue is to consider the effectiveness of the proposed features for inter-city classification (see section 5.5.2) as to distinguish different areas of London at three levels: downtown, mid-zone, and outskirts (see Figure 5.9). The top five discriminative features are:

- nodes density
- street density
- # valence 3
- # valence 4
- total street length

Using these features for inter-city classification has shown to improve the classification and retrieval results.

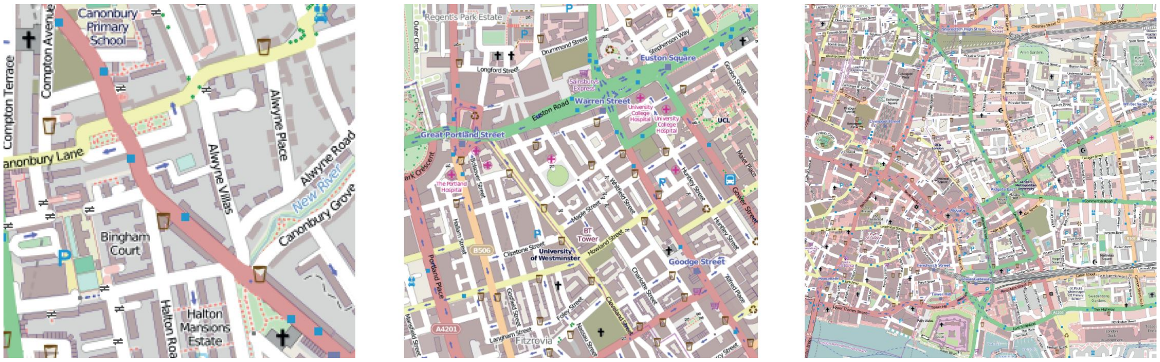


Figure 5.9: Sample map layouts from London at different scales; (left - 0.25 km^2, middle - 1 km^2, and right - 4 km^2).

5.6.3 Best Layout Scale

The scale of the layout has an effect on the features as varying scales give different information regarding quality and size. In this type of data (street networks), a small scale would cover a small area and provide roads connecting the neighborhood while a larger scale could contain the connections between different cities. In the scope of this part, the street network properties could be evaluated better using a particular scale. To investigate the good scale, the proposed experiments evaluated the features at three different scales, namely 0.25 km^2, 1 km^2, and 4 km^{2}blocks, while normalizing by the tile areas. Figure 5.9 shows sample map patches at the three different scales. For most of the cities, the middle-scale, i.e., 1 km^2 was found to be the most effective. When considering the lowest scale, it is expected that the performance drops as smaller regions do not represent the distinct pattern of the patch which was true as the performance degraded rapidly (by 20-30%). Higher scale performance also degraded (approximately 10%) as a larger patch could exhibit more patterns (e.g., downtown and mid-zone areas).

5.6.4 Comparison with Image Features

In order to evaluate the efficiency of the proposed features, they were compared against image-level descriptors. I took the parsed street networks and rasterized them

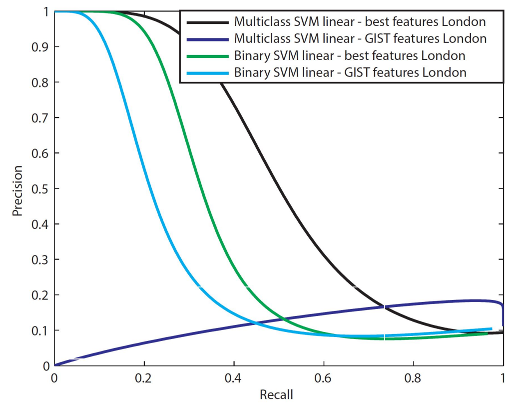


Figure 5.10: Precision-recall curves for classification with 10% training data using GIST descriptors and the most prominent features. Note that the performance is better with the selected features.

to form image patches. I used bag-of-word features with GIST descriptor [120, 121]. Then, supervised classification was performed using GIST descriptors and the proposed descriptors $\vec{f}(L)$ and found the proposed descriptors to be consistently better. For example, using SVM-linear, binary classification accuracy suffered by 20% or more for each of the ten cities. The proposed descriptors perform better as they capture the topological and geometric essence of street networks while GIST descriptors fail to capture network level details.

In order to show the performance difference, it is useful to plot the precision-recall plot (Figure 5.10) for comparing the classification of London layouts using GIST descriptors and the most discriminative features (4-way crossing proportion, connectivity index, # valence 4, # valence 3, number of blocks) identified in section 5.6.1. The experiments used multi-class and binary SVM-linear classification with both sets of features. In both classifications, the classification using the selected proposed features outperformed the classification results using GIST features.

