Pedestrian Patterns and Spaces: Modelling Visitor Engagement Dynamics at the Jeddah Northern Waterfront

by

Lubaba Fakeih

September 2021

Supervisor: Prof. Kerstin Sailer

A Dissertation submitted in part fulfilment of the

Degree of Master of Research (MRes) Built Environment

Space Syntax: Architecture and Cities

Bartlett School of Architecture

University College London

ABSTRACT

This research studies the Jeddah Waterfront in an attempt to first model, then understand the pedestrian-waterfront engagement dynamics. To understand potential factors driving pedestrian engagement, the research explores theories of natural movement, attractors, and prospect and refuge. To contextualize the pedestrian patterns, this study creates a model of the spatial characteristics of the waterfront. Modelling a waterfront space using current VGA tools was challenging. A new tool for creating visibility graphs then was explored and adapted, with components allowing the use of directed visibility graphs, night-time lighting and weighted visual attractors. The visibility results were then tested against pedestrian distribution, based on the snapshot observation method. Where the cumulative isovist count for each grid point was correlated with the predicted visibility from the graph with a r^2 correlation value of 0.78. Pedestrian activity subgroups were then defined, the most common being: observing, settling, resting and travelling. These categories were then explored and associated with various spatial metric patterns resulting in a set of design recommendations for the ongoing Jeddah Waterfront expansions.

CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1
RESEARCH OVERVIEW

1.1. PROBLEM STATEMENT

Jeddah is a coastal city, with its waterfront being its most valuable public resource. Following its renovation in 2017, The Jeddah Waterfront was made open to the public with a high degree of public reception. With parts of this mega-project still under construction, this research aims to study and understand how the public engages with the newly renovated Jeddah Waterfront, what parts are well-utilised and what parts are neglected, in an effort to optimize the design of future expansions and better serve the waterfront visitors.

However, modelling the visual accessibility of landscaped spaces is a separate challenge that is not yet fully covered by todays tools. This gap is made clearer in the case of Jeddah Waterfront, where not only is it a large landscaped space, but it is an outdoor space – with temporal shifts affecting levels of visibility. Additionally, it neighbours the sea – a large visual attractor that is a main reason for pedestrian foot traffic. This creates a complex structure – where some visual aspects (sea view) may be more valuable than others (landscape). In an aim to produce a more refined visual study of complex outdoor spaces, this study attempts to create a model that can accurately portray the visual potential of an urban green space based on the landscape elements present, lighting conditions, and nearby attractors.

By undertaking this issue, this research is tackling two main problems, one being the abstract problem of modelling the waterfront, with the hopes of creating a systemic model that can later be used by others in similar applications, and one being the assessment of the Jeddah Waterfront for an enhanced pedestrian experience.

1.2. RESEARCH QUESTION(S)

The waterfront is the largest public space in the region, and a destination for all. It is an integral part of Jeddah's identity and it is central to establishing the notion of placemaking in the city. In addition to the recently constructed 5 km linear stretch, large extensions to both the north and south are currently underway. It is then crucial to ask:

How are the people engaging with the new waterfront and can the design layout be enhanced for maximum pedestrian interaction?

This research question can be further divided into sub-questions that would then be answered by the upcoming chapters in this research, as illustrated in figure 1.1. These questions include the following:

- What is pedestrian engagement, and how can it be defined?

- What tools can we use to measure public engagement?

- How is the public engaging with the waterfront?
    - *On a macro level*: *When* do people visit the waterfront? With the waterfront being a 5 km stretch of land, what *waterfront sub-section* is the most active?
    - *On a micro level*: Where are the different *visitors allocated* within the waterfront spaces, and what are their *activity patterns*? With Jeddah being a night-time city are activity patterns similar over the course of the day?

- How can we optimize the use of space and to better serve visitors in the design of future waterfront expansions?

1.3. RESEARCH METHODOLOGY

This research adopts a standard case study approach where an appropriate combination of quantitative and qualitative measures is used to assess the problems that arose in this particular case. The research addresses the waterfront engagement dynamics on a ranging scale of macro to micro; where macro addresses questions measuring pedestrian activity on a larger scale, meso assesses the waterfront as a space providing spatial context, and micro answers questions on pedestrian activity patterns and allocation the waterfront.

On a macro scale, online analytics are utilized for automated data collection on a larger scale. Spatial assessment tools such as a DepthMap are also used to understand the placement of the waterfront on the larger axial map of Jeddah. On a meso scale Space Syntax tools are adapted to model movement patterns and visibility in the waterfront. To model routes of movement, DepthMap is used to create segment models – adjusted on QGIS. To model visibility, site visits for data collection and a combination of software for analysis are used, including DepthMap for calibration and analysis, Java for custom tool development, and QGIS for visualization. On a micro scale, pedestrian observation techniques such as the snapshot method are adopted and expanded upon, using software such as Java to map out pedestrian isovists and lines of sight.

All in all, the result of this project would be a more in-depth understanding of Jeddah Waterfront on a multitude of levels, including both spatial characteristics of the space and the visitor engagement dynamic and experience. helping to optimize visitor experience and future design.

1.4. RESEARCH STRUCTURE

To fulfil the aims of this research – the upcoming chapters would be structured as follows to answer the respective sub-questions defined in the previous segment.

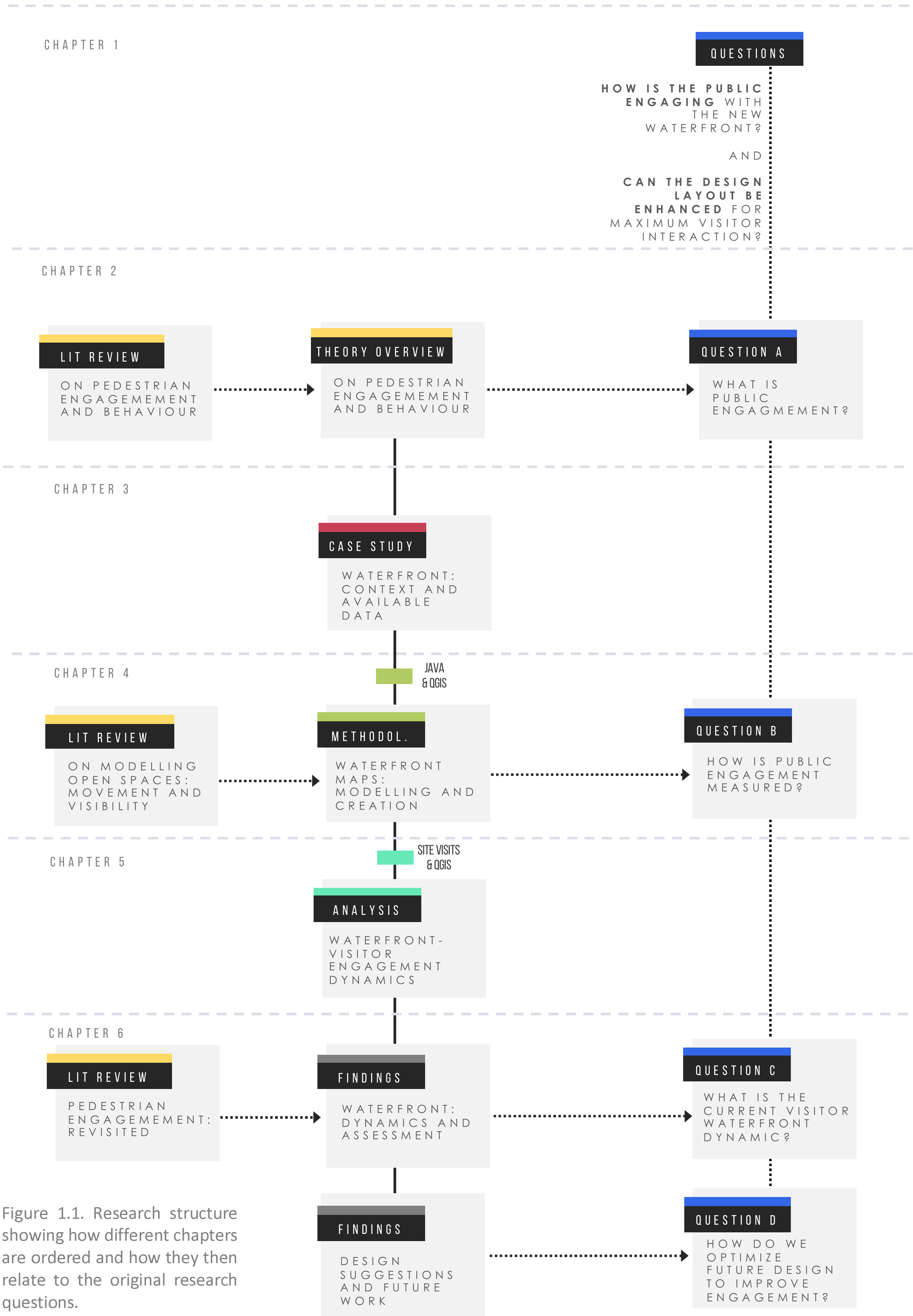


Figure 1.1. Research structure showing how different chapters are ordered and how they then relate to the original research questions.

CHAPTER 2

ON VISITOR ENGAGEMENT PATTERNS IN THE SPATIO-TEMPORAL REALM

2.1. CHAPTER OUTLINE

To better understand the link between the spatial characteristics of the public waterfront space and its subsequent visitor engagement rates, each theoretical component needs to be explored individually. As such this literature review will be divided into three separate, but inter-related, sections. The first section will explore theme of space. This section will highlight the importance of the spatial context and how it defines an area's *engagement potential*. The segment will introduce the concept of space, and the inherit characteristics it holds via its placement on a larger spatial network. The second section explores pedestrian movement patterns. Here, theories on natural movement and attractors will be explored, comparing the value of spatial context to spatial content. The third, and final, section will explore the concept of pedestrian engagement – turning towards theories on prospect and refuge, and the effect of visibility on pedestrian behaviour and interaction with the surrounding spaces. This is especially important in studying the effects of visibility in the Jeddah Waterfront two main reasons. Firstly, the existence of the sea, a visual attractor, emphasises the importance of visibility as a prospect. Secondly, privacy is of a high cultural value in Saudi Arabia, where many culturally reserved groups seek out private areas in public spaces. By understanding the theories behind pedestrian movement and interaction, this research can then begin to link together the spatial characteristics of the Jeddah Waterfront and its subsequent effects on pedestrian activity patterns.

2.2. ON SPACE: THE INHERENT SPATIAL CONTEXT OF OBJECTS

No tangible object embedded in space is devoid of spatial context. Douglas describes it best when questioning what makes a house a home (1991). In her work The Idea of a Home: A Kind of Space, she talks about all the immaterial elements that are combined to construct a sense of order and attachment to the home. One example is the concept of a dining table – where it is more than just a table. It is a place of gathering – established by its spatial positioning in a non-secluded region of the house. It is open to guests and dinner parties – where it displays that potential by its relative shallow-ness in terms of space. It also has a sense of temporality – where it is associated with certain times of the day. It is a table, but its spatial positioning – as well as the context imposed upon it by its residents – transforms it into an object with meaning.

On a larger scale, the meaning behind objects' spatial positioning can also be explored in the building allocation examples found in the Social Logic of Space (Hillier & Hanson, 1984). Significant community buildings, such as churches, are typically found in spaces that are accessible to the direct community but somewhat segregated from the exterior larger realm. This distinction between the more public, or shallow, spaces, and the more private, or deep, spaces can also be found within buildings. The functionality of spaces often separates the residents of a neighbourhood, to where they occupy the deeper spaces, and the strangers occupy the shallower areas.

In this context, an interaction with a space or design element is limited or encouraged by its distinctive spatial position in the overall global network of spaces. The study of the spatial network is therefore a fundamental part of understanding the context surrounding a space, and subsequently, understanding the *engagement potential* a space inherently holds. The positioning of a feature within an existing overall network can either magnify or diminish its intended effect. Global positioning, visibility, and service availability all have a large effect on public reaction and behaviour within a designed space. Foot traffic and contact with an area has been found to correlate to the integration values of the space (Peponis et al., 2004). Visibility has also been linked to increased levels of engagement, where an individual design feature or space is greatly influenced by the larger spatial arrangement; the visual accessibility of an activity point or a space from a neighbouring area has been shown to influence how frequented the feature is (Peponis et al., 2004). The availability of seating areas and places to rest, as well as availability of nearby services increases the potential of space revisitability (McLane & Kozints, 2019). Atmospheric context of where the features are allocated can also shape emotional responses from visitors. Tzortzi and Schieck (2017) found a relationship between certain types of exhibits and where they were located – where interactive and playful exhibits seemed to be placed in more integrated and immersed areas, and more educational and informative exhibits were placed in segregated areas to offer an air of introspection. These surroundings are then an integral part of the inherent characteristics of a space.

2.3. ON PEDESTRIAN ACTIVITY: THE THEORY OF NATURAL MOVEMENT VS ATTRACTORS

Spatial embedding drives pedestrian traffic. The *theory of natural movement* talks about how the distribution of pedestrian density along spaces in a network is a natural consequence of the spatial hierarchies of streets or spaces. To further elaborate: a central street system with one large street connecting to smaller branching sub-streets would carry more pedestrian traffic than a distributed street system with no central street. Firstly, in the central system, through-movement would all occur through that one main street (as it is the fastest, and at times the only, option to get from point A to B). Secondly, potential for to-movement is also created due to the short mean depth (or shallowness) of the central street to the overall network (Hiller, Penn, & Hanson, 1993). The theory states that the pedestrian foot traffic comes first, and the retail and commercial centres come later – either for the centrality and shallowness of the location within the network, or for the relatively high pedestrian density. This then creates a multiplier effect, further increasing foot traffic and acting as 'attractors' (Hillier, 1996). The value a space holds in terms of integration then reflects on its ability to generate movement and encounters (Peponis, 1993). Many research papers have since gone on to confirm this relationship; recently, a research paper by Hacar, Gulgen and Bilgi (2020) used regression analysis to establish a relationship between the integration value of streets and pedestrian density, while another paper by Monokrousou & Giannopoulou (2016) showed a very strong correlation between topological segment analysis and pedestrian density in Athens. According to the Space Syntax theory, spatial position, and not necessarily activity points or services, encourages engagement; or at the very least, determines engagement potential.

Attractors, however, are certainly a factor of influence when it comes to pedestrian movement. While the theory of natural movement has been shown to correlate to pedestrian movement on larger scales – studies carried out on a more micro-scale have had varying results. In a study on seating preference, Psathiti and Sailer have shown a higher correlation in seating choice to spatial characteristics only in the absence of attractors (2017). Route choice has also been shown to shift away from the more integrated routes of movement in the presence of attractors, requiring an adjustment to the classic Space Syntax model to better predict movement patterns (Sailer, 2007). Spatial characteristics of an area are then indicative of movement patterns only in a neutral space or in larger studies where the effects of individual attractors are neutralised by sample size. Otherwise a combination of local attractors and spatial configurational patterns have been shown to best predict pedestrian movement (Sailer and Penn, 2009).

2.4. ON VISITOR ENGAGEMENT

Visitor engagement is defined as prolonged contact or interaction with a space, as opposed to visual contact or through-movement, in terms of pedestrian allocation (Peponis et al. 2004). Visibility, in general, plays a significant role in determining visitor engagement and movement patterns. This effect ranges from interaction with exhibits, where Peponis et al. (2004) have established a link between exhibit engagement and their respective rates of visibility, to interaction with spaces. A paper on experiencing museum layouts through visibility talks about how isovist length and occlusivity predict user movement choices. Where visitors stopping to observe *space* (as opposed to the individual exhibits) correlated with more open spaces, conversely, visitors engaging with exhibits gravitated to more visually secluded spaces (Rohloff, Psarra, & Wineman, 2009).

More specifically, visibility has been linked to the theory of prospect and refuge, a concept describing pedestrian behavioural patterns, with applications in landscaped spaces, amongst others. The theory of prospect and refuge alludes to the human need of safety and stimulation, linked to the need to see (prospect), without being seen (refuge) (Dosen and Ostwald, 2013). In their paper *Prospect and Refuge Theory* (2013) Dosen and Ostwald explore Appleton's descriptions of types of prospect (including panoramas and vistas, with observation decks listed as areas with high vantage points) and types of refuge (including built structures and vegetation refuges). These examples would be touched upon later in the research to identify spaces as potentials for both areas of prospect and areas of refuge.

Measuring how open or safe a space can be is less straightforward and findings are times inconclusive (Dosen and Ostwald, 2012). This theory is often associated in literature with the use of isovist metrics to understand the visual properties of a space. A paper on the concept of *schema theory* visually mapped out the reoccurring pattern of experiences visitors go through while following a set route. Measures of mean depth, connectivity, maximum radials, and occlusivity of a point were used to reflect the space's cognized publicness, its perceived publicness, its depth of view, and its sense of perpetual uncertainty, respectively (Zook, 2017), where openness and implies prospect – and occlusivity implies refuge. Spatial characteristics such as openness, area size, and perceived portion of the enclosed space have all been used to measure prospect and refuge of indoor spaces (Psathiti and Sailer, 2017).

Additionally, the theory of prospect and refuge could be studied using outdoor space at night. Studying pedestrian activity at night, both Fotios, Unwin and Farral (2015), and Tagliabue et al. (2002), discuss reassurance and perceived safety and how that affects pedestrian movement choices. Results have shown that routes with more exit points (as opposed to long paths that allow pedestrians less movement options) are preferred over those without, and routes that give more room for looking ahead (prospect) are favoured over more occlusive paths. Those favouring prospect would gravitate towards the more well-lit areas. As refuge is associated with a sense of safety and limited visual vulnerability, it follows that the lighting has an added effect on seeing and being seen, where those looking for refuge might seek dimly lit spaces.

2.5. KEY FINDINGS

In summary, studying the spatial characteristics of the waterfront is integral in understanding the potential the space has in terms of user engagement. Pedestrian patters are expected to follow the configurational logic to a certain extent but may be redirected due the existence of a large attractor, the sea, which could require adaptation of existing tools to model pedestrian patterns. According to the theory of prospect and refuge, visual characteristics of a space often guide pedestrian behaviour on a micro-scale. Methods to measure visibility and user patterns, as well as methods on modelling spaces and general waterfront visibility will then be explored in the following chapters.

CHAPTER 3

JEDDAH'S WATERFRONT: A RECREATIONAL PUBLIC SPACE

3.1. CHAPTER OUTLINE

This chapter aims to introduce the Jeddah Waterfront site in specific, and the city of Jeddah in general. It will first introduce the role Jeddah plays in the Saudi cultural context. It will then focus on the importance of Jeddah's coastline, a.k.a. the Corniche, as an integral part of the city's cultural identity. The chapter will then introduce the different parts of the existing pedestrianised coastline strip, elaborating on the Northern Corniche Waterfront, the focus of this project. Recent and future public events taking place in the Northern Corniche Waterfront vicinity have elevated the significance of the Jeddah Waterfront even further. These include local events, such as the Jeddah Season festivities, and international events, like the upcoming Formula 1 Saudi Arabian Grand Prix. The chapter will end by expanding upon the different subsections of the Northern Corniche Waterfront project to help with the familiarization process with the site in question.

3.2. THE JEDDAH WATERFRONT: AN OVERVIEW

Jeddah's pedestrianised coastline, known locally as the Corniche, extends along the city form north to south, bordering the Red Sea for around 11 km. The previous coastline region had paved areas overlooking the sea, with few allocated seating areas and small-scale artistic sculptures. Within the past 5 years, a mega-project began surrounding the renovation of the coastline, known today as the Jeddah Waterfront, to create a more formally designed recreational space – complete with playgrounds, sustainable landscape, water features, seating areas, viewing docks and more (Mostafa, 2017).

3.2.1. The Urban Context of Jeddah

Located in the western region of Saudi Arabia [figure 3.1], Jeddah is a major commercial city with an area of 1,600 km2 and around 4 million inhabitants. It is the second largest city in the country, second only to the capital, Riyadh. With a climate classified as arid, temperatures in Jeddah remain warm throughout the year. At summer, the temperature often reaches 50°C, with precipitation rates falling under 70 mm of annual rainfall (Youssef et al., 2015).