5.6.5 Classification Using Distinctive Features

Section 5.6 discussed the effectiveness of the proposed features and identified the most discriminative features. This section shows the effect of using only these distinctive features for the classification. The analysis performs the classification while ignoring the less important features (metric reach, redundancy, node density, street density, total street length, and histograms) and plots the precision-recall curves. As shown in Figure 5.11, the performance of all the classifiers improves when the classification uses only the most prominent features (only London example was shown here).

The confusion matrix reveals interesting patterns about the features. The top confusion happens between London↔Los Angeles; the second on the list is London↔Paris, and the third is Los Angeles↔Beijing. The least confusing set was Cardiff↔London. However, the selected descriptors could reliably distinguish between London↔New York.

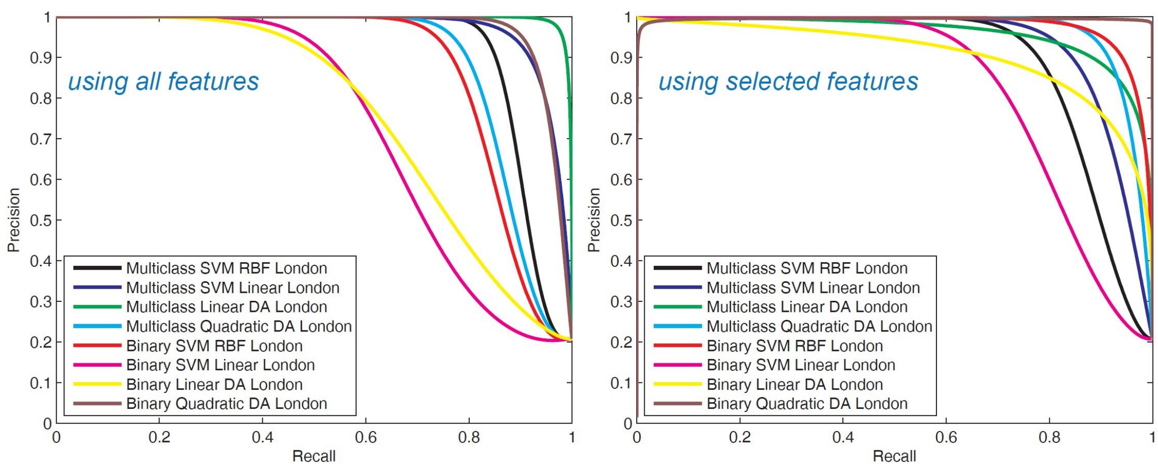


Figure 5.11: Precision-recall curves for various classifiers with 10% training data. Note that the performance is better with selected features.

5.7 Conclusion

This chapter of the dissertation presented an analysis algorithm for the high-level urban data; more specifically street networks of cities. It proposed a set of various topological and geometric features to characterize street networks across different cities. The algorithm learns to classify the cities using the data obtained from ten cities, each with 100 different map tiles collected at three different scale levels. The classification results allow determining the relative importance of the proposed features which enable further enhancements of the classification tasks. Moreover, the proposed features are useful to support other retrieval tasks, for example finding cities with similar patterns and characteristics as that of 'central London'. This approach is the first attempt to capture the functional behavior of street networks, in parallel to the familiar image or geometric descriptors.

Besides, the proposed features can be used in other directions of applications. For instance, it would be possible to measure the 'functional validity' of street networks which has immediate relevance to city planning and procedural modeling (of street networks). Moreover, the analysis of the geometry and connectivity aspects of a street network gives novel insights related to the 'essence' of the network. Therefore, it would be possible to use the proposed descriptors to evaluate different synthesized street networks based on their functionality. In the future and beyond synthesis applications, it would be interesting to incorporate other attributes and evaluate the effect of other factors like population density, street width, etc. on the behavior of cities. The relation to traffic simulation also poses interesting research questions from an inverse modeling perspective.

Chapter 6

Concluding Remarks

This chapter concludes the dissertation with a summary of the contributions and some thoughts about extensions of this work and future research.

One big goal in computer graphics is being able to create realistic procedural models for living spaces, neighborhoods or cities. This dissertation targeted several types of urban data at a variety of scales and proposed a set of methods to analyze these data. It first addressed the *exterior* layout of an urban space through the analysis of building facades. It proposed a method to decouple window-level deformations (i.e., windows opening parameter) and detect the structural grid. Then, it focused on the evaluation of the *interior* layout of urban spaces using a congestion-aware interior configuration method to optimize the placement of items in their storage locations within a warehouse. The last part considered the data on a larger scale by characterizing *street network* patterns with the goal of capturing city behavior based on a set of topological and geometric descriptive features which were used in a supervised setting. Each chapter presented the results that demonstrate how to use the proposed algorithms to enable a variety of novel applications.