Locally, the country's population is growing at an accelerated rate – with the inhabitants doubling from 16 million in 1990, to 30 million in 2010, with an additional 10 million increase in growth forecasted by 2030 (Belarem et al., 2018). The country has also experienced a large shift in lifestyle. Today, 95% of the current inhabitants live a sedentary lifestyle, with 78.8% of the population residing in cities. A stark contrast to the country's mostly nomadic population in 1960 (Ali and Ameur, 2018).

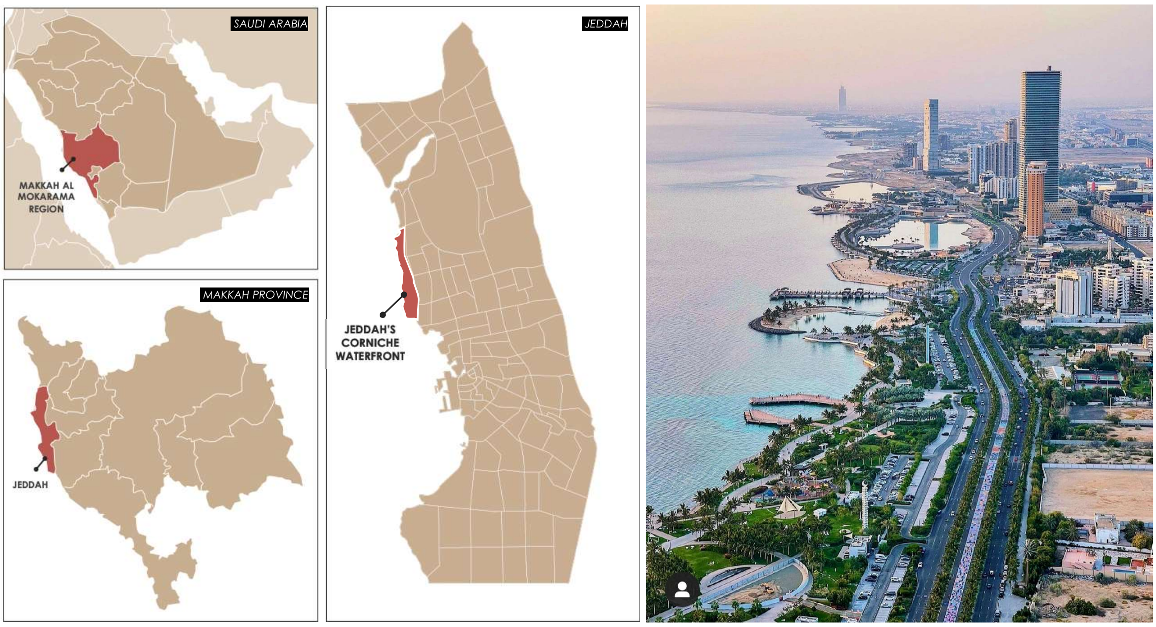


Figure 3.1. Aerial view of the Northern Corniche Waterfront (Source: Thabit, 2021) [right] and location of corniche in Jeddah, and on a larger scale, Saudi Arabia [left].

Globally, as it is the official gateway to the two holy cities, Makkah and Madinah, the city sees a large influx in global visitors. This is especially true during the holy months of *Ramadan* and *Dhul-Hijja* – where Muslims worldwide arrive to perform their yearly pilgrimage (Arab News, 2021). In 2019 the country opened its doors to tourism, creating yet another boom in international visitors and tourists all year around (BBC News, 2019).

3.2.2. The Waterfront as a Cultural Symbol

The coastline is one of the city's key attraction points. Jeddah is a linear city build around the coastline. With around 40 km of coastline, it is home to the largest port on the Red Sea (Helmy, 2017). As such the coast of Jeddah is an attraction within itself. Known locally as the 'Corniche', the coastline hosts a variety of pedestrian spaces – including a main road running parallel to the coastline, distributed sandy beaches, walkways, restaurants and assorted activities.

Jeddah's climatic conditions does not encourage public gathering in outdoor spaces. As such there are a limited number of large public outdoor recreative spaces, open to the general population. While Jeddah hosts around 200 small district level parks and open spaces, the Corniche strip is perhaps the only large public outdoor space of its kind, equipped with seating areas and few landscaped spaces (Mostafa, 2017).

In 2011, the municipality took on the mega project of renovating the Corniche strip into what is today known as the Jeddah Waterfront [figure 3.1], choosing the symbol of the seagull to represent the newly developed coastline. This project aimed to develop the existing Corniche coastline into a recreational space for locals and visitors, with a budget of SR800 million (£154 million).

The main phases of the project development include: the *Northern Corniche Waterfront* development, the *Central Corniche Waterfront* development, the *Southern Corniche Waterfront* development, the *Open Museum*, *Palestine Street* (located adjacent to the coastline in central Jeddah), and *South Obhur Corniche development* (where Obhur is a district located in far north of the city) (Mostafa, 2017).

As the coastline, along with its pedestrian spaces, covers a significant portion of land, this project will focus on one portion of the coastline: the recently renovated *Northern Corniche Waterfront*. Spanning about 5 km, with an area of 73 hectares, the project was open to the public by 2017 – with an ongoing expansion of the renovated waterfront under construction (Arab News, 2018).

3.2.3. The Rise of Public Spaces and Events

Since its opening to the general public, The Northern Corniche Waterfront has grown in popularity by hosting several public events amidst the Jeddah Season festivities and will soon house the Formula 1 Saudi Arabian Grand Prix circuit.

The Northern Corniche Waterfront has been an integral part of the Jeddah Season festivities, taking place in 2019. Jeddah Season is a part of a larger government-sponsored initiative (*Saudi Seasons*) created in an effort to align with the new 2030 vision; where the 2030 vision is a strategic framework aiming to reduce the country's dependency on oil by developing public service sectors, such as recreation and tourism (Krimly, 2019). Consisting of a series of festivals around the country, the *Saudi Seasons* initiative hosts cultural, sports, and entertainment events funded by the government – with each major city hosting its own festival at designated parts of the year. Jeddah Season first started in June 2019, lasting for two months; the season saw a variety of workshops, circus acts, water sports, limited opening of global restaurants, open air cinemas, art exhibits, concerts, comedy shows, sports events, and more. These events were hosted in 5 separate regions of the city: with water sports limited to *Obhur* (northern Jeddah), heritage events in *Al Balad* (historic centre of Jeddah), food festivals in *Al Hamra* (central Jeddah), sports events and concerts in *King Abdullah Sports Stadium*, and assorted entertainment and activities in the newly renovated *Northern Corniche Waterfront* (Francis, 2020).

Today, the Northern Corniche Waterfront is being prepped to host the Formula 1 Saudi Arabian Grand Prix in December 2021. The circuit was designed to be around 6 km long, making it the second longest circuit in the sport – aiming to be the fastest street circuit in history of the race (Saudi Gazette, 2021). As it is the first race to take place in the country, this event is expected to draw global attention to the Northern Corniche Waterfront, increasing both real-life exposure of the site and virtual media coverage.

3.2.4. The Northern Corniche Waterfront: A Walkthrough

The Northern Corniche Waterfront covers a 4.5 km stretch of land, designed for allowing around 120,000 visitors at a time with over 700,000 m^2 of paved areas for pedestrian movement and fully landscaped regions (Alawi and Mostafa, 2019). The renovated waterfront is equipped with seating areas, playgrounds for kids, kiosk and cafes, sandy beaches, an open-air gym, viewing decks, fishing piers, parking for ±3000 cars, interactive water features, art pieces and sculptures, and a linear promenade running parallel to the recreational waterfront used as a pedestrian walkway, a biking lane and a jogging lane for those wanting to exercise.

The Northern Corniche Waterfront is segmented into 7 themed areas [figure 3.2]: the Nawras strip (a), the Shell strip (b), the Unification strip, (c) the Sand strip (d), the Pearl strip (e), the Fisherman strip (f), and the Gulf strip (g). Each strip was named after its design contents – offering pedestrians a varied selection of activities and visuals, depending on the area (Jeddah Municipality, 2019).

Figure 3.2. Jeddah Northern Waterfront 4.5 km strip recreated using Google Earth satellite images, showing the different waterfront sub-divisions.

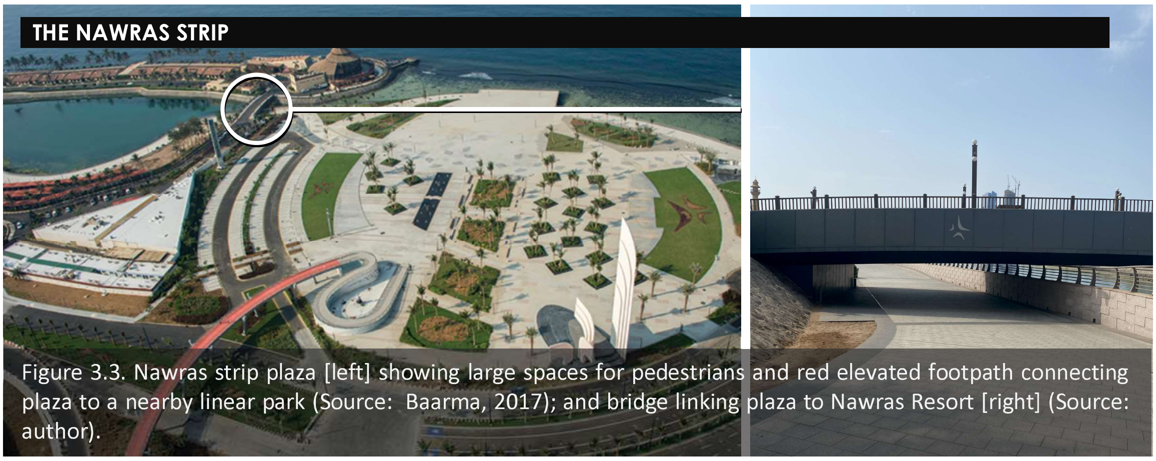


Figure 3.3. Nawras strip plaza [left] showing large spaces for pedestrians and red elevated footpath connecting plaza to a nearby linear park (Source: Baarma, 2017); and bridge linking plaza to Nawras Resort [right] (Source: author).

Nawras Strip (a)

Named after the large iconic seagull sculpture it houses, a major landmark in Jeddah, where “Nawras” is an Arabic word for seagull. Once at the centre of a roundabout, the sculpture was preserved in the Northern Corniche Waterfront renovation and was made the focal point of this area [figure 3.3]. This strip is by far the largest and most culturally significant area of the 7. This region also serves as the connection between the waterfront and the neighbouring Nawras Resort. A historic hotel chain built on an island overlooking the sea. The resort is soon to be renovated into a recreational strip housing comedy clubs, a cinema, a public beach, and a variety of restaurants and cafes linked to the southern extension of the waterfront by a series of bridges. The Nawras strip also features an elevated walkway, giving pedestrians an aerial view of the waterfront and linking the extended walkway running parallel to the waterfront to the newly developed region.

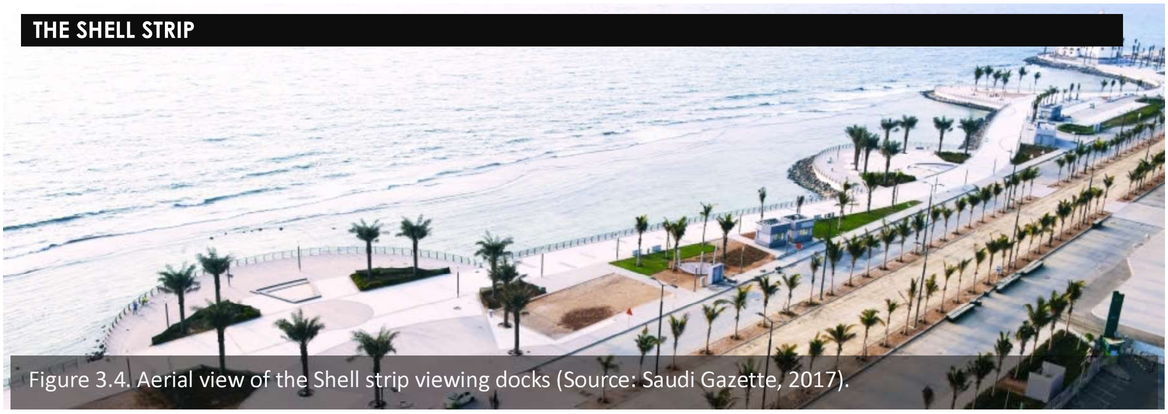


Figure 3.4. Aerial view of the Shell strip viewing docks (Source: Saudi Gazette, 2017).

Shell Strip (b)

Named for the shell-shaped observation decks it houses [figure 3.4]. This strip is a linear paved area with observation decks that are easily accessible from the main road.

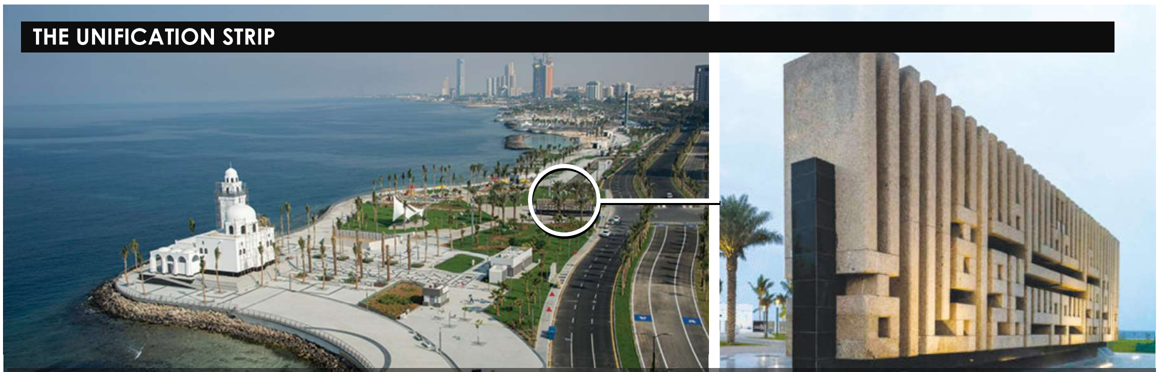


Figure 3.5. View of central mosque in Unification strip [left] (Source: Baarma, 2017), and unification sculpture upon which strip was named [right] (Source: Arab News, 2018).

Unification Strip (c)

Named for the artistic sculpture located in the centre of the strip depicting the Islamic phrase:

لا إله إلا الله محمد رسول الله
(There is no God but Allah, Muhammed is the Messenger of God)

This strip also houses a mosque [figure 3.5], and a variety of themed playgrounds for kids.

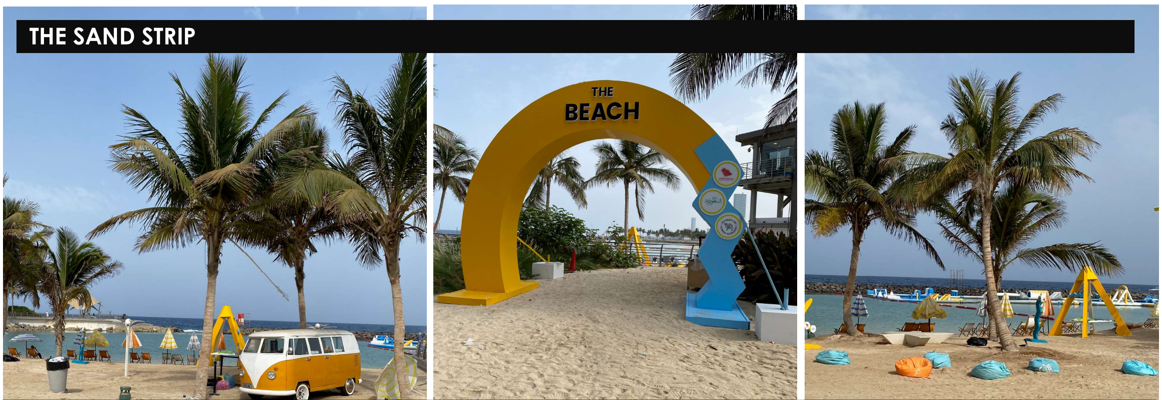


Figure 3.6. Assorted views of the beach facilities located in the Sand strip (Source: Author).

Sand Strip (d)

Named for the sandy beach it encompasses [figure 3.6]. This is the first fully serviced public beach in Jeddah, with a collection of water activities and sports. The beach was briefly made open to the public on August 2, 2021, for a period of 2-3 weeks before closing back up for reasons not made public. It is unknown when the beach will be open to the public again.

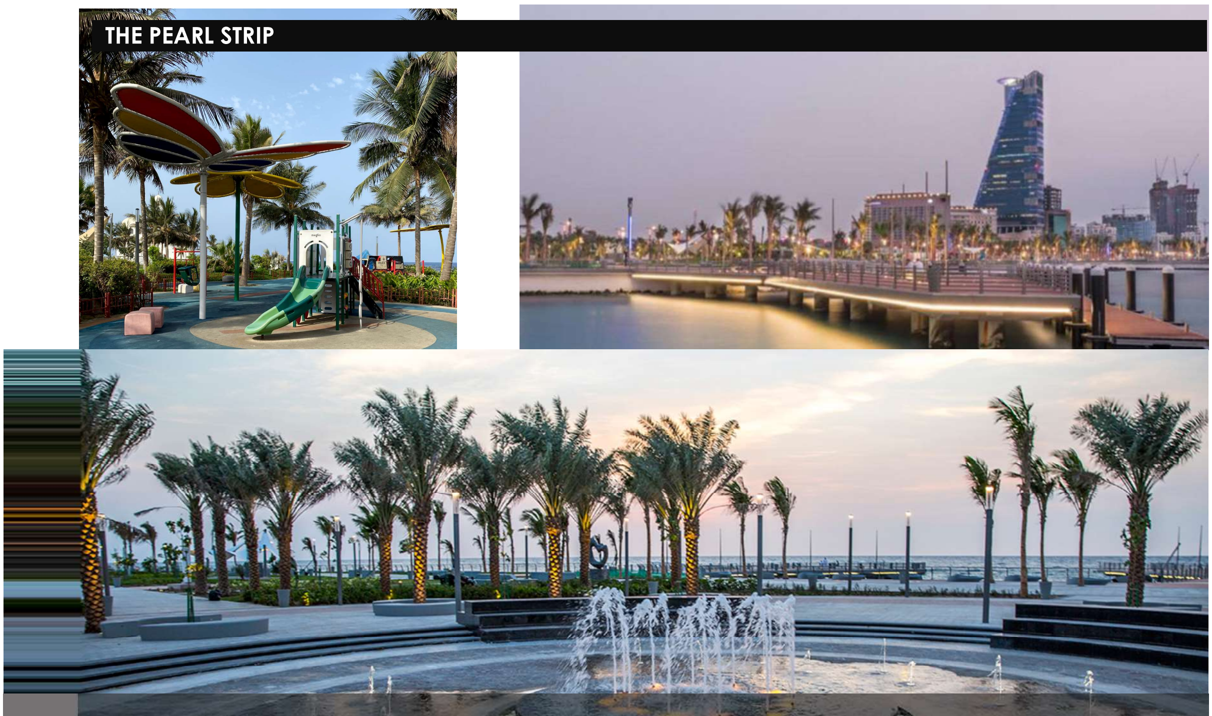
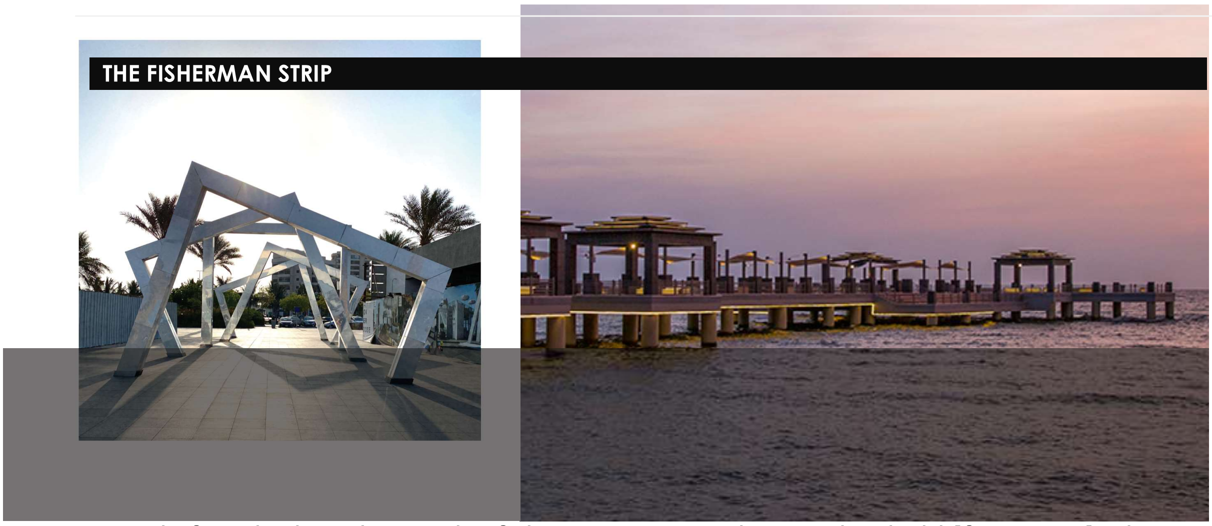


Figure 3.7. Pearl strip facilities including playgrounds [top left] (Source: Author), piers [top right] (Source: Gulf News, 2017), and central plaza with water features [bottom] (Source: Jeddah Municipality, 2019).