6.1 Contributions

In the context of computer graphics and procedural modeling, the main contributions of this dissertation are:

- Introducing the factored facade representation that accurately encodes window-level low degree of freedom movements.
- An optimization framework for items allocation in a warehouse layout while minimizing its traffic congestion.
- A framework for characterizing and classifying city patches at the street network level based on a set of features representing their street network patterns.

6.2 Summary

Analysis on a Variety of Scales

The vast availability of urban data encourages their analysis with the aim of converting the data into meaningful information that would help people improve their living spaces. These spaces could be represented at different scales to reflect various magnitudes of individuals' livings. They could be as large as a whole city or as small as an interior room layout. The underlying theme of this dissertation had been to analyze urban data on a variety of scales. It first started by examining the outer facades of buildings to extract the regularity pattern among the repeated window elements, while also identifying the corresponding window deformations. Then, it moved to improve the efficiency of interior layouts by focusing on warehouse storage assignment to promote items allocation through the analysis of the interior layout and its effects on traffic congestion. Finally, it moved to an outer scale which covered the functional performance of road networks and provided novel ways of looking at city patches.

Use of Suitable Features

An important part of any analysis problem is deciding the features to be used to represent the interesting characteristics of the data being studied. Each part of this dissertation proposed a set of features to facilitate analyzing the data for different purposes. For instance, the analysis of windows requires the use of features within the facade images such as color intensities or edge properties to find the windows and their deformation parameters. Coupling 2D and 3D features was also used to handle the nature of rotating shutter windows (as discussed in section 3.4.2). Moreover, evaluating the efficiency of storage assignment policy within a warehouse was possible by assessing congestion probabilities and their effect on time and distance needed to perform order picking processes. With a view to analyzing the behavior of road networks in cities, a set of topological and geometric features was proposed which enabled further studies to understand the functionalities of cities.

Applications

A set of applications were discussed throughout the dissertation allowing novel possibilities toward realistic procedural modeling. *Factored Facade* representation was proposed to encode a facade image and its elements which enables a range of unique interactions to bring static (single) facade images to life. Besides, analyzing congestion within a warehouse allows new applications to enhance the effectiveness of storage assignment and improve the interior layout efficiency. Moreover, the effectiveness of the proposed features reflecting street network patterns was evaluated through their use for classification in a supervised setting. This evaluation enables a unique understanding of the functionality of street network layouts that is important for city planners and decision makers.

6.3 Future Work

Urban data are continuously growing with the ever going advancement of technologies and acquisition possibilities. This growth poses new challenges for researchers to utilize and process these data. Therefore, it is expected to see more future work in the domain of urban data analysis to facilitate further realistic reconstruction and modeling applications.

Chapter 3 addressed the problem of facade structure analysis which is a widely studied area tackled through a variety of approaches. The chapter presented an initiative to analyze the per-element deformation and approximate it using a parametric value. Such approximation would be very useful to explore image parts and could be employed further to detect irregularities. The proposed framework enables the possibility to perform interactive edit operations to synthesize new facade elements. Editing the facade has been recently addressed in many directions such as completing missing elements from an observed structure [122], interactive facade editing while preserving dominant structural relations [29] or layout regularization based on the detected constraints [123].

Considering the environment setting and evaluating its efficiency is an integral part of the process of generating realistic models. The ability to predict the behavior of environments based on their usage is very critical. In this regard, chapter 4 addressed the interior layout setting of a warehouse concerning storage assignment of items. While the analysis studied resource allotment for warehouses, the method could also be applied to a broader range of graphs such as network design in city layouts or floor plans in buildings. Therefore, it fits very well with the long-term goal of realistic procedural modeling. A recent example was presented by Peng et al. [37] who proposed a method to create a layout network by defining the functional behavior specifications and used it in the context of urban street layouts, floor planning, and game level design.

The last part of the dissertation proposed a set of features to measure the functional validity of street networks. These measures could be utilized to support many synthesis operations such as generating new network layouts, copying a network pattern or combining different patches. Nishida et al. [124] made a recent effort in these directions who presented an interactive tool to design roads with complicated, realistic details and styles. Roads are generated by growing a geometric graph either as example-based growth which preserves some interesting structures using patches extracted from the source example or as procedural-based growth using the statistical information of the source road networks. Such syntheses and exploration of urban road networks enable new techniques to support urban planning, traffic simulation, 3D content creation and procedural modeling operations. Evaluating the validity of the synthesized models could be achieved by using the proposed geometric and topological features.