Pearl Strip (e)

Named after the pearl-inspired artistic sculptors scattered along the strip. Aside from the open museum of artistic sculptures, the area also contains a large central water feature, several playgrounds, landscaped spaces and parks, and extended piers [figure 3.7].



Named after the largely popular fishing pier it was designed to hold [figure 3.8]. This strip also has a variety of cafes, playgrounds, an open gym and landscaped spaces.

Gulf Strip (g)

Named after its protruding land formation that extends outwards to sea, it is expected to contain the highest interactive fountain in the Northern Corniche region. This strip is currently under construction and has not yet been made accessible to the public.

3.3. KEY FINDINGS

The Jeddah Waterfront is a key cultural symbol for the city and one of the only large recreational outdoor spaces available for the general public. The recent renovation of the Northern Corniche Waterfront, along with the public events hosted within the waterfront boundaries, have led to the increased popularisation of the site. With the capacity to hold over 120,000 visitors at a time, the mega-project is subdivided into 7 themed segments (from south to north): the *Nawras* strip, the *Shell* strip, the *Unification* strip, the *Sand* strip, the *Pearl* strip, the *Fisherman* strip, and the *Gulf* strip. With the final segment, the *Gulf* strip, still under construction. The remainder of this research will focus only on the Northern Corniche Waterfront; the terms Jeddah Waterfront and Northern Corniche Waterfront will be used interchangeably from this point onwards.

CHAPTER 4
TOOLS FOR MEASURING PUBLIC ENGAGEMENT

4.1. CHAPTER OUTLINE

To address the research questions, this chapter will define various methods and tools to analyse the Jeddah Waterfront. This chapter will explore various methodologies on scales ranging from macro to micro. It will first explore techniques used to collect traffic data using online analytics to assess the site on a macro level. Here methods on extracting pedestrian activity levels from Snapchat heat map are explored in an effort to understand pedestrian distribution on a larger scale. Live Google Traffic data is also used to understand peak hours. By understanding *when* pedestrians visit, and *where* they are the most active, the scope of study can be better defined. The next segment will explore methods relevant to modelling the waterfront space on a meso level. This part explores current segment and visual graph models, and their respective limitations when it comes to landscaped spaces in general, and waterfronts in specific. Here an adapted segment model for landscaped spaces will be explored, and a new visibility graph model will be created to represent the Jeddah Waterfront visibility levels in both the day and night. By understanding pedestrian spaces, a framework is created to contextualise any pedestrian patterns explored in the following section. Finally, methods to understand pedestrian engagement on a micro level are discussed; these include the snapshot method and a subsequent study of pedestrian isovists.

4.2. MACRO: UTILIZING ONLINE ANALYTICS

To measure engagement on a macro level, social media and online metrics were used to aid in engagement level monitoring. Social media is gaining popularity as a measuring tool for data analytics on the population's spatiotemporal activities. The use of Location-Based Social Networks (LBSNs) helps gather information on both macro and micro scales. Where on the macro scale it tracks public behaviour and movement by sharing spatiotemporal information on the individual with public, and on the micro scale the content being shared is telling of more minute details such as likes, dislikes, and interests. Essentially, social media and online metrics can be used to estimate pedestrian and vehicular traffic in a certain region, at a given time. Botta, Moat and Preis (2020) used Instagram to estimate the real-time size of crowds in football stadiums. Juhasz and Hochmair (2018) used Snapchat's heat map to trach pedestrian density over three different cities: Miami, LA, and New York. Another paper used Snapchat's heat map feature to track crowd intensity over time at the holy mosque in Makkah, Saudi Arabia (AlGhamdi, AlRajebah & AlMegren, 2019). Google Traffic Live has also been used as valid data source for research on prediction of urban air quality (Zalakeviciute et al., 2020) and traffic congestion patterns (Ramírez, 2020).

This research utilized the available online metrics of two digital platforms: Snapchat and Google Traffic – to track pedestrian activity within the waterfront vicinity. The Snapchat heat map was used to track *where* people were the most active, while live Google Traffic data was used to track *when* people visited the waterfront. To support Google Traffic data in assessing vehicular traffic flow, an axial map of the city was created and used. However, as the concept of an axial map will be described in the following section, and no special methodology is required to create this map, there will be no further discussion on this particular map in this chapter.

4.2.1. Monitoring Activity Levels: Snapchat Heat Map

To understand pedestrian activity levels on a larger scale – the Snapchat heat map was used for data extraction. The Snap Map is a feature on Snapchat, with a global heatmap based on the number of public posts in a given location. Snapchat is a popular social media platform amongst the Saudi population. With 19.6 million users, Saudi Arabia ranks 5th in the world in number of users (Omnicore, 2021). As posting on a social media platform requires a level of engagement with a space or design element, the heatmap is representative of the engagement level per area. The heatmap, however, is not representative of the entire demographic spectrum. Around 80% of all users are under the age of 35 (Omnicore, 2021), skewing results to show where the younger population is active, as opposed to the average engagement level of the entire waterfront demographic. That being said, the heat map was still utilized to gather information and will later be supported by site visits and traffic data from the following section, which will include a less bias population sample, to validate the represented activity levels.

The Snap Map was used to collect the total number of posts across the Northern Corniche Waterfront – to understand visitor activity/engagement level distribution across the waterfront region. To gather data from the Snapchat heat map, a screen shot of the waterfront region was taken every 3 hours (6 am, 9 am, 12 pm, 3 pm, 6 pm, 9 pm, 12 am, and 3 am) for 21 days (18 July 2021 – 7 August 2021). All screenshots taken were of the same size and the exact same location on screen to allow for automated data conversion. A resultant map of the overall activity level of each waterfront segment was created and used to determine scope of the study area.

In the process of data collection, two main challenges were faced: one of maintaining data consistency, and one of converting the screenshots to usable data [figure 4.1].

The first issue was that the heat map displayed *relative* area popularity; where a larger map of Jeddah will show the most active areas in the city, and a zoomed in map of a district, will readjust the values to show the most active areas in the region. Any shift in the map's visible region will then affect the projected heat map values, giving an inconsistent reading. To counter this issue, a separate web browser tab was dedicated to monitor the Snap Map – framing the waterfront in the centre of the visible map. The hyperlink was saved for extra measure, so that the exact same visible map portion could be recreated if needed, to ensure no accidental shifts in display occurred by panning or zooming in/out. It was impossible to view the entire waterfront on the heatmap without also viewing nearby activity points, which included an aquarium, a popular restaurant, and a theme park. The rise and fall of the activity levels in these surrounding locations affected the waterfront heat map results. As the waterfront is usually filled with active users, the most active parts of the waterfront are often represented in red. However, if the theme park was particularly active, the most active portion of the waterfront would be yellow, as opposed to its usual red. As a solution – the resultant map of values was based on the sum, and not the average, of each area.

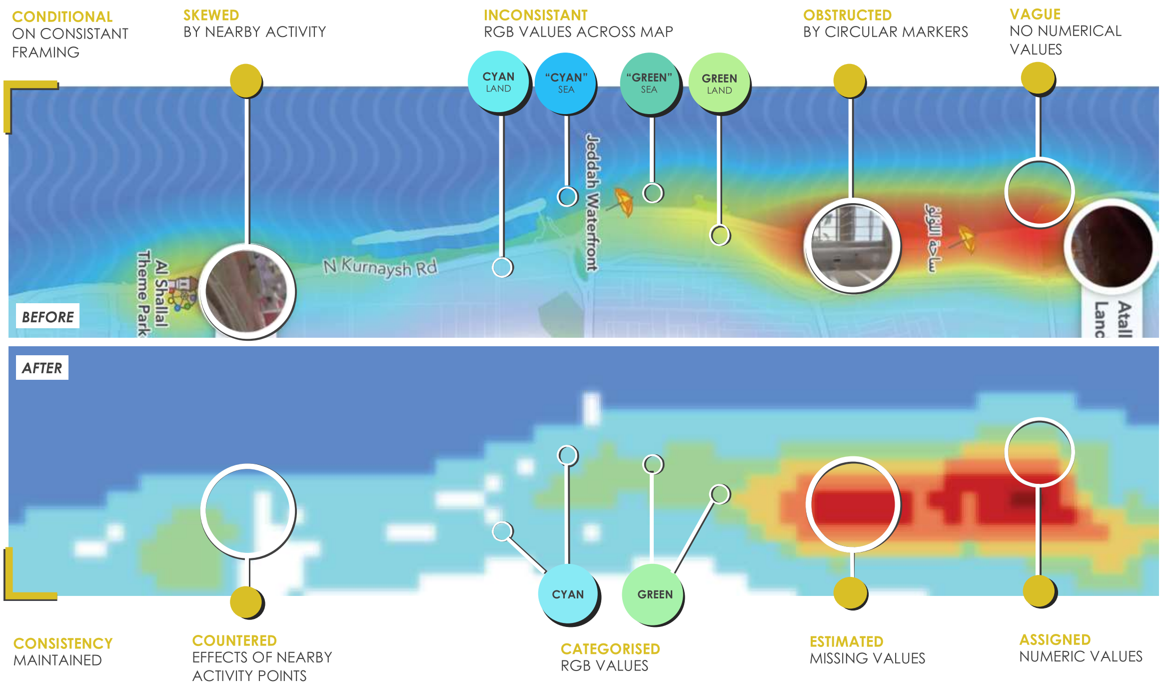


Figure 4.1. Limitations in current Snapchat heat map image and various ways they were addressed.

The second issue lied in Snapchat not providing actual numbers or statistical measures for public use; the only available data is then the visual heat map displayed. Additionally, the posts have a time limit of about 24 hours. It is then difficult to tell how popular the area is at a given time – as the posts can be indicative of current popularity, or of accumulated popularity over the past day. As this portion of the study is only interested in relative engagement level distribution, and not the actual numbers or timings, a visual heat map is sufficient to understand the popularity of each waterfront segment. However, these values are represented in image format – and not extractable data points. Additionally, the heat map values are intuitively clear – but computationally vague. This means that while the average user is able to determine what portions are have high activity levels (red) and what areas have low activity levels (cyan), the RGB values are hard to define as the heat map is translucent – showing both the activity levels and the colours found in the background map of Jeddah. 'Red' would therefore look different if the area being assessed was a district (beige background), a park (green background), or near a water feature (blue background). Another issue faced was the placement of obstructive large circular markers on popular destination spots. These markers are meant to show any posts and user activity within that defined project, with the marker inconsistently appearing and disappearing depending on number of posts. While most marked locations lie outside of the waterfront vicinity (restaurant, aquarium etc.), some areas of the waterfront (public beach) are also marked as popular locations and, when appearing, cover relevant parts of the heatmap. While these markers *can* be removed by inspecting the html code on the webpage – removing the markers also removes all posts within that marker – giving an inaccurate representation of the waterfront's activity level.

To address these problems a number of steps were followed. Initially, the quality of the heat map was reduced to a 70x45 pixel sized image. This reduction in quality greatly reduced the combination of colours displayed – where each colour category was visually, and computationally, easier to distinguish. A program was then created on Java using Eclipse [*version Mars.2*] to translate the heat map into usable data [figure 4.3].

Colour Sets Defined
First, a representative pixel was chosen to represent each heatmap colour category (maroon, red, orange, yellow, green, and cyan). A pixel to represent the sea (blue) was also selected. These categories all had one colour $\boldsymbol{c}$, defined by the colour of the pixel that best represent the colour set in question, with respective $\boldsymbol{c}_r$, $\boldsymbol{c}_g$, and $\boldsymbol{c}_b$ RGB values. A function was then created to compute all colours on the map that were 'similar' to the base pixel of each category. This function uses the equation below to create a set of similar colours $\boldsymbol{S}$, where a colour $\boldsymbol{x}$ is an element of $\boldsymbol{S}$, if the difference between the RGB values of $\boldsymbol{x}$ and that of the set representative $\boldsymbol{c}$, is smaller or equal to the user-defined tolerance level $\boldsymbol{t}$.

$$S = \{\, x \mid \sum_{i \in I} (c_i - x_i)^2{}^{1/2} \le t \} \quad I = \{r, g, b\}$$

As this equation tests the difference in colours against the variable $\boldsymbol{t}$, a tolerance level based on user input is allowed, either increasing or decreasing the number of colours within the similarity set. Different $\boldsymbol{t}$ values were tested for each colour to see which numbers best represented the visual colour range associated with each category [table 1].

Minimal Map Created
The next step was to get the RGB value of each pixel. The program would test to see if it fit into any of the predefined colour sets. If it passed the similarity test, the pixel was recoloured to the base colour of that set. If not, the pixel was recoloured to white. The resultant map was a heatmap, with areas missing where markers were located.

Missing Values Estimated
This step aimed to recreate the values hidden behind the circular markers. Initially, each heat map colour was given a value [table 1]. For each white pixel, an average value of its surrounding colours was calculated and assigned [figure 4.2]. A minimum threshold of non-white pixels was defined by the user to control the number of iterations required. Where a larger minimum threshold would provide a more accurate representation, as it ensures that the pixel wasn't coloured based on a single non-white value – for instance, a smaller number minimum threshold provides a faster method requiring less iterations. For this case, a minimum threshold of 3 was found to be ideal – with 2 iterations required to estimate the missing values.

Borders Recreated
As the bordering colours (green/cyan) hold a smaller value than the central colours (maroon/red), continuing the above estimation process led to inaccurate results, where the central colours continued to bleed out of their predefined borders. To counter that problem, a second function was created to give preference to border colours. Here, after two iterations of the previous process ensured that the missing values were recreated, the code was adjusted to allow for easier border creation. For each remaining white pixel with non-white pixels above the minimum threshold, the surrounding pixels are checked. If the set included both a border colour (cyan) and a pre-border colour (green), the pixel was recoloured as a border (cyan). Where the reason the code included a pre-border colour was to ensure the pixel was within the border boundaries of the heatmap, and not an outside pixel neighbouring the border. If the pixel does not include both these colours, the estimation process continued as before. This step iterated automatically until no further changes could be made to the map.

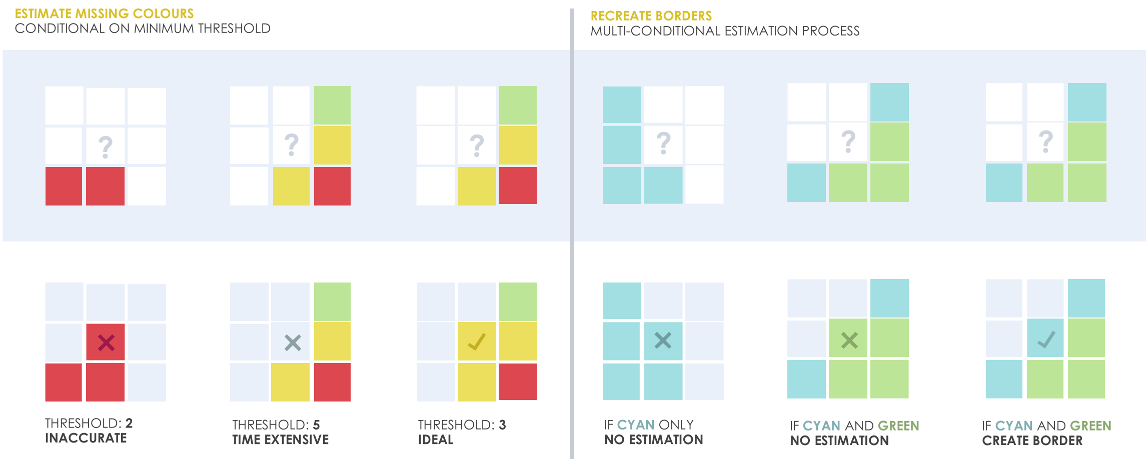


Figure 4.2. Visual elaboration on the technique used on the Snapchat heat map using various thresholds to estimate missing pixels and recreate heat map borders.

Data Map Exported

Finally, as all of the pixels were coloured, and all of the colours had preassigned values – an excel sheet with value of each pixel (based on colour) was exported. Where columns represented the different times/days of the screenshot and rows represented each pixel with its estimated coordinates on the respective QGIS map. An additional column representing the total sum of each pixel was also added, and later visualised on QGIS.

4.2.2. Measuring Peak Hours: Google Traffic Live

To engage in pedestrian observation studies on a micro level, a set time had to be defined for site visits. Visiting the site in non-peak hours would be misrepresentative of the typical activity patterns of the waterfront. As such the primary reason for vehicular traffic monitoring was to estimate *peak visiting hours* of the Northern Corniche Waterfront. A secondary reason of the study would be to understand the *distribution of the traffic* along the 4.5 km waterfront stretch. The results would then be used to gain a better macro understanding of visitor behaviour – as Snapchat metrics alone are not representative of the entire population. Live Google Traffic data was monitored over a period of 21 days (18 July 2021 – 7 August 2021), with screenshots taken every 3 hours (6 am, 9 am, 12 pm, 3 pm, 6 pm, 9 pm, 12 am, and 3 am) in an effort to study traffic flow over time. Screenshots were taken consistently with the exact same placement and dimensions.

To extract usable data from the google traffic screenshots, a code was written on Java (Eclipse)[version Mars.2] that achieved the following [figure 4.4]:

Data Points Allocated

Firstly, a set of predefined points on QGIS were allocated along each traffic lane, spaced around 50 meters apart. Each point was then manually assigned a pixel on the screenshot that represented the point location on the image. A set of all the representative pixels was then created – with each pixel having a corresponding QGIS traffic data point.

Colour Sets Assigned

The traffic data has 4 colour categories that represent the different traffic values (maroon, red, orange, and green) [table 2]. As there were no distortions in map colour, defining the different colour sets based on the extracted RGB values was a straightforward process. For each pixel in the predefined list, the RGB colour was identified, and the corresponding value assigned [table 2].

Data Map Exported

A data map was then created on Excel, with each row representing a pixel, along with its QGIS coordinates, and each column a day/time. The values were then averaged based on several different categories (ex: weekdays, weekends, timings) and imported into QGIS for visualization.