Furthermore, coupling the road network with more interesting attributes such as population, age, income or other demographics would introduce new perspectives to view and analyze living spaces. This investigation, in turn, adds new questions to answer, hence, enhances the generated models and harvests the benefits of urban data. Such an approach was followed recently by Ferreira et al. [125] in *Urbane* framework which provides new ways to analyze data and visualize properties of neighborhoods and buildings as well as simulate developments. Such a system is very useful for architects, developers, and planners with the goal of improving decision-making in urban development.

REFERENCES

[1] KAUST. King abdullah university of science and technology. [Online]. Available: www.kaust.edu.sa

[2] Esri. Cityengine. [Online]. Available: http://www.esri.com/software/cityengine

[3] FalconViz. Falconviz. [Online]. Available: www.falconviz.com

[4] S. AlHalawani, Y.-L. Yang, P. Wonka, and N. J. Mitra, "What Makes London Work Like London?" *Computer Graphics Forum*, vol. 33, no. 5, pp. 157–165, Aug. 2014.

[5] F. Bao, M. Schwarz, and P. Wonka, "Procedural facade variations from a single layout," *ACM Transactions on Graphics*, vol. 32, no. 1, pp. 1–13, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2421636.2421644

[6] H. Liu, Y.-L. Yang, S. AlHalawani, and N. J. Mitra, "Constraint-aware interior layout exploration for precast concrete-based buildings," *Visual Computer (CGI Special Issue)*, 2013.

[7] Openstreetmap. [Online]. Available: https://www.openstreetmap.org

[8] M. Il? and T. Auzinger, "Layer-based procedural design of façades," vol. 34, no. 2, 2015.

[9] M. T. Tsige, "Improving order-picking efficiency via storage assignment strategies," 2013.

[10] K. R. Gue and R. D. Meller, "Aisle configurations for unit-load warehouses," *IIE Transactions*, vol. 41, no. 3, pp. 171–182, 2009.

[11] C. Wu, J.-M. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," in *CVPR*, 2011.

[12] J. B. MacQueen, "Kmeans Some Methods for classification and Analysis of Multivariate Observations," *5th Berkeley Symposium on Mathematical Statistics and Probability 1967*, vol. 1, no. 233, pp. 281–297, 1967. [Online]. Available: http://projecteuclid.org/euclid.bsmsp/1200512992

[13] O. E. Dictionary. [Online]. Available: http://www.etymonline.com/index.php?term=urban

[14] *World Urbanization Prospects: The 2014 Revision.* United Nations, Department of Economic and Social Affairs, Population Division, 2014, (ST/ESA/SER.A/366).

[15] W. Harry, "Are compact cities a desirable planning goal ?" 1997.

[16] M. Government. [Online]. Available: http://www.makkah.gov.sa/news/show/56

[17] M. B. Hui Lin, *Virtual Geographic Environments*, M. B. Hui Lin, Ed. ESRI Press, 2009.

[18] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A survey of urban reconstruction," in *EUROGRAPHICS 2012 State of the Art Reports*, ser. EG STARs. Eurographics Association, May 2012, pp. 1–28. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2012/musialski-2012-sur/

[19] 3dmekanlar. (2008) Dolmabahce palace. [Online]. Available: http://www.3dmekanlar.com/en/dolmabahce-palace.html

[20] Opel. Opel adam with intellilink best connected small car. [Online]. Available: http://www.opel.com/news/index/2013/07/ADAM_with_IntelliLink_best_connected_small_car.html

[21] W. D. P. Pixar. (2009) Up. [Online]. Available: http://www.imdb.com/title/tt1049413/

[22] U. of Cape Town. Zamani project. [Online]. Available: http://www.digitalmeetsculture.net/article/africa-and-beyond-slave-trade-in-3d-by-zamani-project/

[23] J. Yao, H. Tawfik, and T. Fernando, "A gis based virtual urban simulation environment," *Computational Science Ű ICCS 2006*, vol. 3993, pp. 60–68, 2006. [Online]. Available: http://dx.doi.org/10.1007/11758532_10$\delimiter"026E30F$nhttp://download.springer.com/static/pdf/409/chp%3A10.1007%2F11758532_10.pdf?auth66=1389520892_a87060a19c7d772fa585716572301931&ext=.pdf$\delimiter"026E30F$nhttp://link.springer.com/chapter/10.1007/11758532_10

[24] P. Müller, G. Zeng, P. Wonka, and L. Van Gool, "Image-based procedural modeling of facades," in *ACM SIGGRAPH 2007 papers*, ser. SIGGRAPH '07. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1275808.1276484

[25] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu, "Adaptive partitioning of urban facades," in *Proceedings of the 2011 SIGGRAPH Asia Conference*, ser. SA '11. New York, NY, USA: ACM, 2011, pp. 184:1–184:10. [Online]. Available: http://doi.acm.org/10.1145/2024156.2024218

[26] P. Musialski, M. Wimmer, and P. Wonka, "Interactive coherence-based façade modeling," *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2012)*, vol. 31, no. 2, pp. 661–670, May 2012. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2012/musialski-2012-icb/

[27] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, "Shape grammar parsing via reinforcement learning," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, pp. 2273 –2280.

[28] P. Musialski, C. Luksch, M. Schwärzler, M. Buchetics, S. Maierhofer, and W. Purgathofer, "Interactive multi-view façade image editing," in *Vision, Mod-*

eling, and Visualization (2010), C. R.-S. Reinhard Koch, Andreas Kolb, Ed. The Eurographics Association, 2010.

[29] M. Dang, D. Ceylan, B. Neubert, and M. Pauly, “Safe: Structure-aware facade editing,” *Computer Graphics Forum*, vol. 33, no. 2, pp. 83–93, 2014.

[30] P. Galle, “An algorithm for exhaustive generation of building floor plans,” *Commun. ACM*, vol. 24, no. 12, pp. 813–825, Dec. 1981. [Online]. Available: http://doi.acm.org/10.1145/358800.358804

[31] D. F. Wong and C. L. Liu, “A new algorithm for floorplan design,” in *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, ser. DAC ’86. Piscataway, NJ, USA: IEEE Press, 1986, pp. 101–107. [Online]. Available: http://dl.acm.org/citation.cfm?id=318013.318030

[32] F. Bao, D.-M. Yan, N. J. Mitra, and P. Wonka, “Generating and exploring good building layouts,” *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, Jul. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2461912.2461977

[33] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, “Interactive furniture layout using interior design guidelines,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 87:1–87:10, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964982

[34] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, “Make it home: Automatic optimization of furniture arrangement,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 86:1–86:12, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964981

[35] C. Ma and A. Inria, “Game Level Layout from Design Specification,” vol. 33, no. 2, 2014.

[36] M. Ebrahimi, M. Daneshtalab, F. Farahnakian, J. Plosila, P. Liljeberg, M. Palesi, and H. Tenhunen, “Haraq: Congestion-aware learning model for highly adaptive routing algorithm in on-chip networks,” *Proceedings of the 2012 6th IEEE/ACM International Symposium on Networks-on-Chip, NoCS 2012*, pp. 19–26, 2012.

[37] C.-H. Peng, N. J. Mitra, F. Bao, D.-M. Yan, and P. Wonka, "Computational Network Design from Functional Specifications," no. April 2016, 2015. [Online]. Available: http://arxiv.org/abs/1510.09203

[38] C. Doersch and S. Singh, "What makes paris look like paris ?" 2011.

[39] L. Yang, J. M. Normand, and G. Moreau, "Augmenting off-the-shelf paper maps using intersection detection and geographical information systems," *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, pp. 190–193, 2015.

[40] C. A. Vanegas, D. G. Aliaga, B. Beneš, and P. A. Waddell, "Interactive design of urban spaces using geometrical and behavioral modeling," *ACM Trans. on Graph.*, vol. 28, no. 5, 2009.

[41] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and P. Waddell, "Inverse design of urban procedural models," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 168:1–168:11, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366187

[42] Y.-L. Yang, J. Wang, E. Vouga, and P. Wonka, "Urban pattern: Layout design by hierarchical domain splitting," *ACM Trans. on Graph.*, vol. 32, no. 6, pp. 181:1–181:12, Nov. 2013.

[43] S. AlHalawani, Y.-L. Yang, H. Liu, and N. J. Mitra, "Interactive facades: Analysis and synthesis of semi-regular facades," *Computer Graphics Forum (Eurographics)*, vol. 32, no. 2pt2, pp. 215–224, 2013.

[44] S. AlHalawani and N. Mitra, "Congestion-aware warehouse flow analysis and optimization," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, Eds. Springer International Publishing, 2015, vol. 9475, pp. 702–711. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-27863-6_66

[45] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, "A survey on procedural modelling for virtual worlds," *Computer Graphics Forum*, 2014.