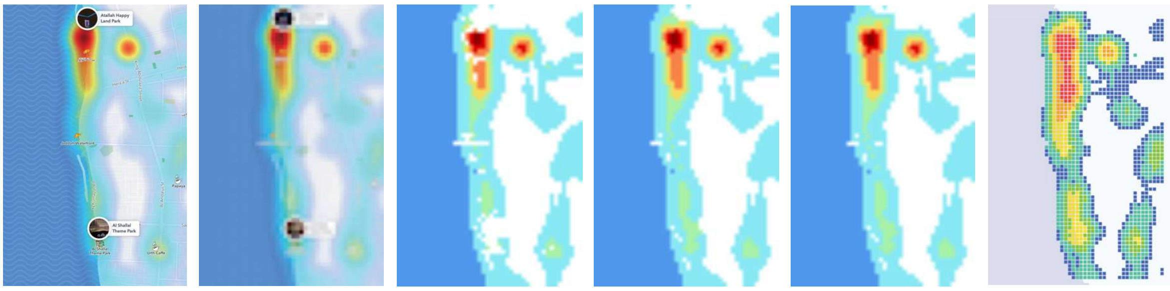


Figure 4.3. [above] Snapchat heatmap recreation process from screenshot to usable GIS data.

Figure 4.4. [right] Google Traffic data conversion process from screenshot to usable GIS data.

Table 1. [right] showing colour sets assigned with respective pixel RGB values, tolerance levels, and a sample of the various shades associated with each set.

Table 2. [below] showing colour sets defined by Google, along with their respective RGB values used to collect traffic data from the images collected.

SNAPCHAT HEAT MAP COLOUR CHART

	VALUE	*PIXEL*	*RGB*	*TOLERANCE*	SIMILAR COLOURS
MAROON	6		143 0 0	65	
RED	5		227 0 0	65	
			245 38 47	45	
ORANGE	4		216 105 61	20	
			236 107 62	20	
			244 150 76	30	
YELLOW	3		239 234 103	35	
			172 201 105	20	
GREEN	2		191 245 161	30	
			109 208 180	25	
CYAN	1		128 243 236	20	
			52 194 241	15	
BLUE	0		80 152 237	15	

GOOGLE TRAFFIC COLOUR CHART

	VALUE	*PIXEL*	*RGB*
MAROON	4		157 19 19
RED	3		231 0 0
ORANGE	2		241 125 2
GREEN	1		132 203 81

4.3. MESO: MODELLING PEDESTRIAN SPACES

To explore the characteristics of the waterfront's spatial network, the theory of Space Syntax has a number of key indicators signifying the positioning and value of a space within the network. These measures will be adapted in an aim to understand and predict *movement patterns* and *visibility*. However, as opposed the typical street networks and building spaces, waterfronts pose different spatial challenges – particularly when it comes to modelling these spatial metrics. This difficulty appears in both movement pattern modelling and visibility studies.

4.3.1. Modelling Routes of Movement

To understand movement patterns in linear spaces (such as street networks or pathways), Hiller creates what he calls an *axial map* (Hillier & Hanson, 1984). An axial map is a collection of the minimal number of straight lines needed to cover and connect an entire network of spaces. This map is then used to generate metrics including measures of connectivity, integration and choice (Hillier & Hanson, 1984). Connectivity is the number of other spaces a space is directly connected to; integration is the average topological depth a space has in relation to all other spaces in the network; and choice is the number of paths that are required to go through the given space to complete a journey (Peponis, 1993). As such, connectivity measures the potential *options* one has, integration measures the potential a space has in terms of *to-movement* (i.e. the space is the destination), and choice measures the potential a space has in terms of *through-movement* (i.e. the space is a transitional point). This axial map can be used as-is, or it could then be transformed into a *segment map* (Turner, 2000) which subdivides each axial line at any point of intersection it may have, creating a more fine-tuned model that integrates *segment length* (distance) and *segment angle* (angular change required). The angular segment map has been shown to highly correlate to pedestrian movement by (Hiller and Ilda, 2005; Turner, 2005) and will therefore be used as the standard metric for movement in this research.

4.3.1.1. Current Model Limitations

Modelling routes for movement in a large landscaped space is challenging. These spaces are often less represented in Space Syntax literature, thus there is no systematic way to model landscaped and open spaces. Issues faced while modelling axial or segment maps onto landscaped spaces often fall under two categories. The first being a problem of unclear boundaries. Zhai and Baran (2013) discuss in their paper Application of Space Syntax Theory in Study of Urban Parks and Walking the difficulties in modelling landscaped spaces. Accessible spaces in landscaped areas such as waterfronts or parks are often a combination of predesigned paths and grassy areas. Pedestrians are not restricted to defined footpaths in their movement across the park. One such difficulty then lies in differentiating between modelling pathways and modelling their neighbouring grassy lawns, where both spaces are technically accessible to pedestrians. To understand the paths of movement as designed, Zhai and Baran suggest modelling only the pathways. Exceptions are made when pathways naturally lead to activity points (playgrounds, landscape) where the lines are extended to cover the activity space. Other issues realting to unclear boundaries are less clearly addressed, such as that of 'fuzzy boundaries', where vegetation appears to create a certain boundary around an area – but the option to walk through it remains. Here, the decision of whether to include this informal access point as a formal path within the model depends on the question the research aims to answer. The second problem is one of overrepresentation. Parks and green spaces often have curved and winding paths, more so in smaller gardens than in larger parks. These paths do not necessary create a fragmented line of sight. To preserve the concept of an axial model Zhai et al. adopt a stroke-based representation of the existing network, where excessive curved lines are represented with a single 'stroke', or axial line (2018). Figueiredo and Amorim also use a single axial line to represent unnecessary fragmented paths (2005). Where unnecessarily fragmented lines are those that are fragmented despite having a continuous line of sight (ex: not surrounded by heavy vegetation where directional turns are expected to affect pedestrian vision).

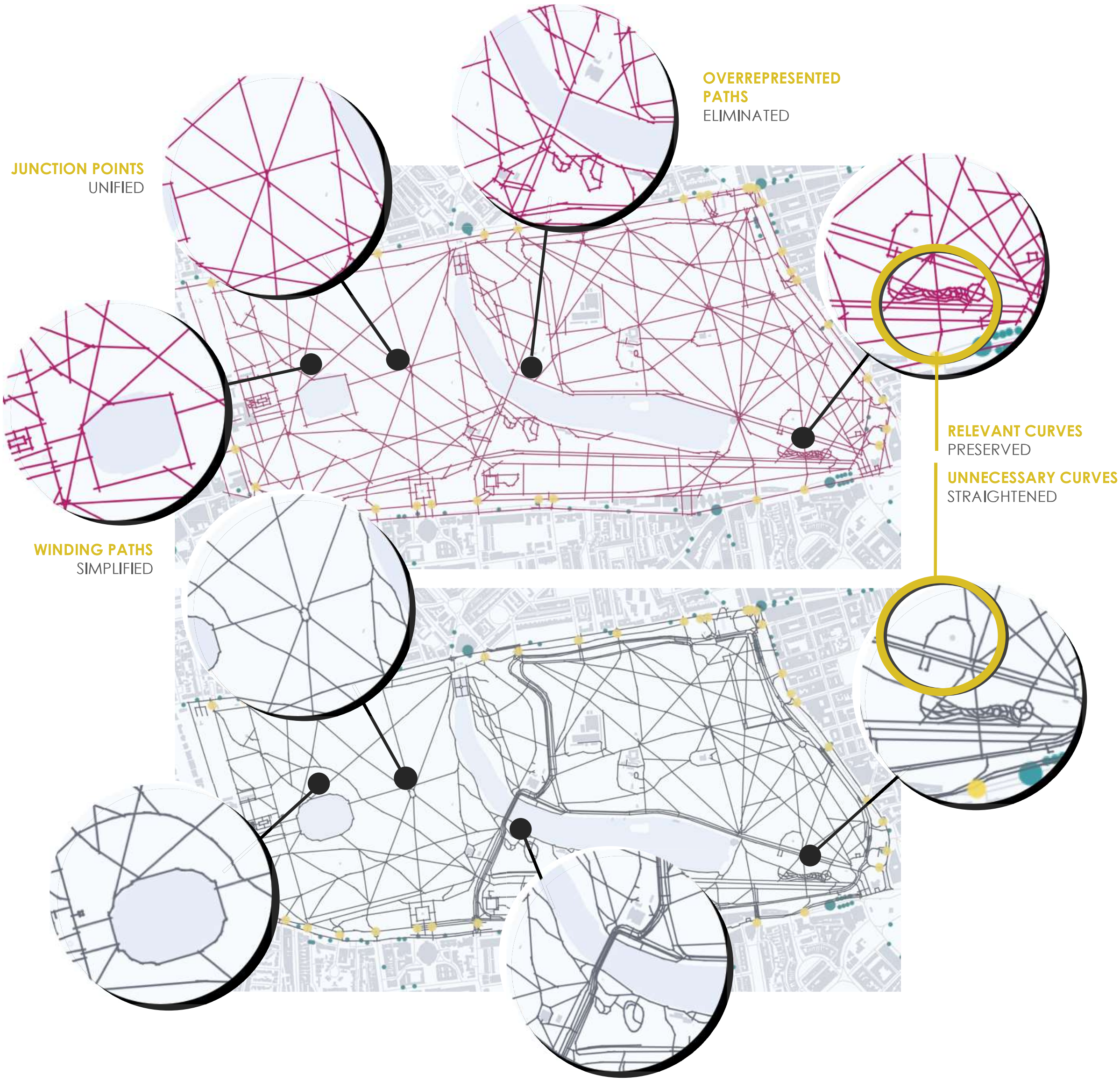


Figure 4.5. Sample image showing how axial lines are modified to create better representation of landscaped spaces. Image depicts Hyde Park before [below] and after [above] modifications.

4.3.1.2. A Segment Model for Landscaped Spaces

To offset the problems of overrepresented lines and unclear boundaries, a segment map of the waterfront was created using the following steps. First, a boundary around all designed pathways and paved spaces was created on QGIS [version 3.10], including any objects or buildings that may block pedestrian routes of movement. This map was then imported into DepthMap [version 0.7.0] and an axial map was automatically generated. Following the generation of the axial map, the lines were reimported back into QGIS and modifications based on the relevant literature were made. Figure 4.5 summarises the general points referenced in the literature on modelling landscapes spaces. While modelling the waterfront, special attention was given to extending pathways onto activity zones, such as playgrounds, where relevant; incorporating smaller trails on grassy fields, where these trails led to paved seating areas; eliminating multiple lines that represented a single path, such as bike lanes alongside pedestrian walkways; eliminating multiple lines that represented a route of movement, mostly in open plazas; and straightening any unnecessary curves. The adjusted map was once again imported into DepthMap as an axial map, with the software then converting it into a representative segment map ready for analysis.

4.3.2. Modelling Visibility

As a concept, visibility is less indicative of movement patterns and more reflective of the visitor experience. Globally, visibility can then be assessed from each grid point to every other grid point – this is known as VGA, or Visibility Graph Analysis (Turner, 2001). To calculate metrics of visibility, Turner formulated a grid like structure to uniformly represent the different points in a pre-defined space (2001). VGA measures correlate to the global visibility of the space. The visual connectivity measure, for instance, gives an objective indicator to how visible the space (or grid point) is when compared to the remaining spaces in the grid, irrespective of any individuals' lines of sight. It paints a picture of what areas are easily *seen* (controllability), what spaces hold the largest *vantage points* (control), and how the visibility shifts over the neighbouring spaces (Hacar, O., Gulgen, F. & Bilgi, S., 2020). Visibility can also be measured locally. Measuring visibility form a single vantage point is the Isovist. Isovists are fields of vision from the perspective of an individual and have been developed and modelled by Benedikt (1979). Field of view, depth, area, and occlusivity are all examples of measures used to study the field of vision at any given point. These measures often correlate with the perceived publicness of a space, or the sense of safety one has in a particular place (Zook, 2017; Fotios, Unwin & Farrall, 2015). Traditionally, visibility measures have been applied to uniform spaces with few environmental variations.

4.3.2.1. Current Model Limitations

Creating a representative visibility graph of the waterfront is complicated, requiring special attention to issues faced in modelling all outdoor spaces, landscaped spaces, and waterfronts.

The element of temporality plays a large role in visibility of outdoor spaces – with the waterfront being open to pedestrians at all times of the day and night. Navigating a space generally relies on a sense of spatial cognition. This sense highly depends on visual input, a factor that is affected by lighting at night (Dwimirnani et al., 2017). Areas that are ill-lit are linked to higher crime rates and feelings of unease negatively affecting pedestrian movement (Fotios et al., 2015). Whereas lighting is not a factor during the day, pedestrian movement at night requires a different model of visibility to accurately reflect cognitive choices.

Additionally, landscaped spaces are typically more difficult to model in fine-tuned visibility graphs. As opposed to street networks where the obstacles are typically built structures or walls, landscape often introduces vegetative elements that require closer inspection to determine their relative obtrusiveness. In non-typical environments, changes have been made to the model to allow for more flexibility in data input. Araguez and Psarra (2017) for example, adapted the VGA model to work with changing topography, and Zhai et al. adapted their model by removing inaccessible spaces from the overall VGA graph of a landscaped urban park (2018).

Waterfronts in particular have the added element of the sea-view as an integral part of the visibility model. This poses a number of issues including two-way vision versus one-way vision, where the pedestrians near the waterfront edge can see more of the sea, but they are not seen in return. Questions on sea-view assigned weight, i.e. *how much* of the sea should be included in the visibility model? Finally, as environmental conditions vary in outdoor spaces throughout the day, the shift from day light to streetlights could greatly affect the visibility of the waterfront areas. While some spaces might be upgraded from invisible to highly visible, other spaces might see a downfall in visibility due to lack of adequate lighting. The remainder of this section then aims to address the limitations found while trying to model the waterfront using the current available tools.

To address the challenges faced with attempting to model the Jeddah Waterfront, summarised in table 3, a new model was created on Java (Eclipse)[version Mars.2] (full code available in appendix). The following sections expand upon each suggested solution, translating each step into a base component in the new visibility model created for assessing large outdoor landscaped spaces, both during the day and night. This study utilises all components to analyse the Jeddah Waterfront (with the exception of tracking object visibility, later omitted due to time restrictions, which could be expanded upon in future research). However, integrating attractors, night time visibility, and isovist model calibration is optional, and can be included/omitted by future researchers depending on study needs.

4.3.2.2. Flexible Grid Systems for Large Scale Models

The flexible grid system was designed to addresses multiple limitations with the current square grid model; these limitations relate to grid-size, uniformity, complex obstacles and inaccessible areas, particularly in relation to outdoor landscaped spaces.

To accurately portray landscaped spaces with the current model, a finer grid is required. The waterfront is a massive public landscaped space with heavy vegetation. Creating a representative visibility graph using current tools would require a fine-scaled grid to account for all vegetation (kiosks, sculptures, shrubs, tree trunks, etc.). As DepthMap represents each grid point as a square, an accurate representation needs grid points to be as small as the smallest relevant obstacle in the model. Palm tree trunks, for example, are around 50 cm wide on average (Jahromi et al., 2007). With a 2-meter grid spacing system on DepthMap, the palm tree would either go undetected as an obstacle, or the entire square encompassing the tree would count as an obstacle, distorting the data either way.

As the current model only allows grids that are uniform in nature, creating a fine-scaled grid is especially challenging if only parts of the site require a more detailed representation, with a large portion of the site sufficiently represented by a larger scale grid. A park with a few narrow passageways between heavy vegetation would require the entire site to be modelled using a fine grid – or risk having these narrow passageways left out of the graph. Due to the uniform nature of the grid – one would have to choose between computation complexity or representational errors. Of course, the option to link the grid points manually remains. However, this method is time consuming for larger sites, with the manual nature of the process leaving some spaces susceptible to being overlooked.

Furthermore, the current square grid system requires smaller squares to represent non-uniform objects. Landscaped spaces and shrubs often form complex shapes that are difficult to accurately model with a non-fine-scaled grid.

Finally, as the grid system only recognizes two categories for each grid point: *accessible area* or *obstacle* – inaccessible regions cannot be omitted without affecting the remaining graph structure. A simple example of an inaccessible region would be the sea between the waterfront edge and a neighbouring pier. Modelling the sea as an obstacle would imply that people on one side have no visual access to those on the other side. On the other hand, modelling the sea as an accessible area would distort the data. Firstly, it would require unnecessary computational power while assessing visibility levels of sea points to other sea points. Secondly, it would increase the level of visibility of each nearby grid point, as each point would now (inaccurately) be labelled as 'seen' by all the inaccessible points at sea [figure 4.8].

These challenges are addressed in the new model by creating a point-line system to replace the square-grid system. Here, grid points are represented by a collection of points, while each object is represented by a collection of lines.

Firstly, a uniform grid of points is created within the boundaries of the site, along with a line representation of all obstacles. Secondly, all points in inaccessible areas are removed (including any points found on obstacles), leaving only accessible and visible grid points. Thirdly, any required adjustments to individual grid points are made. By allowing flexibility as opposed to rigid grid uniformity, additional points can be added where required to prevent discontinuity. While the model aims to stay as close as possible to a uniform grid for an even representation, the choice to add or remove specific points from the grid allows for a larger-spaced model that is less computationally expensive. Fourthly, a standard radius is defined by the user; in this case the radius was chosen to be 150 m based on on-site visibility, tested at several points along the waterfront, as there was no set standard in the literature on pedestrian visibility. Fifthly, a connectivity graph is created [figure 4.6].

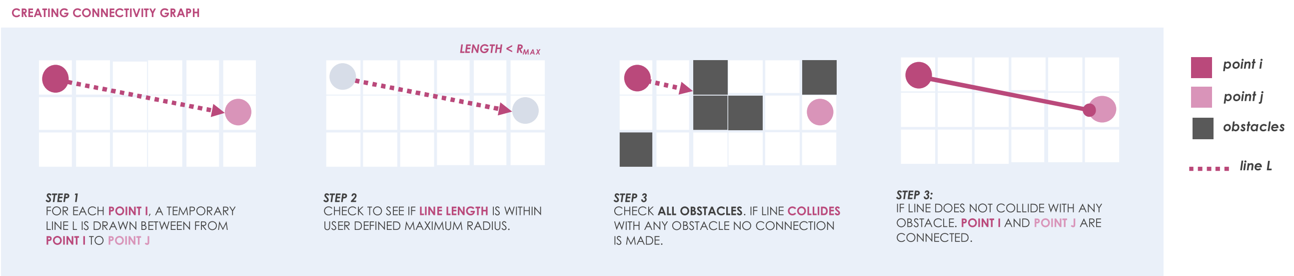


Figure 4.6. Illustration of how the point-line grid system works.

As this system has a separate form of representation for obstacles and grid points, a fine grid is no longer required to accommodate for smaller obstacles, non-uniform obstacle outlines and narrow passageways. Inaccessible spaces are also easily modelled with no added effects on the current graph metrics. As there is no formal boundary, a grid point i has visual access to grid point j as long as it satisfies two conditions: one, there is no obstacle interrupting the line of sight, and two, the length of the line is within the specified radius bounds.

As a result, a representative graph of the Jeddah Waterfront using the new model allows for grid points to be spaced 5 meters apart (around 5000 total nodes), allowing for a less computationally expensive model with more flexibility. In comparison, to get a representative graph on DepthMap, the grid points had to be spaced at 1 meter apart (graph estimated to contain around 25,000 nodes). As the computational expense of creating a visibility graph is at the very least $O(n^2)$, the new model should theoretically take no more than 4% of the total time required to compute the first model.

Additional changes made to the system included the creation of *zones* to optimize computing speed. To optimize the model for large scale projects – an optional zoning system was created. This system would divide the current grid into a number of zones based on user input, where each point, and obstacle, would be assigned to their respective zone(s). The graph creation process would then be adjusted accordingly [figure 4.7].

This adjustment aims to reduce the computational time by omitting unnecessary collision checks with obstacles that are far away from the points in question. The creation of zones improved computational speed up to a certain point – after which the increase in number of zones led to an increase in computational time. For this study, the optimal number of zones was 9, but case-by-case tests are recommended to identify the ideal number of zones for each project.

With further optimization to the code created, the actual time taken to create a visual integration model (radius=n) of the waterfront was 8 minutes, compared to 23 hours on DepthMap (1% of the total time). While visual integration values were computed to test the new model against the current model, the remainder of this research will continue to work with visual connectivity exclusively – with visual integration metrics potentially explored in future research.

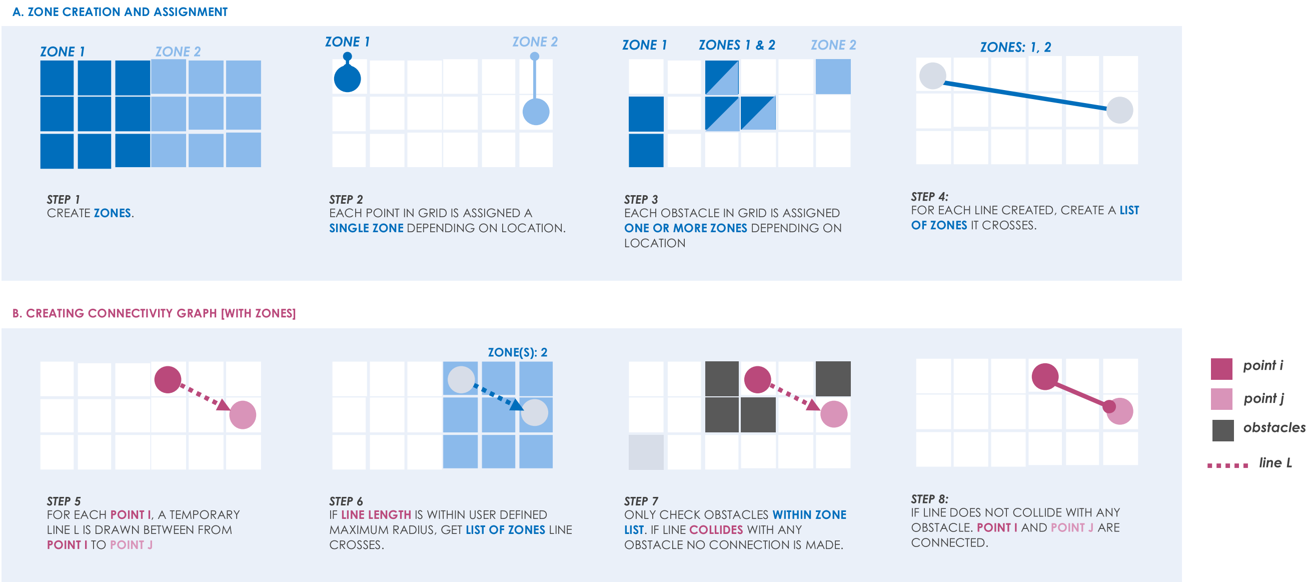


Figure 4.7. Illustration of how the point-line grid system is adapted to work with zones.

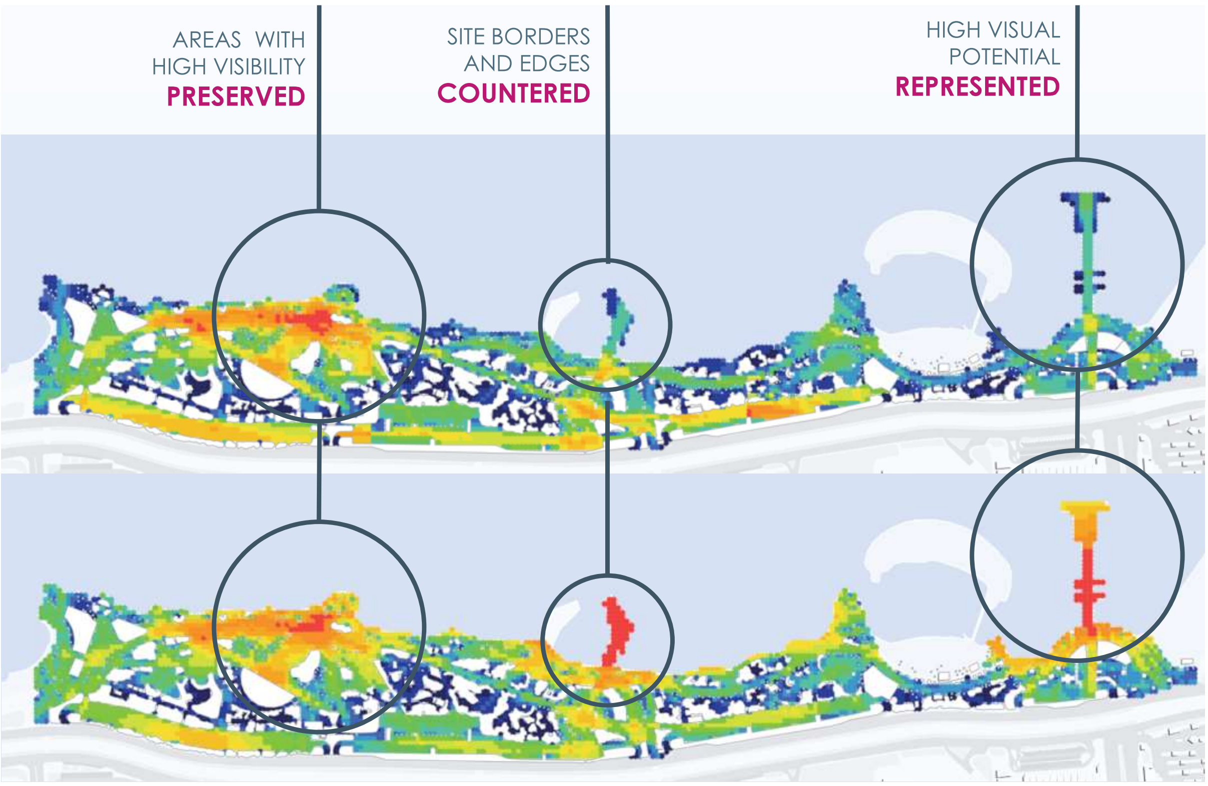


Figure 4.8. A comparison of two different visibility maps of the waterfront illustrating the current model limitations when representing waterfront spaces. The figure shows a visual connectivity map using classic DepthMap tools [above] and the modified code [below].

Modelling Challenges	Existing Fix	Current Model Limitations	Suggested Solution
Smaller Obstacles	Use a fine-scale grid of 1 meter or less	Computationally expensive, estimates and not exact results if gird isn't small enough; inconvenient if only portions of the site require a smaller grid to be accurately represented	Point-Line Grid System
Irregular Obstacles			
Narrow Passageways			
Large Inaccessible Spaces	Include spaces into model	Data distortion if spaces are integrated into model, and inaccurate portrayal if they are represented as obstacles instead	
*Obstacle Visibility**	Create separate graphs removing one obstacle at a time	To understand effect of obstacle removal, two separate models need to be created. No current model offers immediate access to visibility metrics of different obstacles and objects.	Tracking Object Visibility
One-Way Vision	N/A	No flexibility in current model to label visual connection as one-way; difficult to model spaces where visibility is unbalanced (e.g. topography, visual vistas, etc.)	Directed Graph
Night Time Visibility	N/A	No existing model that incorporates area brightness and pedestrian visual perception	
Global Connectors *(lines longer than x meters)*	Subtract local values from global values	No standardised model allowing for radial lengths above a certain threshold	Multi-Radial Length Model
Unobstructed Vistas	Include vistas into model	Visibility graph skewed towards vista, and model would be computationally expensive	Weighted Nodes
Attractors	N/A	No current model incorporating concept of visual attractors (e.g. exhibits) where the closer one is the higher the visual value	

**Obstacle Visibility component was created but will remain unused throughout this research due to time constraints.*

Table 3. Summary of challenges faced while attempting to model the waterfront using the current available tools, and suggested approaches for each limitation.

4.3.2.3. Tracking Object Visibility: The Hit System

Another addition to the current system is the option of tracking obstacle hits. There are two main objectives to this metric. The first objective is to understand how many lines of sight each obstacle is blocking. This could be used as a pre-design assessment to easily visualise what objects would have the largest impact on area visibility if removed. The second objective is to measure how seen each object is. This is especially helpful in assessing visibility of kiosks or exhibits, where the number of hits would correlate to the number of times the object was seen from the surrounding grid points.

To track the number of hits, a counter was added to each object – with the number of hits increasing by one point each time the object collides with a potential line of sight. To adopt this method, a slightly more computationally expensive approach needed to be taken. As obstacle hit tracking is not the main goal of the new system, currently, the code asks for *any* obstacle x found blocking the line of sight L between a point i and a point j to stop the process, labelling points i and j as disconnected in an optimal time [figure 4.9]. In this case, obstacle x is not necessarily visible to points i and j. Obstacle x might lie in between two other obstacles y and z – but technically still colliding with the line of sight L, in this scenario – whichever obstacle is tested first gets an increase in the number of hits, regardless of accuracy. To accurately measure the number of times an object was hit, all objects blocking the line of sight L first have to identified (no stopping conditions to optimize the code). The closest object to point i (obstacle y, in this example) is then assigned as the object that would get an increase in hit count.

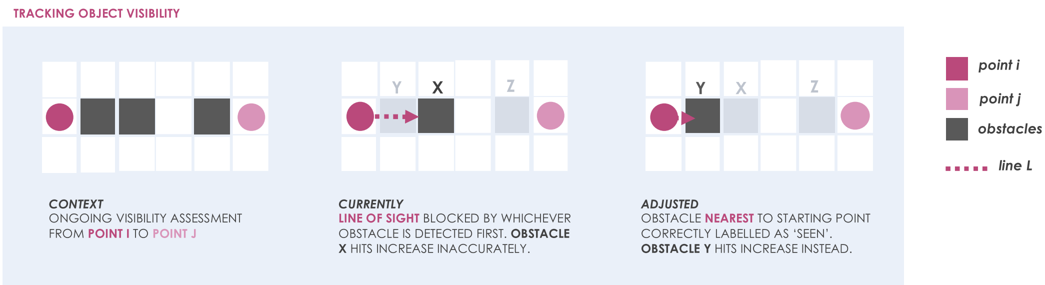


Figure 4.9. Illustration of modifications required to implement the object hit system.

As this study is not directly concerned with object visibility or increasing visibility values in certain spaces, this component will not be further explored in this study and might instead be expanded upon in future research.

4.3.2.4. Working with Directed Graphs: One Way VS Two Way Vision

Visibility does not always imply two-way vision. While in most cases, one grid point having visual access to another implies the reverse, there are cases where one-way visibility is the norm. An example relevant to this study is the sea. A viewing dock near the waterfront edge has a high visual range, but as the sea is inaccessible, the added visibility does not affect how *visible* the viewing dock is. To model the visibility of waterfront, and night-time visibility in the following section, the introduction of a directed graph is required. As the new model is represented as an adjacency matrix – implementing a directed graph is straightforward; where each row of the matrix would represent how much a point can see (*visual potential*), and each column how seen a point is (*visual vulnerability*) [figure 4.10].

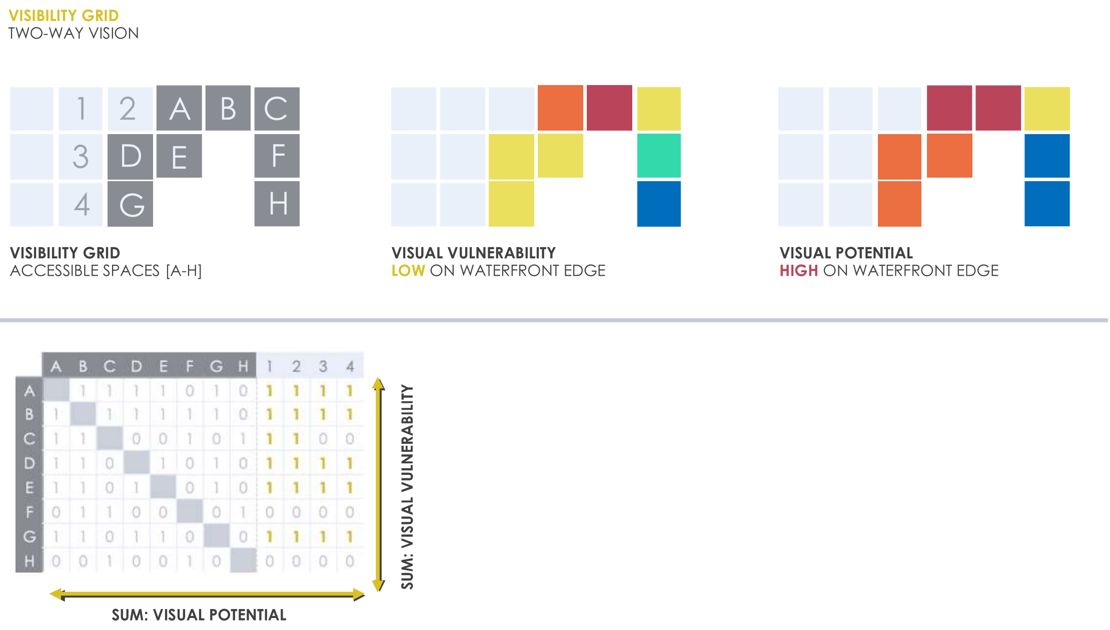


Figure 4.10. Illustration of a directed visibility graph comparing between a points visual potential and visual vulnerability.

4.3.2.5. Modelling Night-Time Visibility: Introducing Multi-Radial Length Models

As this is a custom model – there is added flexibility when it comes to modelling visibility graphs with different radii. This flexibility can be used to model local visibility models, a visual model of global connectors, or it can be used to model a more complex system: the night-time visibility model.

While the current model allows for local visibility models with radii under a user-defined value, there is no option to create a visibility model showing only points with radii larger than a given value. Whereas *local visibility* graphs are valued for showing points with high levels of direct visibility – where points in large open spaces or squares seem to score the highest, a graph showing the *visual connectors* is important for highlighting points with long visual ranges – here points along long vistas with unobstructed views score the highest.

To create the more complex night-time model, the luminosity of the surrounding area first needed to be assessed; thus, a set of standards was created [table 4]. Brightness points were then assigned to each grid point, before a multi-radial method was introduced in the creation of the connectivity graph.

Since lighting at night fluctuates greatly from one area to another, a systemic method of measurement was required to assess the luminosity of the waterfront spaces before introducing it into the new model.

Initially, an application called *Light Meter* was used to measure the area luminosity. This application, along with other phone-based applications, uses the phone's front and rear cameras to assess exterior lighting conditions. During the day, a varied range of readings was provided, depending on how shaded/well-lit the region was. At night, however, the application was unable to detect any exterior lighting – unless specifically pointed towards the source of light. A number of alternative applications were tested, including *LUX Light Meter*, *Photo Light & Exposure Meter*, LuxSeeker, and *Lumu Light Meter* with the same results. As this study is not interested in luminance of the specific lighting structures, but the general lighting of the areas (which includes luminance, number of lighting structures, height, etc.) the information gathered by the applications was insufficient and had to be omitted.

Instead, a set of standards was defined and used to categorise the waterfront spaces into one of four categories: *well-lit*, *dim*, *dark*, and *indiscernible*. After surveying the site, and understanding the existing variation in lighting, the criteria in table 4 was created for each lighting category, with a second site visit later conducted to classify each portion of the waterfront.

To integrate the luminosity scale into the new model, each grid point was assigned a value from 0-1 based on the criteria set in table 4. This value would represent what percentage of the maximum radius each lighting category would be assigned. As lighting at night differs greatly from natural daylight, a maximum radius of 100 m was defined as the maximum input radius for the model. A luminosity value of 0.4 would then indicate that a maximum visibility radius of 40 m (0.4 x 100) would be assigned to that specific brightness category. The base for establishing luminosity values for each brightness set was established by on-site testing, where visual perception was examined over many areas to determine the maximum visible distance for each lighting category.

	WELL-LIT	**DIM**	**DARK**	**INDISCERNABLE**
Light Fixture Conditions	*Satisfies* ***4 Conditions*** *Tall* *Bright* *Evenly Spaced* *Unobstructed*	*Satisfies* ***3 Conditions*** *Tall* *Bright* *Evenly Spaced* *Unobstructed*	*Satisfies* ***0-2 Conditions*** *Tall* *Bright* *Evenly Spaced* *Unobstructed*	*contains disruptive lighting that* ***obstructs vision***
*Maximum Visibility**	60 - 80 m	~ 40 m	~ 20 m	<10 m
Maximum Radius		*100 m*		
Luminosity	*0.8*	*0.4*	*0.2*	*0.1*

**Maximum visual range allowing pedestrians to detect one another and mid-range to smaller-range objects. Based on site visit measurements.*

Table 4. Lighting standards defined and used to categorise waterfront spaces, along with the respective luminosity value each category is assigned. Luminosity values range from 0-1 and represent what percent of the maximum radius a person in that area can see.

The connectivity graph was then adapted to work around the luminosity of each grid point. Conceptually, this implies that the darker the area is – the less likely it is to be seen from afar, and the brighter an area is, the more visible it is to those from a distance. The luminosity of the starting point is then irrelevant, what matters visually is the luminosity of the end point. The graph creation process is then adapted to accommodate the varying radial lengths [figure 4.11].

The graph created is a directed graph. The contrast between the dark areas and lit areas change the visual balance, so that one point having visual access to the other does not necessarily imply the opposite. The result is then two separate connectivity graphs, one representing how seen each area is (*visual vulnerability*), and one representing how much each grid point can see (*visual potential*).

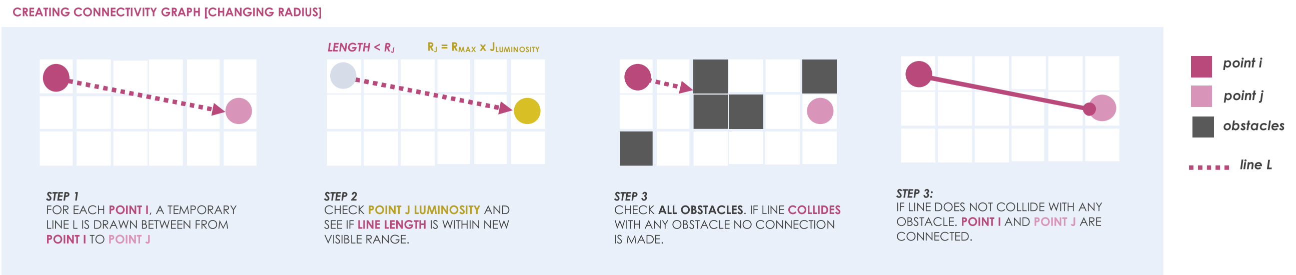


Figure 4.11. Illustration of how the point-line grid system works with area luminosity and varying radial restrictions.

4.3.2.6. Visual Vistas as Weighted Nodes: Working with Attractors

The sea, in the case of the waterfront, is assumed to be the largest and most influential attractor – as it is the focal point around which the waterfront is constructed. It is then important to treat it as such. While the introduction of the one-way visibility system is a step in the right direction, it still does not capture the effect of the sea view for two reasons.

One, the sea is an unobstructed landscape – the question of how much of the sea to incorporate into the model is a valid one. Too many grid points would skew the entire visibility graph towards only the waterfront edges, as well as make the model computationally expensive with the introduction of all the sea nodes. Too little grid points would be misrepresentative – treating the sea as an extended area of paved land.

Two, the sea is clearly an attraction point, where the observation decks are coveted for the sea view. The *distance* to the sea then matters. Regardless of how many nodes are technically visible from afar, the closer one is the sea the clearer the view, and the further away one is the less of a focal point it is.

To treat the sea as an attractor point, three additions were made to the model. Firstly, a single row of grid points representing the sea were created, bordering the waterfront edge. These grid points were labelled as *edges*, where they are only seen, and never see. Secondly, these edges were assigned adjustable weights – working with the concept of weighted nodes. The weight, in this case, equals the number of nodes this one edge would represent. So, given a weight of $\boldsymbol{w} = 10$, a point seeing an edge point would be equivalent to the point seeing 10 individual points. This allows for an easily adjustable weight, at no additional computational expense. Thirdly, to address the issue of distance – the distance between each grid point and edge point would be factored into the calculation of the final weight. Each point $\boldsymbol{i}$ would then get a custom weight, $\boldsymbol{w_i}$, depending the maximum weight allocated by the user, $\boldsymbol{w_{max}}$, and on its distance from the edge node, $\boldsymbol{d_i}$, when compared to the maximum distance between a grid point and an edge node, $\boldsymbol{d_{max}}$.

$$w_i = w_{max} \cdot \left(1 - \frac{d_i}{d_{max}}\right)$$

The smaller the distance between the grid point and the waterfront edge, the more visual connectivity points a grid point received, and vice versa – where $\boldsymbol{w_{max}}$ is the maximum possible points received, and zero the minimum. As the connectivity of each point is calculated after the creation of the adjacency matrix – no adjustment to the core code was required. The additional weight was calculated afterwards, where if point $\boldsymbol{i}$ was marked as connected to a point $\boldsymbol{j}$, one point would be added to its overall connectivity score. But if point $\boldsymbol{j}$ happened to be an edge node, $\boldsymbol{w_i}$ would be added to the overall score instead.