[46] T. Hasti, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer-Verlag, 2009.

[47] C. Bishop, *Pattern recognition and machine learning.* Springer, 2009.

[48] Y. Liu, J. H. Hays, Y.-Q. Xu, and H.-Y. Shum, "Digital papercutting," in *Technical Sketch, SIGGRAPH*, 2005.

[49] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *ECCV (2)*, 2006, pp. 508–521.

[50] N. J. Mitra, L. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 25, no. 3, pp. 560–568, 2006.

[51] ——, "Symmetrization," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 26, no. 3, pp. #63, 1–8, 2007.

[52] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering structural regularity in 3D geometry," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. #43, 1–11, 2008.

[53] N. J. Mitra, A. Bronstein, and M. Bronstein, "Intrinsic regularity detection in 3d geometry," in *ECCV*, 2010, p. to appear.

[54] C. Wu, J.-M. Frahm, and M. Pollefeys, "Detecting large repetitive structures with salient boundaries," in *Proceedings of the 11th European conference on Computer vision: Part II*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 142–155. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888028.1888040

[55] C. Wu, J. Frahm, and M. Pollefeys, "Repetition-based dense single-view reconstruction," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, pp. 3113 –3120.

[56] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen, "Smartboxes for interactive urban reconstruction," *ACM Trans. Graph.*, vol. 29, no. 4,

pp. 93:1–93:10, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778830

[57] D. G. Aliaga, "Coupled segmentation and similarity detection for architectural models," pp. 1–11, 2015.

[58] D. Ceylan, N. J. Mitra, H. Li, T. Weise, and M. Pauly, "Factored facade acquisition using symmetric line arrangements," *Computer Graphics Forum (EUROGRAPHICS)*, 2012.

[59] S. Alsisan and N. J. Mitra, "Variation-factored encoding of facade images," in *EUROGRAPHICS Short Paper*, 2012.

[60] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of building facades using procedural shape priors," in *CVPR*, 2010, pp. 3105–3112.

[61] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 11–20. [Online]. Available: http://doi.acm.org/10.1145/237170.237191

[62] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," in *SIGGRAPH Conference Proceedings*. New York, NY, USA: ACM Press, 2006, pp. 835–846.

[63] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *CVPR*, 2009.

[64] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan, "Image-based façade modeling," in *ACM SIGGRAPH Asia 2008 papers*, ser. SIGGRAPH Asia '08. New York, NY, USA: ACM, 2008, pp. 161:1–161:10. [Online]. Available: http://doi.acm.org/10.1145/1457515.1409114

[65] J. Xiao, T. Fang, P. Zhao, M. Lhuillier, and L. Quan, "Image-based street-side city modeling," in *ACM SIGGRAPH Asia 2009 papers*, ser. SIGGRAPH Asia

'09. New York, NY, USA: ACM, 2009, pp. 114:1–114:12. [Online]. Available: http://doi.acm.org/10.1145/1661412.1618460

[66] M. Steinberger, M. Kenzel, B. Kainz, J. Müller, W. Peter, and D. Schmalstieg, "Parallel generation of architecture on the gpu," *Computer Graphics Forum*, vol. 33, no. 2, pp. 73–82, 2014. [Online]. Available: http://dx.doi.org/10.1111/cgf.12312

[67] P. Musialski, M. Recheis, S. Maierhofer, P. Wonka, and W. Purgathofer, "Tiling of ortho-rectified façade images," in *Proceedings of 26th Spring Conference on Computer Graphics (SCCG 2010)*, R. K. Helwig Hauser, Ed. Comenius University, Bratislava, May 2010, pp. http://www.cg.tuwien.ac.at/research/publications/2010/musialski-2010-tof/

[68] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, vol. to appear, no. to appear, p. to appear, 2008.

[69] X. Ning, X. Zhang, and Y. Wang, "Automatic architecture model generation based on object hierarchy," in *ACM SIGGRAPH ASIA 2010 Sketches*, ser. SA '10. New York, NY, USA: ACM, 2010, pp. 39:1–39:2. [Online]. Available: http://doi.acm.org/10.1145/1899950.1899989

[70] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. J. Mitra, "2d-3d fusion for layer decomposition of urban facades," in *ICCV*, 2011.

[71] L. B. S. Warren Hausman and S. C. Graves, "Optimal storage assignment in automatic warehouse systems," *Management Science*, 1976.