The final visibility graph should then portray the sea as an attractor, with an option to input various weights that can be later used to calibrate the effect of the sea with the help of pedestrian observation methods.

4.4. MICRO: MEASURING PUBLIC ENGAGEMENT

The success of the design layout could be tracked and measured using visitor engagement as a metric. Engagement is defined as the active decision to pause and observe, or actively interact with, a design element or space. It is distinguishable from contact, where a design element or space is simply within the field of vision of a visitor with no little to no active engagement (Peponis et al., 2004). It then follows that a detailed observation of waterfront visitors is essential to assess how they are engaging with specific waterfront elements and spaces. These observation methods should provide a detailed account of what the pedestrians are visually engaged with, which, in return, requires a broader context of what the pedestrians are in visual contact with.

4.4.1. The Snapshot Method: Tracking Pedestrian Patterns

Pedestrian observation techniques can be categorised into methods to understand *movement*, such as pedestrian traces and directional splits, methods to understand *traffic and flow*, such as gate counts, and methods to understand *activity*, such as snapshots. As this segment of the research is interested in pedestrian engagement patterns, the snapshot method will be adapted to track and monitor pedestrian allocation and activity patterns. Snapshots are often used to track pedestrian *routes of movement* as well as the *activities* they are engaged in. Snapshots also provide a static, but larger, picture, of what a group of individuals has *visual access* to at a given point in time. In their paper on Seeing and Being Seen Inside a Museum and a Department Store (2015), Kwon and Sailer used snapshots, considering the individuals' directional orientation to study the visual range, or isovist, of each visitor – allowing the researchers more insight on who was seen by whom at any given point. Snapshots therefore also allow the introduction of static isovists – where isovists based on the location of the visitors can provide insight on what parts of the waterfront are seen and interacted with, as opposed to other neglected parts; and whether some activities are more obscured from the public view while others are publicly 'seen'.

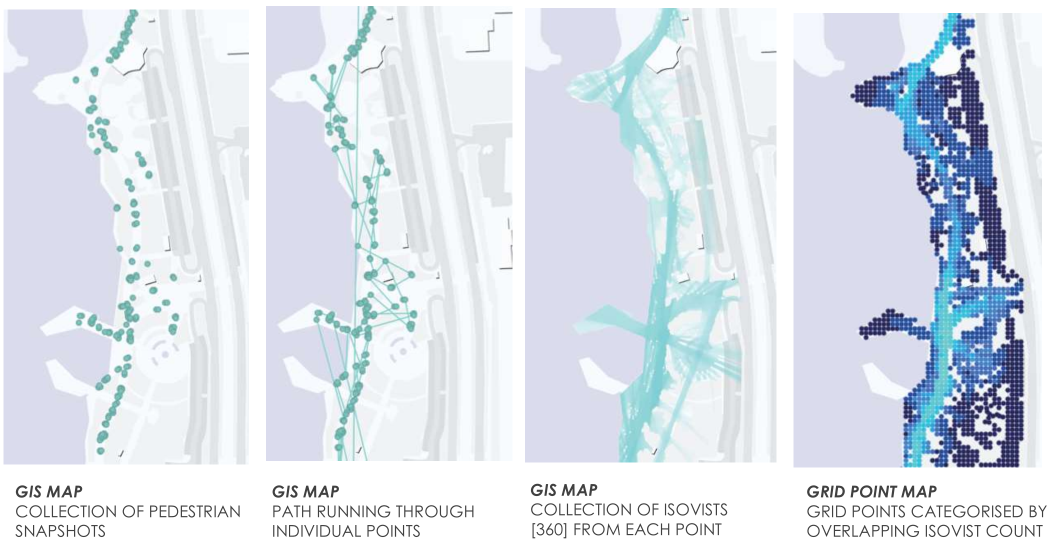


Figure 4.12. Depiction of how pedestrian visibility graphs are created using snapshot data.

4.4.2. The Isovist as a Tool for Verification

By overlapping isovists of waterfront visitors, the previous visibility model can be tested for accuracy. A grid point might be highly visible in theory, but it might have no actual visitors passing through that point for external reasons, rendering its high potential for visibility irrelevant While the visibility graph indicates the *visibility potential* a grid point has, a study of visitor isovists throughout the day indicates the *actual visibility* of each point, incorporating the element of visual *frequency* of each grid point.

To understand how well the visibility model represents actual visibility in the waterfront, a cumulative isovist graph based on visitor snapshots was created, to later correlate to the previously created visibility graphs. To create the isovist graph, no adjacency matrix is needed. Each grid point starts off with zero points, and has an additional point added to its overall score every time it is seen by a pedestrian [figure 4.12].

The resultant graph indicates how visible each point is depending on visitor movement and density. This graph can be used on its own to assess pedestrian movement and visual contact, or it could be used along with the visibility graphs as a calibration tool – to understand which visibility graph best represents reality and why.

4.5. KEY FINDINGS

To define the project scope, the Snapchat heat map along with traffic data from Google Traffic live will be used to determine the area of interest and the peak times. To understand the spatial characteristics, a revised segment model based on the literature was adopted to evaluate the waterfront movement metrics, addressing aspects such as pathway vs activity area modelling, fuzzy boundaries, and overrepresentation. Visually, a modified VGA model was created to account for all the environmental and contextual conditions encountered in the Northern Corniche Waterfront region, accommodating for larger grid points, one-way vision, attractor points, night time visibility, and visitor distribution. Finally, the snapshot method was chosen to assess activities on a micro level, allowing for the use of isovists to assess pedestrian engagement with the surrounding spaces and fellow pedestrians.

CHAPTER 5

ASSESSING VISITOR-WATERFRONT ENGAGEMENT DYNAMICS

5.1. CHAPTER OUTLINE

This chapter will use the tools developed in the previous section to understand the visitor-waterfront dynamic. It will begin by assessing the spatiotemporal distribution of pedestrians on a macro scale by combining the results of the Google Traffic data, the Snapchat heat map data, and the Jeddah axial map. It will then analyse the waterfront characteristics on a meso scale, by studying the adapted segment map to understand movement and the multiple visibility graphs generated. Finally, it will observe and interpret pedestrian activities on a micro scale, both in terms of pedestrian distribution, and in terms of how each subcategory of visitor interacts with the waterfront.

5.2. MACRO: DEFINING THE PROJECT SCOPE

To select a subsegment of the waterfront to study in more detail, online analytics were used to observe pedestrian traffic throughout the day. By understanding when people were the most active, and where they were allocated during peak hours, observational studies would be more representative of the typical waterfront activity patterns.

5.2.1. Defining Temporal Boundaries: Peak Hours

Traffic data using Google Traffic live analytics was imported into QGIS [figure 5.1] showing the 21-day average of the *Corniche Road* traffic at each of the following hours: 6 am, 9 am, 12 pm, 3 pm, 6 pm, 9 pm, 12 am, and 3 am. According to the data, 12 am – 3 am had the heaviest traffic, followed by 6 am, with 6 – 9 pm following shortly behind. The remaining hours saw no traffic at all, regardless of weekday/weekend status. This is most likely due to the high temperatures in Jeddah (magnified by the fact that this study was carried out in Summer), making the outdoor recreational space a night-time destination.

As traffic is an indicator of vehicular activity, and not necessarily interaction with the waterfront, an axial map of Jeddah was created and used to contextualize the information (figure 5.2.). The waterfront, although relatively high in global integration, ranks very low in global accessibility. This is in part due to it being on the edge – where it is not an ideal choice route for through-movement. This implies that only cars that are travelling *to* the waterfront, and any nearby destinations, would pass through this road. Additionally, the Corniche Road is filled with speed bumps and has an enforced speed limit of 50 km/h, as opposed to the 100 km/h speed limit on a neighbouring parallel main street in Jeddah (*King Abdulaziz Road*), making it even less likely that many cars would pass through the road for reasons other than the waterfront and the neighbouring facilities.

Finally, to factor out nearby destinations as the source of traffic, a site visit was conducted to understand the neighbouring active facilities, with only a few activity points found at the locations of high traffic. While there are a number of kiosks and small cafes, the waterfront was by far the largest attractor – with nearby hotels still under construction, and few other attractors (theme park, relatively close) that would generate this amount of traffic on a daily basis.

The traffic is also slightly indicative of the areas where the people might be the most active, while the Nawras strip has high traffic regularly, it is located along a roundabout and at a traffic light junction, creating a misrepresentative activity level. The northern areas (Fisherman strip, Pearl strip, etc.) seem to be the most active, with the activity level spreading further south as traffic increases.

As a result, this study can safely say that the traffic is representative of activity levels around the waterfront over the course of the day. Where peak hours during the day would be around ***6 am***, and peak hours at night would be from ***12 – 3 am***, with traffic bypassing the daytime peak hour activity levels.

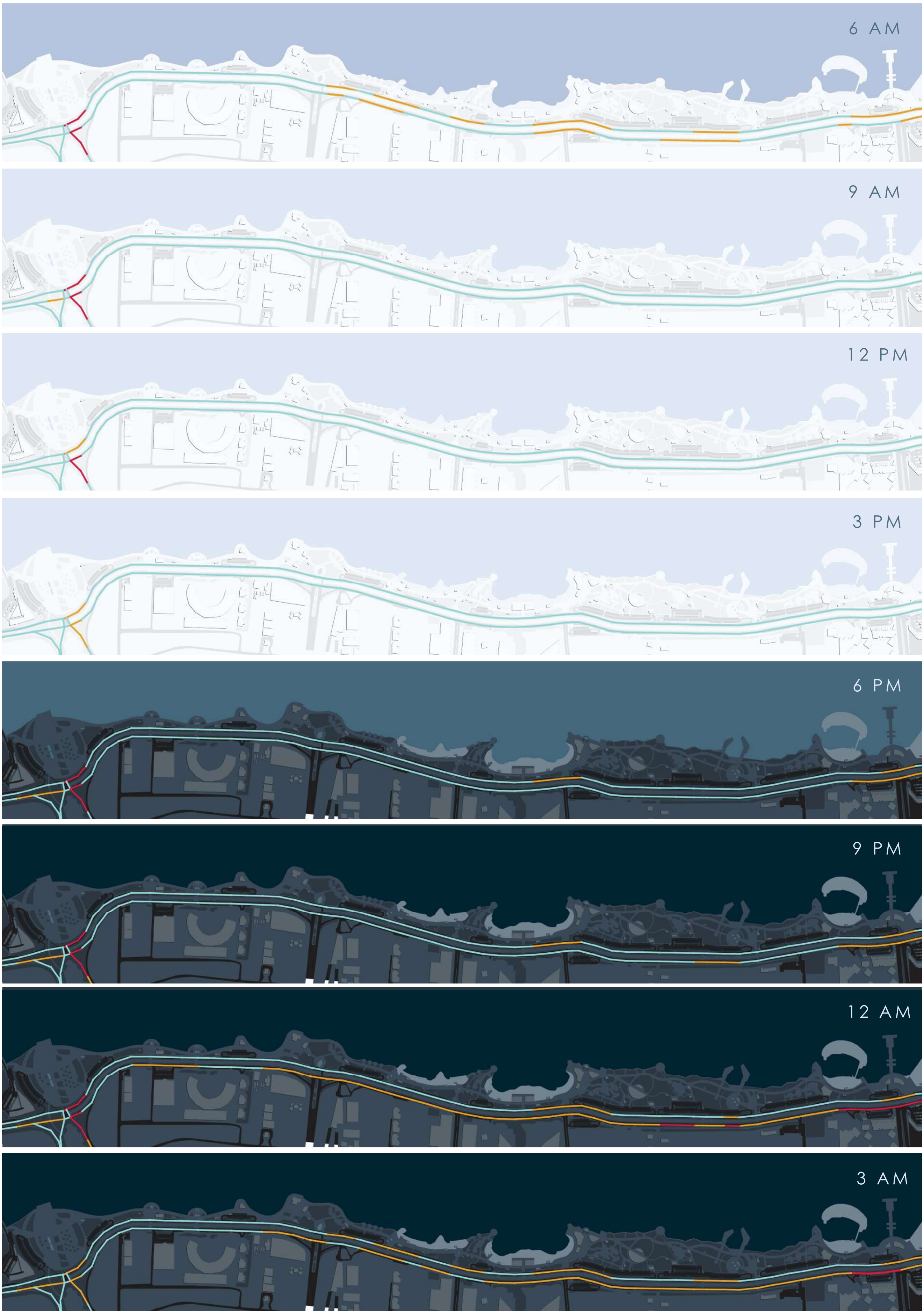


Figure 5.1. Average Google Traffic Data (21 days) displayed by the hour to illustrate peak times and high activity zones.

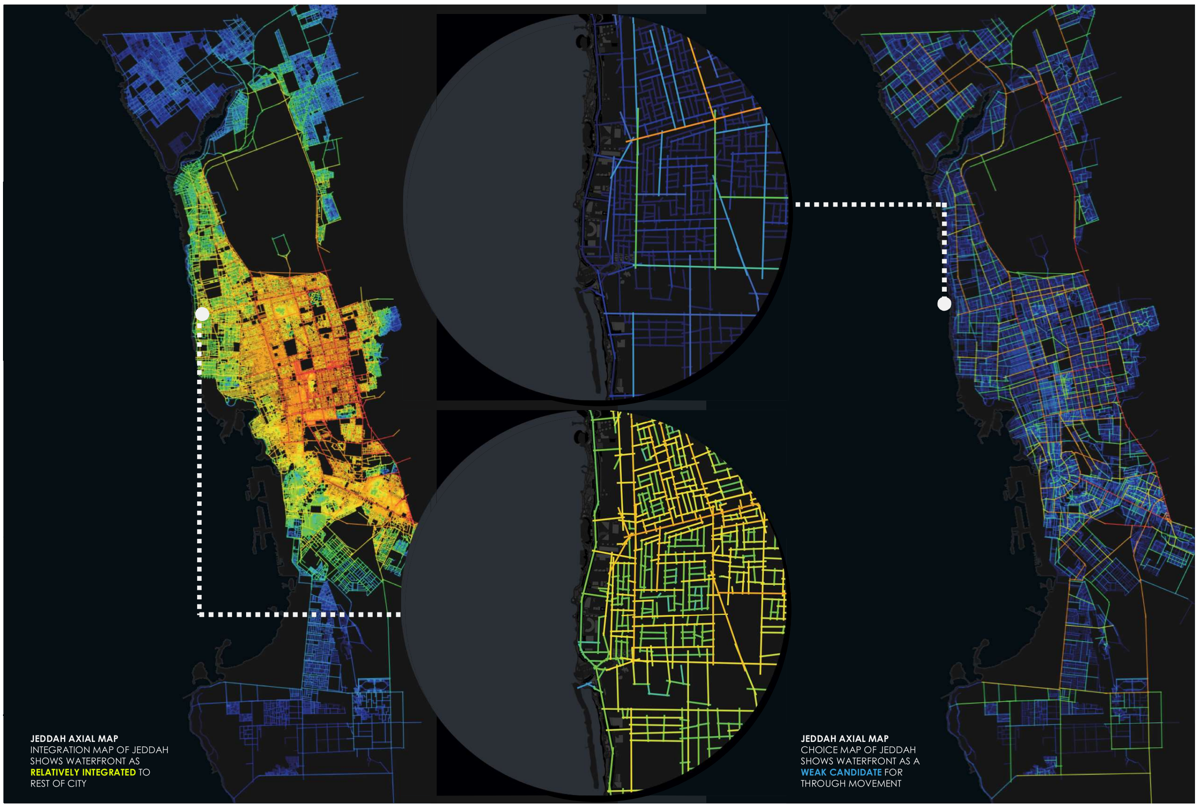


Figure 5.2. Location of waterfront on Jeddah's axial map: integration map [left] and accessibility map [right].

5.2.2. Defining Study Area Boundaries: Activity Level Assessment

While the traffic maps showed an indication of higher vehicular traffic around the northern region of the waterfront, Snapchat heat map activity levels were studied to get a more direct reading of pedestrian, and not vehicular, activity levels. Figure (5.3) shows the 4 sub-regions of the Jeddah Northern Waterfront that were consistently within the red zone of the heat map, indicating high activity levels.

While Snapchat users are not representative of the entire demographic, the fact that the highlighted areas also ranked high in vehicular traffic confirms the high activity level overall, and not just a virtually high activity level on social media represented by the younger demographic.

5.2.4. Scope of Study

Based on activity levels and traffic data, the remainder of this study will focus on the Pearl strip, and the Fisherman strip exclusively, as the Sand strip has been inconsistently open/closed to the public over the course of this research and the Gulf strip is still under construction. Any site visits and observations during the day will take place between 6-7 am, and visits during the night will take place at 12-1 am.

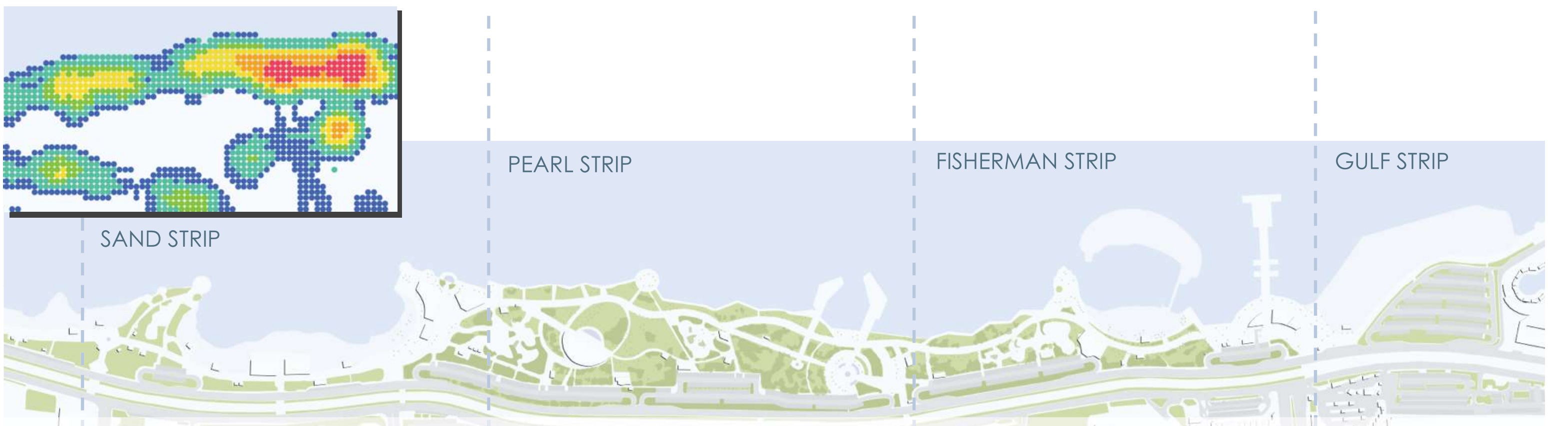


Figure 5.3. Snapchat heat map results, with regions within the highly active red zone magnified and displayed in the image below.

5.3. MESO: EXPLORING WATERFRONT CHARACTERISTICS

After defining the scope of study, this section explores the routes of movement through segment angular analysis, and visibility through the model created in the previous section. The characteristics of the waterfront are defined and summarised to contextualise pedestrian activity in the following section.

5.3.1. Routes of Movement

According to the segment angular analysis results (figure 5.4), the waterfront has a number of smaller local hubs that are dispersed, two main global hubs that are more centralised, and a linear route moving through them all that is both high in integration and in choice. Space *B* is mostly a large park-like area including a playground, with a viewing dock near the sea, this region seems to be central to both through-movement and to-movement indicating a high level of pedestrian activity, and a varied level of engagement – as it would function as both a passageway and a destination. Space *D* is a paved plaza, with a large central water feature leading up to two piers (one of which is out of service). This area seems to play an important role in the spatial make-up of the waterfront as it is a local hub, a global hub, and a central part of the accessibility core. Areas *A* and *D* seem to function as local hubs, *A* being a viewing dock, and *D* a plaza connecting pedestrians to a long fishing pier.

It then seems that the waterfront as a whole is well connected, with a central global centre and distributed local centres along the far ends of the region.

5.3.2. Visibility

To understand the visibility measures of the waterfront, several maps were created measuring connectivity (figure 5.5). For the daytime measures – a radius of 150 meters was found to best represent perception at the waterfront and was used as a base for all daytime graphs, while a radius of 100 meters was used for the night-time graphs as

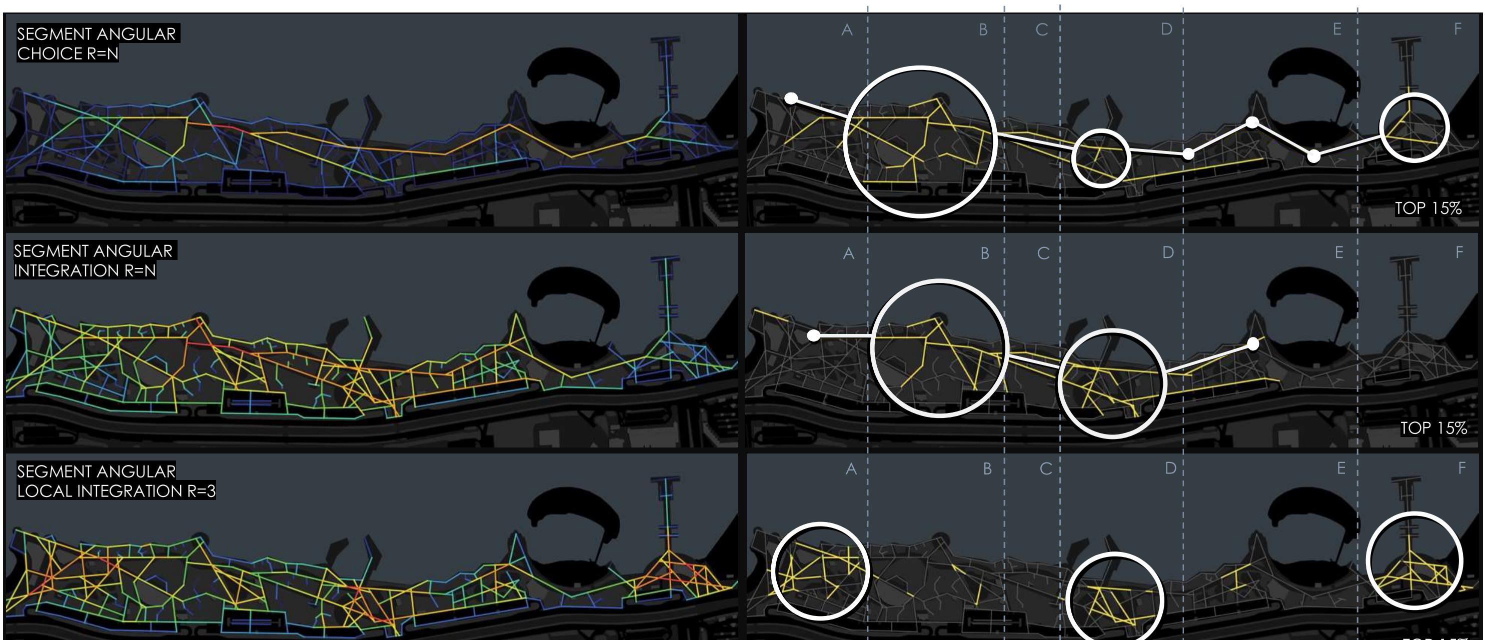


Figure 5.4. Segment map of waterfront study area showing metrics of choice, integration and local integration along with their respective top 15% core.

visibility was lower at night. Daytime graphs included a standard visibility graph, a weighted graph, a local visibility graph, and a graph showing global connectors. Night-time graphs included a night-time visual potential graph (what one could see), a weighted graph, and a night-time visual vulnerability graph (how much one is seen). Weighted graphs factor in the sea as an attractor – where added value is given to sea nodes by adjusting the weight factor. Several graphs were created with weights of 5, 10, 15, 25, 35 and 45 to understand the behaviour of the visibility model as the weight changes.

In the standard visibility graph, area *B* seems to rank high in visibility measures, as well as the two piers (in areas *D* and *G*). Lines of movement that ranked high in integration and choice are also high in visibility, with dark blue patches of heavy vegetation creating pockets of secluded areas in between. The weighted graph (w=25) is a heavily skewed model for reference, where most of the value is attributed to areas that view the largest amount of sea, thus the hotspot in the extended pier at *G*. With the exception of, *F*, which seems to act as a corridor, and the heavily vegetated areas of *B* and *C*, the local visibility graph shows most areas to be surrounded by spaces of high local visibility. The map of global connectors shows areas with longer range of visibility as opposed to high local exposure. Here routes of movement seem to be highlighted, as they consist of longer paths- as well as extended piers.

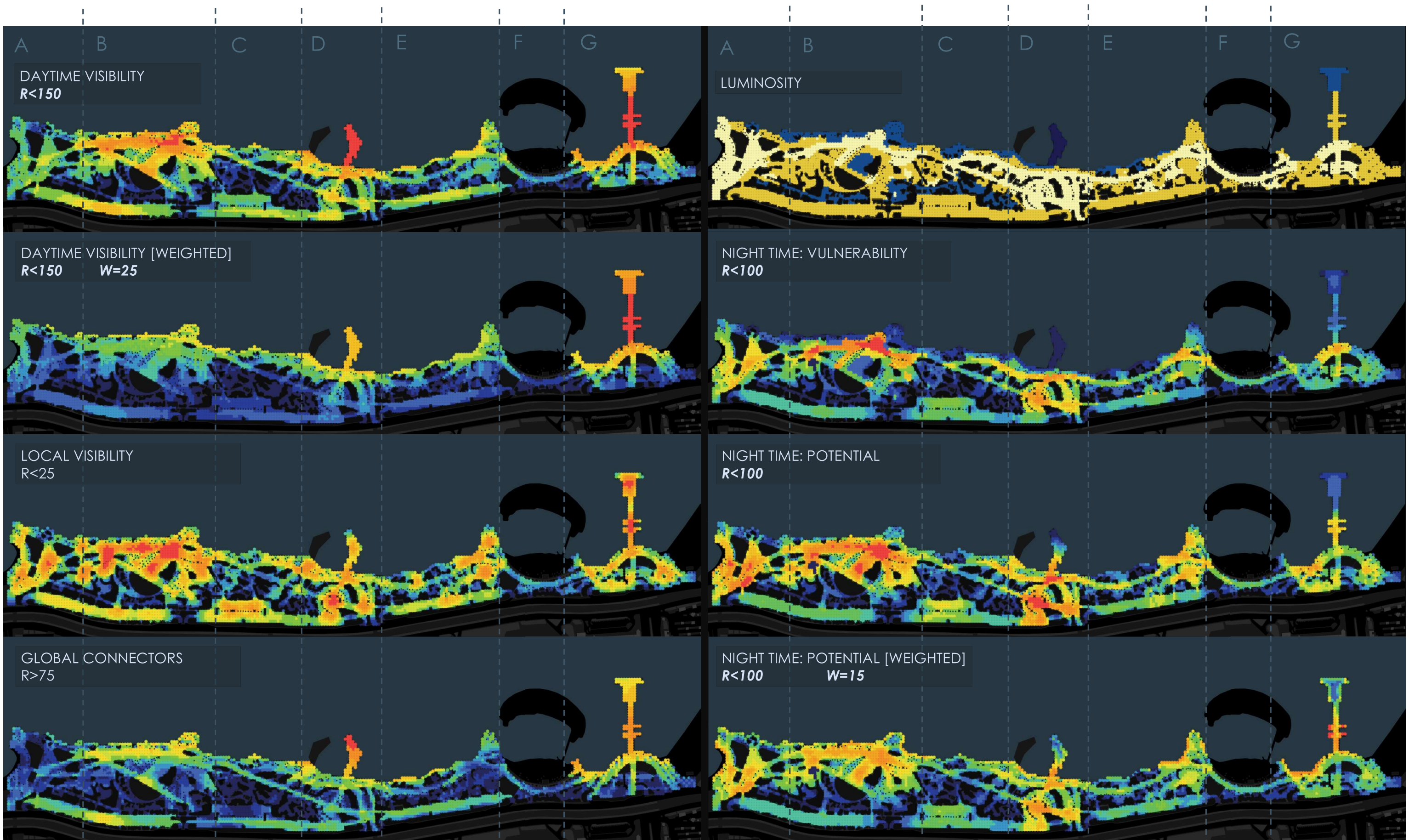


Figure 5.5. Visibility maps of the waterfront. [Left] Top to bottom showing daytime metrics: standard visibility, weighted visibility, local visibly, and global connectors. [Right] Top to bottom showing night-time metrics: luminosity, night-time vulnerability, night-time potential, and night-time weighted.

The night-time model incorporates the luminosity values into its calculations. Areas *B* and *D* are shown to be the most visually vulnerable, they are also the areas with the highest visual potential. These areas mirror the globally integrated regions in the segment model. The extended piers behave differently at night as they are not well lit; the weighted model then only affects the pier edges. As the weight increases, the waterfront edges increase in value, creating a boundary of high visual value surrounding an area of low visibility still.

5.3.4. Spatial Characteristics of the Jeddah Waterfront

A visual summary was created to represent the important areas and hubs as characterised by the spatial metrics (figure.6). The summary illustrates areas of high and low visibility during both the day and night, routes of movement, low visibility areas, inaccessible areas, and important activity points.

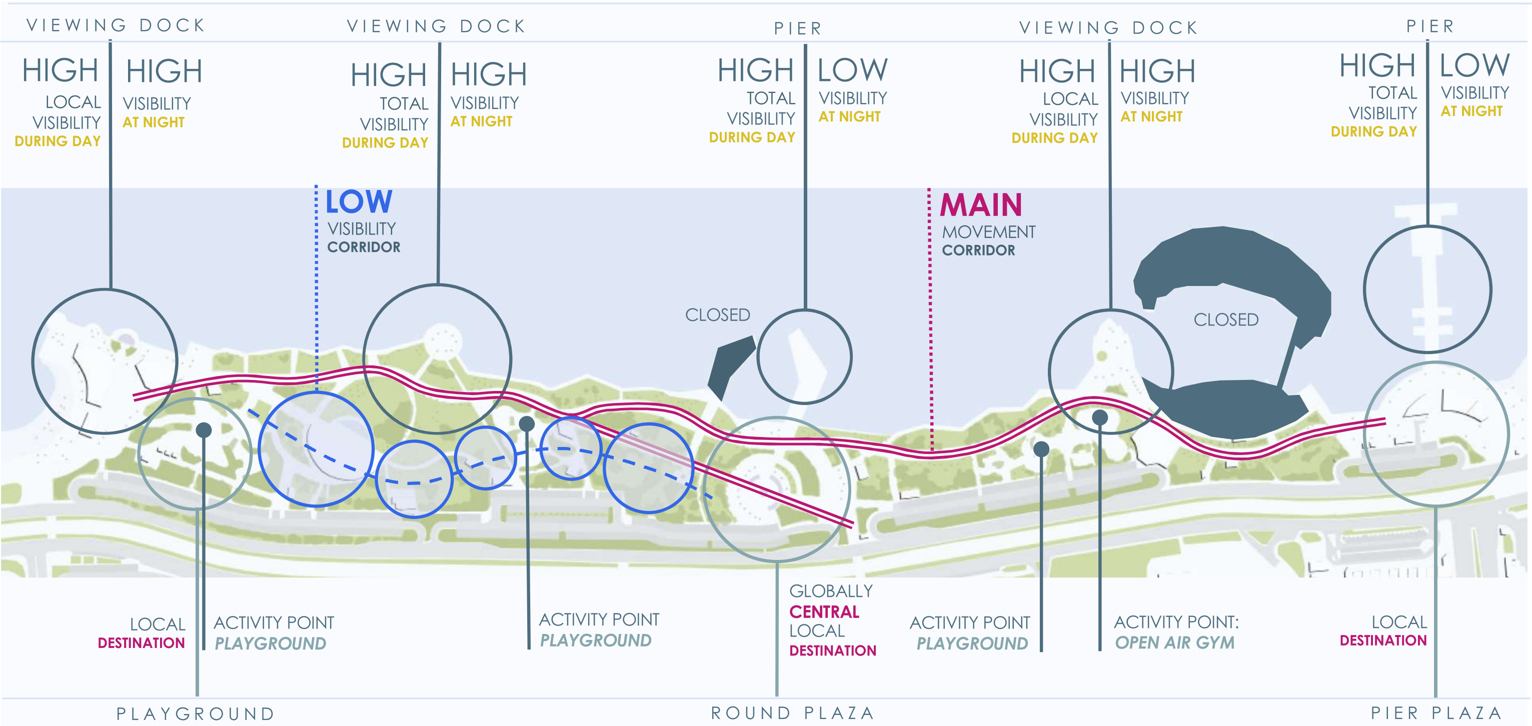


Figure 5.6. Visual summary of spatial characteristics of the waterfront study area.

5.4. MICRO: VISITOR-WATERFRONT ENGAGEMENT DYNAMICS

Based on the previous segment, this section will explore pedestrian activity and see if there are any consistencies or discrepancies with the predicted movement patterns as indicated by the spatial characteristics. Pedestrian distribution based on snapshot data would be explored; additionally, pedestrian activities would be studied to understand repeated behavioural patterns of engagement.

5.4.1. Pedestrian Distribution

5 site visits were conducted throughout the course of a week (8 – 14 August 2021). A total of 1409 visitors were recorded over the course of 3 days for the daytime snapshots, and a total of 2328 snapshots in total were recorded over the course of 2 nights for the night time snapshots. All snapshots were recorded mid-week (no weekend data). As the site is large, each visit took around 1-2 hours with each visited area counted once before moving on to the next – creating a metaphorical collage from a series of back-to-back snapshots.

The resultant distribution is illustrated in figure 5.7 where the overwhelming observation is the distance of the visitors to the sea. The effect of the sea as nana attractor is visually obvious, as most pedestrians are found near the waterfront edge, if not on the extended piers. To map out the pedestrian distribution statistically, the distance between each visitor and the closest sea point was recorded. A frequency chart was then created with the number of people mapped against the distance from the sea. For comparison, the same chart was created for the number of grid points – to illustrate the distribution of the potential areas the pedestrians could have visited instead. While the relationship between the graph points to the sea was linear, that of the pedestrians to the sea was exponential – showing an undeniable bias towards areas near the waterfront edge. Additionally, the populated areas that are of a certain distance away from the waterfront edge happen to be activity points (playgrounds and gyms, i.e. other smaller attractors).

Where pedestrians are *not* found, are the globally integrated and highly visible regions of the waterfront. Both the paved plaza with the water features, and the large stretches of green park-like spaces on the left, are constantly found empty. This holds true for the path that runs through the central paved plaza and ranks high in integration, choice and visibility on all scales, during both the day and the night. Additionally, these areas are all well maintained and well-lit at night (with the exception of the park-like space), with no visible detractors or environmental repellents.

As the snapshot distribution clearly shows that pedestrian movement is influenced by the sea as an attractor, an attempt to measure and model pedestrian behaviour was made by creating a visibility graph based on pedestrian isovists (figure 5.8). Here the more a grid point is seen by a pedestrian, the higher its value. This isovist visibility model was then correlated to the various visibility graphs created, where a highly correlated model would have two implications. First, it could be used to predict pedestrian behaviour in different waterfront segments in future, perhaps to test other waterfront design features before launching. Second, it gives more weight to the reliability of the results. Showing that the behavioural patterns of pedestrians are not sporadic in nature and could be modelled allows more concrete conclusions to be made – as a behaviour that can be modelled implies consistency and predictability by definition.

The results of the correlation are shown in table 5, plotting the isovist against the classic visibility model (DepthMap), the adjusted unweighted model, a range of various weights representing the importance of the sea. High correlation values were found between the daytime isovists and the weighted models (with 0.78 being the highest, associated with a weight factor of 15). Night-time isovists, however, had relatively low correlation values, the highest being with the daytime weighted graph – implying that visibility is not a factor in movement, as pedestrian movement patterns at night (on a macro scale) are not dependant on lighting, but on once again, the sea.