[72] J. Gu, M. Goetschalckx, and L. F. McGinnis, "Research on warehouse operation: A comprehensive review," *European Journal of Operational Research*, vol. 177, no. 1, pp. 1–21, 2007.

[73] S. C. Graves, W. H. Hausman, and L. B. Schwarz, "Storage-retrieval interleaving in automatic warehousing systems," *Management Science*, vol. 23,

no. 9, pp. 935–945, 1977. [Online]. Available: http://dx.doi.org/10.1287/mnsc.23.9.935

[74] L. B. Schwarz, S. C. Graves, and W. H. Hausman, “Scheduling policies for automatic warehousing systems: Simulation results,” *A I I E Transactions*, vol. 10, no. 3, pp. 260–270, 1978.

[75] T. Elezkurtaj and G. Franck, “Algorithmic support of creative architectural design,” *Umbau*, 2002.

[76] R. C. JEREMY J. MICHALEK and P. Y. PAPALAMBROS, “Architectural layout design optimization,” *Engineering optimization*, 2002.

[77] S. Schneider, J.-R. Fischer, and R. König, “Rethinking automated layout design: Developing a creative evolutionary design method for the layout problems in architecture and urban design,” in *Design Computing and Cognition*, J. Gero, Ed. Springer Netherlands, 2011, pp. 367–386. [Online]. Available: http://dx.doi.org/10.1007/978-94-007-0510-4_20

[78] D. Camozzato, L. Dihl, I. Silveira, F. Marson, and S. R. Musse, “Procedural floor plan generation from building sketches,” *The Visual Computer*, pp. 753–763, 2015. [Online]. Available: ”http://dx.doi.org/10.1007/s00371-015-1102-2

[79] R. Gallager, “A minimum delay routing algorithm using distributed computation,” *Communications, IEEE Transactions on*, vol. 25, no. 1, pp. 73–85, Jan 1977.

[80] J. Aslam, S. Lim, and D. Rus, “Congestion-aware traffic routing system using sensor data,” *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1006–1013, 2012.

[81] R. D. Meller and K.-Y. Gau, “The facility layout problem: Recent and emerging trends and perspectives,” *Journal of Manufacturing Systems*, vol. 15, no. 5, pp. 351 – 366, 1996.

[82] R. MELLER, “Optimal order-to-lane assignments in an order accumulation/-sortation system,” *IIE Transactions*, vol. 29, no. 4, pp. 293–301, 1997.

[83] J. A. T. J. A. W. Y. A. B. J. M. A. Tanchoco, *Facilities Planning*, 2002.

[84] R. W. HALL, “Distance approximations for routing manual pickers in a warehouse,” *IIE Transactions*, vol. 25, no. 4, pp. 76–87, 1993.

[85] K. J. Roodbergen and I. F. a. Vis, “A model for warehouse layout,” *IIE Transactions*, vol. 38, no. 10, pp. 799–811, 2006.

[86] K. J. Roodbergen, G. P. Sharp, and I. F. Vis, “Designing the layout structure of manual order picking areas in warehouses,” *IIE Transactions*, vol. 40, no. 11, pp. 1032–1045, Sep. 2008.

[87] C. G. Petersen, “An evaluation of order picking routeing policies,” *International Journal of Operations & Production Management*, vol. 17, no. 11, pp. 1098–1111, 1997. [Online]. Available: http://dx.doi.org/10.1108/01443579710177860

[88] ——, “The impact of routing and storage policies on warehouse efficiency,” *International Journal of Operations & Production Management*, vol. 19, no. 10, pp. 1053–1064, 1999. [Online]. Available: http://dx.doi.org/10.1108/01443579910287073

[89] T. S. VAUGHAN, PETERSEN, and C. G., “The effect of warehouse cross aisles on order picking efficiency,” no. November, 1999.

[90] Y. I. H. Parish and P. Müller, “Procedural modeling of cities,” in *Proceedings of SIGGRAPH 2001*, 2001, pp. 301–308.

[91] B. Weber, P. Müller, P. Wonka, and M. H. Gross, “Interactive geometric simulation of 4d cities,” *Computer Graphics Forum*, vol. 28, no. 2, pp. 481–492, 2009.

[92] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang, “Interactive procedural street modeling,” *ACM Trans. on Graph.*, vol. 27, no. 3, pp. 103:1–9, 2008.

[93] M. Lipp, D. Scherzer, P. Wonka, and M. Wimmer, “Interactive modeling of city layouts using layers of procedural content,” *Computer Graphics Forum*

(Proceedings EG 2011), vol. 30, no. 2, pp. 345–354, Apr. 2011. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2011/lipp2011a/

[94] D. Aliaga, C. Vanegas, and B. Benes, “Interactive Example-Based Urban Layout Synthesis,” *ACM Trans. on Graph.*, vol. 27, no. 5, 2008.

[95] D. G. Aliaga, B. Beneš, C. A. Vanegas, and N. Andrysco, “Interactive Reconfiguration of Urban Layouts,” *IEEE Computer Graphics and Applications*, vol. 28, no. 3, pp. 38–47, 2008.

[96] S. V. U. Ignacio Garcia-Dorado, Daniel G. Aliaga, “Designing large-scale interactive traffic animations for urban modeling,” *Computer Graphics Forum*, 2014.

[97] E. Galin, A. Peytavie, E. Guérin, and B. Benes, “Authoring hierarchical road networks,” *Computer Graphics Forum*, vol. 29, no. 7, pp. 2021–2030, 2011.

[98] N. Maréchal, E. Guérin, E. Galin, S. Merillou, and N. MÃľrillou, “Procedural generation of roads,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 429–438, 2010.

[99] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977.

[100] K. Lynch, *The Image of the City*. MIT Press, 1960.

[101] M. Southworth and E. Ben-Joseph, *Streets and the Shaping of Towns and Cities*. Wasington DC: Island Press, 2003.

[102] M. Southworth and P. M. Owens, “The evolving metropolis: Studies of community, neighborhood, and street form at the urban edge,” *Journal of the American Planning Association*, vol. 59, no. 3, pp. 271–287, 1993.

[103] S. Marshall, *Streets & Pattern*. New York: Spon press, 2005.

[104] A. P. Association, *Planning and Urban Design Standards*. Wiley, 2006.

[105] B. Hillier, *Space Is The Machine: A Configurational Theory Of Architecture.* Cambridge University Press, 1996.

[106] Z. Y. J. Peponis, S. Bafna, "The connectivity of streets: reach and directional distance," *Environment and Planning B: Planning and Design*, 2008.

[107] T. R. Board, *Highway Capacity Manual.* Transportation Research Board, 2010.

[108] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *EUROGRAPHICS State-of-the-art Report*, 2012.

[109] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A Survey of Urban Reconstruction," in *EUROGRAPHICS 2012 State of the Art Reports.* Eurographics Association, 2012, pp. 1–28. [Online]. Available: http://www.cg.tuwien.ac.at/research/publications/2012/musialski-2012-sur/

[110] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Proc. of SPIE*, vol. 4387, 2001.

[111] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas, "Discovering structural regularity in 3D geometry," *ACM TOG*, vol. 27, no. 3, pp. 43:1–43:11, 2008.

[112] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *Proc. ACM SIGGRAPH*, vol. 28, no. 3, Aug. 2009.

[113] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE PAMI*, vol. 32, pp. 1362–1376, 2009.

[114] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, "Interactive images: Cuboid proxies for smart image manipulation," *Proc. ACM SIGGRAPH*, vol. 31, no. 4, pp. 99:1–99:11, 2012.

[115] A. Myers, *Complex System Reliability*, 2nd ed., ser. 1614-7839. Springer-Verlag London, 2010.

[116] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, pp. 617–680, 1983.

[117] A. Furieri. Readosm open source library. [Online]. Available: https://www.gaia-gis.it/fossil/readosm/index

[118] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge: Cambridge University Press, 2000.

[119] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[120] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[121] ——, “Building the gist of a scene: the role of global image features in recognition,” *Progress in Brain Research*, vol. 155 B, pp. 23–36, 2006.

[122] L. Fan and P. Wonka, “Structure Completion for Facade Layouts,” 2014.

[123] L. Nan, H. Jiang, L. Nan, D.-m. Yan, W. Dong, X. Zhang, and P. Wonka, “Automatic Constraint Detection for 2D Layout Regularization Automatic Constraint Detection for 2D Layout Regularization,” no. SEPTEMBER, 2015.

[124] D. G. Aliaga, G. Nishida, and D. Aliaga, “Example-Driven Procedural Urban Roads,” no. SEPTEMBER 2015, 2016.

[125] N. Ferreira, H. Werner, and M. Park, “Urbane : A 3D Framework to Support Data Driven Decision Making in Urban Development,” pp. 97–104, 2015.

APPENDICES

A Appendix A - Awards

- The Google Anita Borg Memorial Scholarship Award, 2013
- KAUST Fellowship, KAUST, Saudi Arabia, 2010 - 2016
- KAUST Discovery Scholarship, KAUST, Saudi Arabia, 2008 - 2010