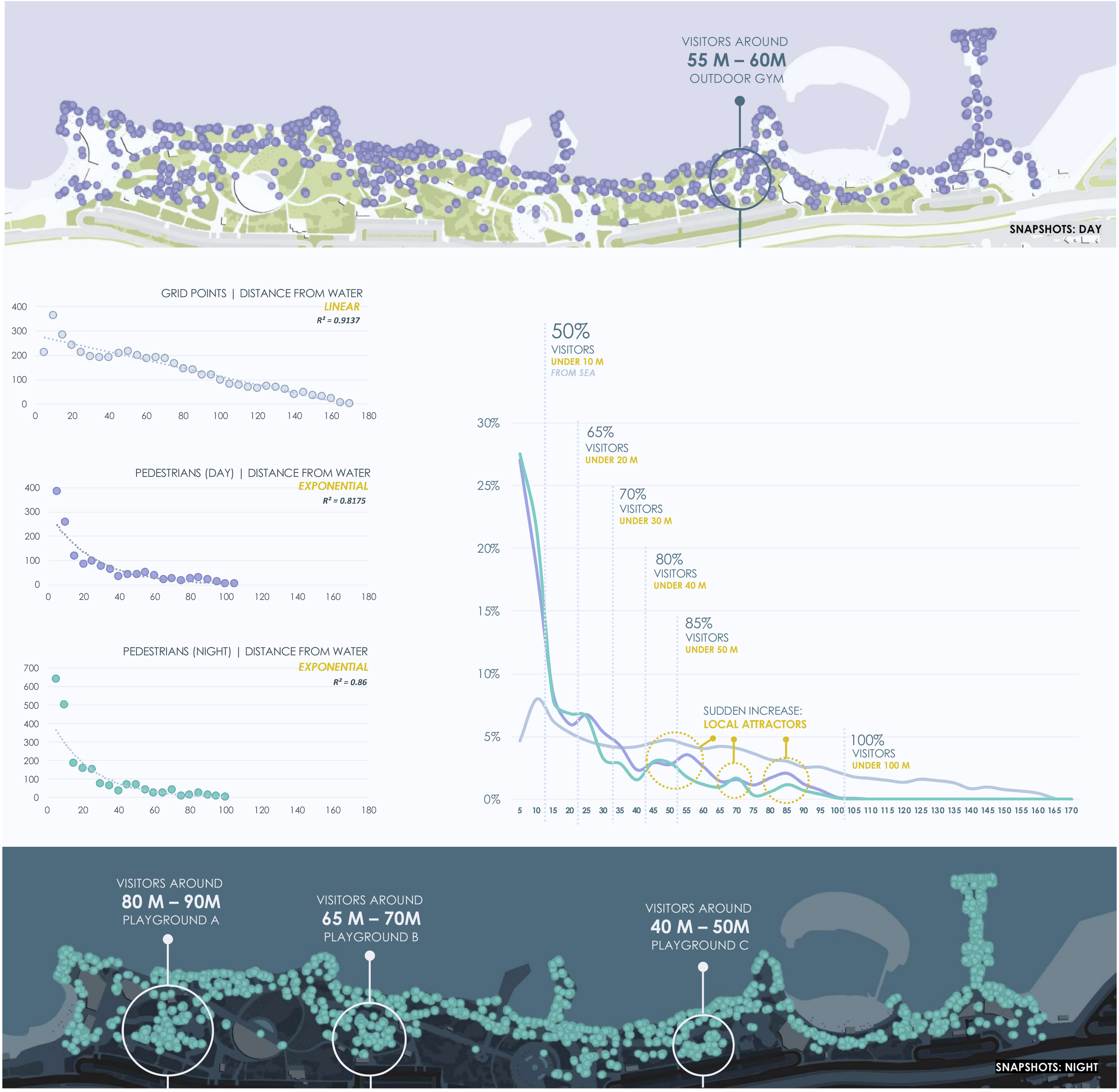


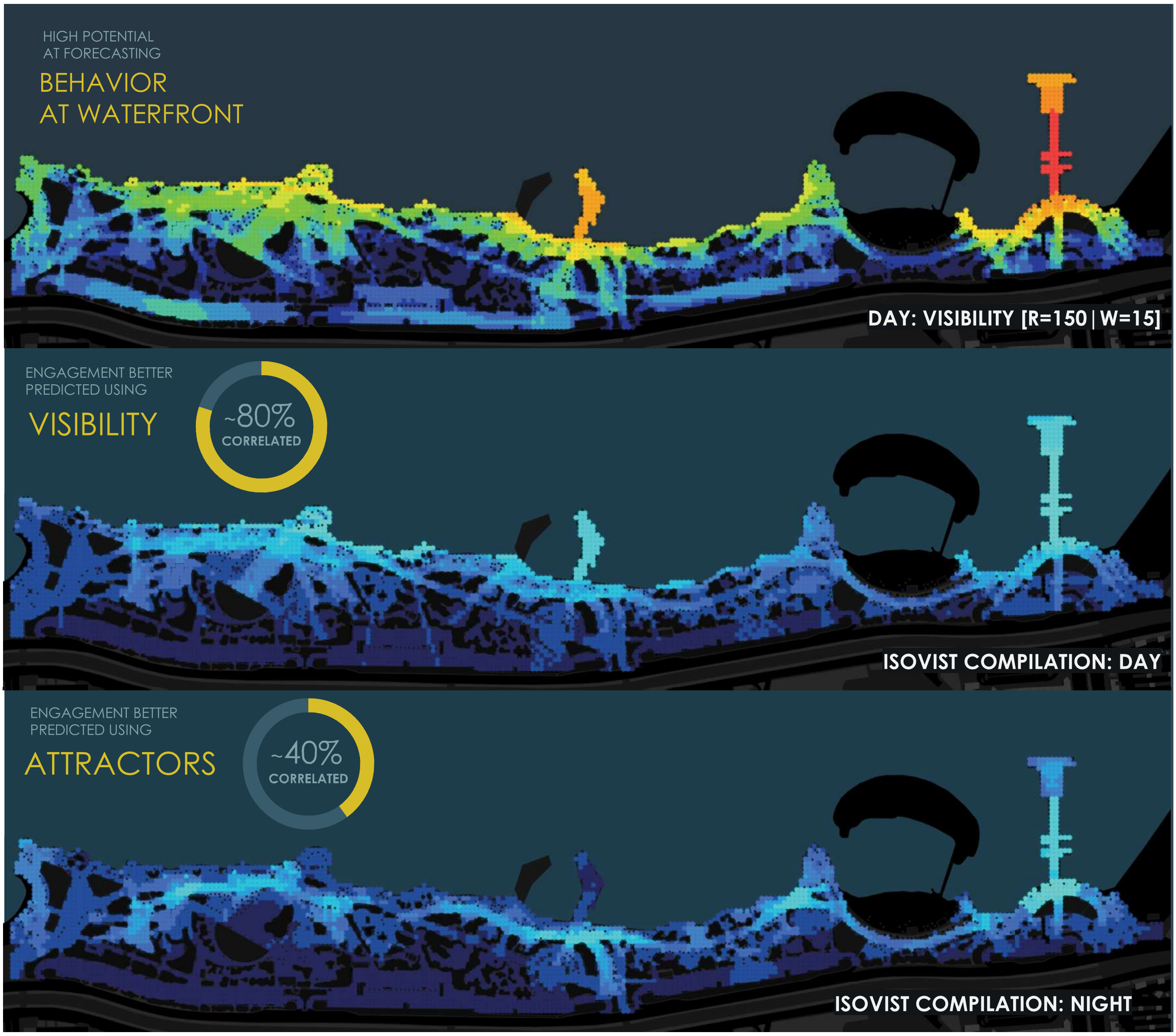
Figure 5.7. Pedestrian distribution metrics evaluating the sea as an attractor. Figure includes general visual snapshots of pedestrian location and graphs comparing pedestrian frequency to distance to waterfront edge.

		R	R^2
DAYTIME ISOVISTS	*Classic*	0.18	*0.03*
	Day W=1	0.75	*0.56*
	Day (P) W=5	0.85	***0.72***
	Day (P) W=10	0.88	***0.77***
	Day (P) W=15	0.88	***0.78***
	Day (P) W=25	0.87	***0.76***
NIGHTTIME ISOVISTS	*Classic*	0.15	*0.02*
	Day W=1	0.51	*0.26*
	Day (P) W=5	0.59	*0.35*
	Day (P) W=10	0.62	***0.38***
	Day (P) W=15	0.62	***0.39***
	Day (P) W=25	0.62	***0.39***
	Night (V)	0.56	*0.32*
	Night (P) W=1	0.39	*0.15*
	Night (P) W=5	0.42	*0.18*
	Night (P) W=10	0.45	*0.20*
	Night (P) W=15	0.47	*0.22*
	Night (P) W=25	0.48	***0.23***
	Night (P) W=35	0.47	*0.22*

sample size (n) = 4547
$p < 0.05$ for all above values is

Table 5. Correlation statistics illustrating the relationship between different visibility models (standard, weighted, night-time, etc.) and the isovist visibility model.

Figure 5.8. Weighted daytime visibility model [w=15] [top], daytime isovist visibility model [middle], and night-time isovist visibility model [bottom].



5.4.2. Pedestrian Activity Sub-Groups

While conducting the snapshot observations, the activities pedestrians were engaged in were recorded. These included activities such as playing, exercising, walking, sitting, observing, etc. Out of all the activities, four were the most common, and the most representative of pedestrian engagement patterns along the waterfront [figure 5.9]. These activities were: observing (standing near the waterfront edge), settling (sitting on the floor in style similar to that of a picnic for an extended period of time), travelling (walking), and resting (sitting on a designated seating area, or bench).

It should be noted that while both observing and settling are flexible activities, in the sense that they are dependent on pedestrian intent, travelling and resting are dependent on design features and designated seating. The location of the second two activities is somewhat controlled (weakly, as it is a park – one could still veer off the designed pathway). There is value still in the data gathered as there is still value to be gained in understanding why some walkways and benches were chosen over others. However, studying those engaged in observing and settling provides the most insight, as it is a pure form of visitor engagement through their own preferred means – choice of location is intended and not (directly) a consequence of the current design.

The following section will explore each of these activities in detail, including ***what demographic*** engages in this activity, ***what they do*** in specific, and ***time engaged*** in activity, and, more specifically, their prominent ***spatial characteristics***. These characteristics were explored by mapping out the location of these subgroups onto the two sets of graphs. To explore the spatial characteristics of the daytime visitors four graphs were studied standard visibility (visibility potential and vulnerability), weighted visibility [w=15] (visibility skewed by value of attractor), local visibility (short range visibility), and global connectors (long range visibility). Night-time visitors were measured against a different set of graphs, night-time vulnerability (how seen one is), night-time potential (how much one can see), night-time weighted [w=25] (visibility skewed by value of attractor) and daytime weighted [w=15] (valuing attractor over/despite visibility).

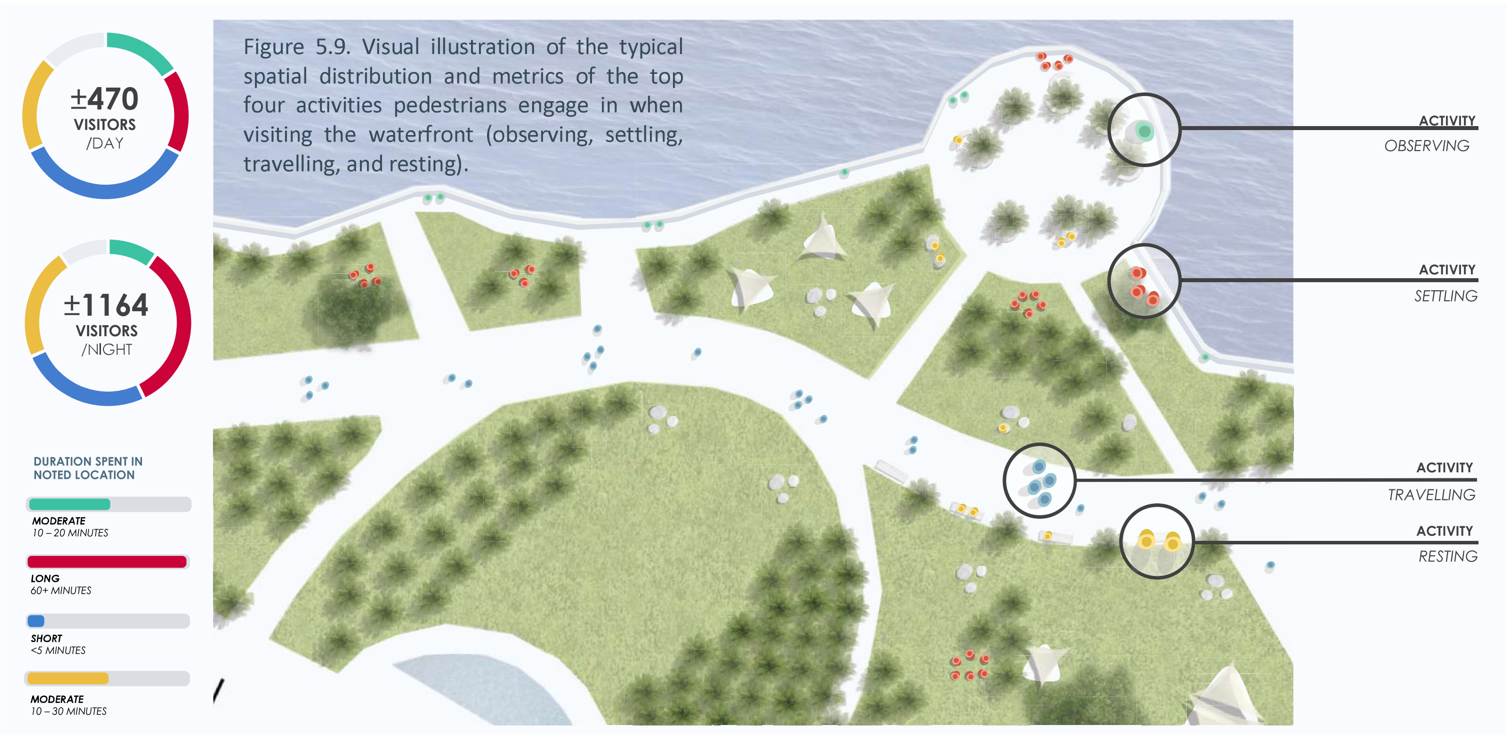


Figure 5.9. Visual illustration of the typical spatial distribution and metrics of the top four activities pedestrians engage in when visiting the waterfront (observing, settling, travelling, and resting).

5.4.2.1. Observing

Those observing are typically found near the waterfront edges, either along viewing docks, or on the extended piers. This form of activity is a direct engagement with the sea, with the waterfront acting as a proxy. This is made clear in the statistical results. During the day, the subgroup associated with this activity is located in areas with relatively high viewing potential, when the graph is weighted, the numbers are slightly skewed to the right showing an average increase. This implies that this subgroup is not attracted to areas with high visibility, but to areas with high visual access to the sea. This subgroup ranks higher than all other groups when comparing *visual value* on the weighted graph. They rank average on local visibility, and high in global connectors as they are found on the waterfront edges and piers – all known for their high global connectivity.

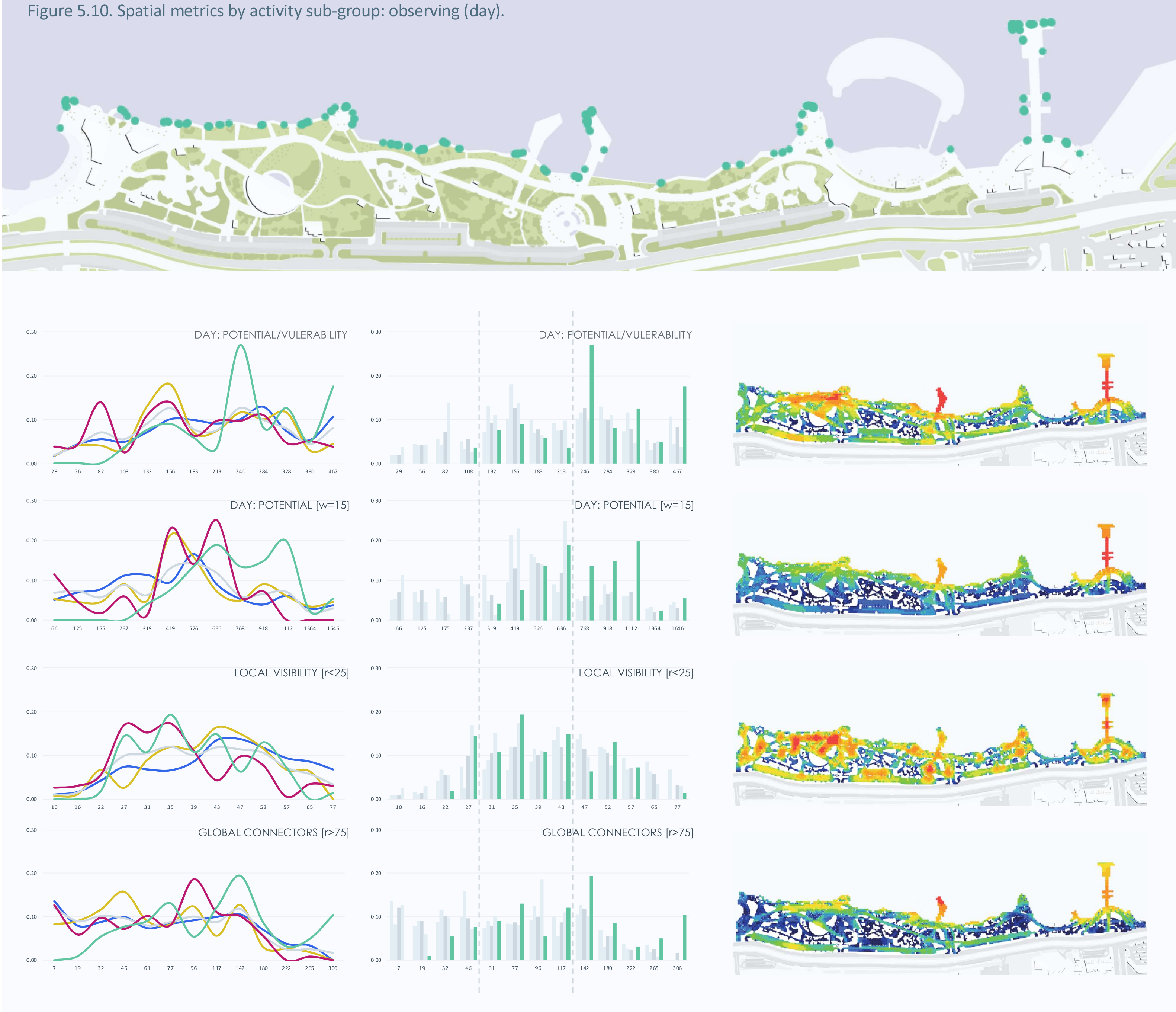
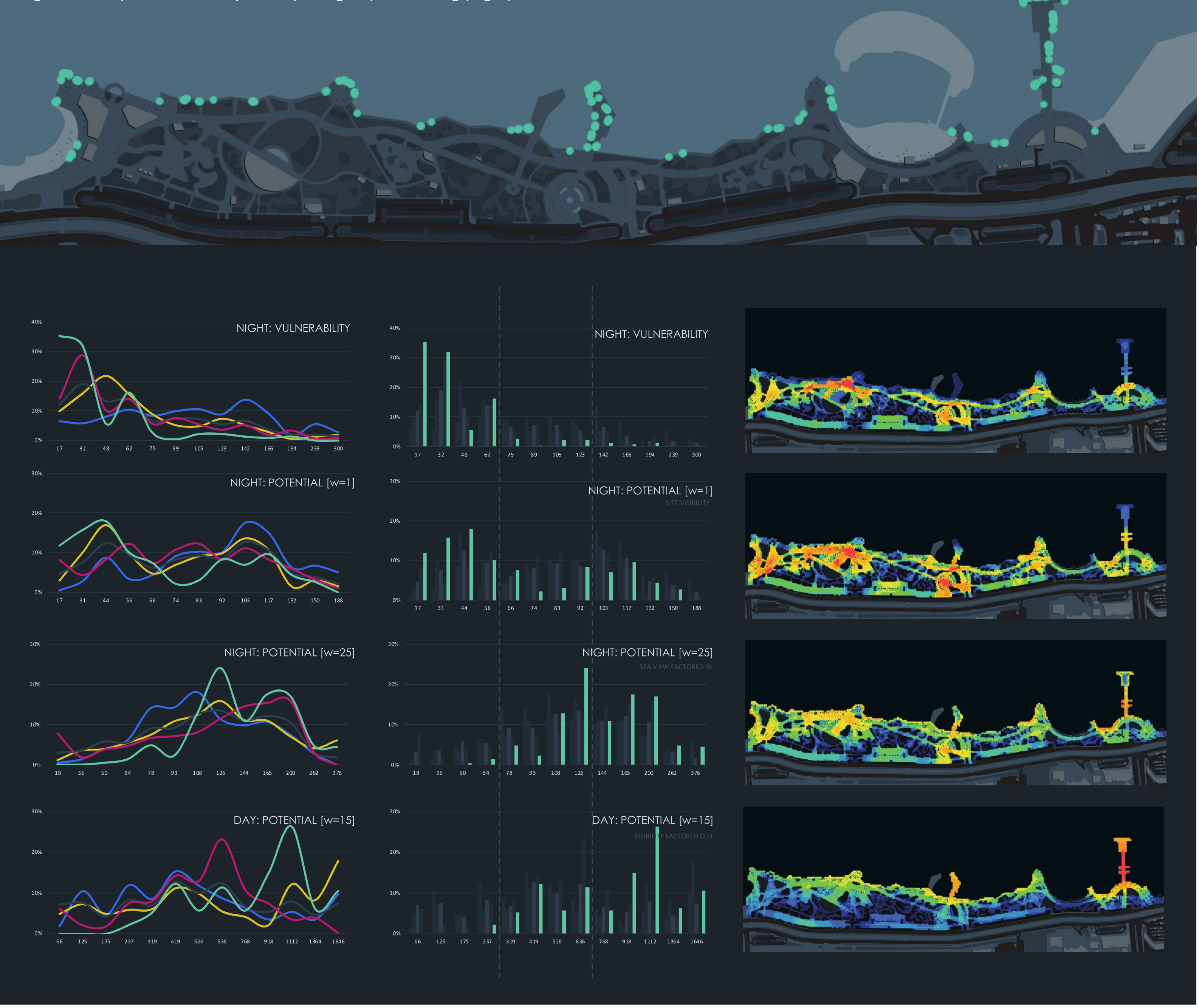


Figure 5.10. Spatial metrics by activity sub-group: observing (day).

At night, they are located in areas with low vulnerability. This does not imply much, as the observer is there for the sea – and almost all edges and piers around the waterfront are dark. The observer is then in areas that are not seen at night because the edges happen to be dark, and not because they necessarily seek out areas with low visual vulnerability. The next three graphs are very telling of the intent of the observing individual; on a regular night-time visual potential graph some rank high, but most rank low, as piers have next to no visual potential at night (in an unweighted graph). When weighted, the numbers go up, the observers are now in areas with high visual potential. Interestingly, when tested against the daytime weighted graph – the numbers go up even further. Indicating that their patterns follow the attractor, despite the lighting conditions at night, and not alongside them.

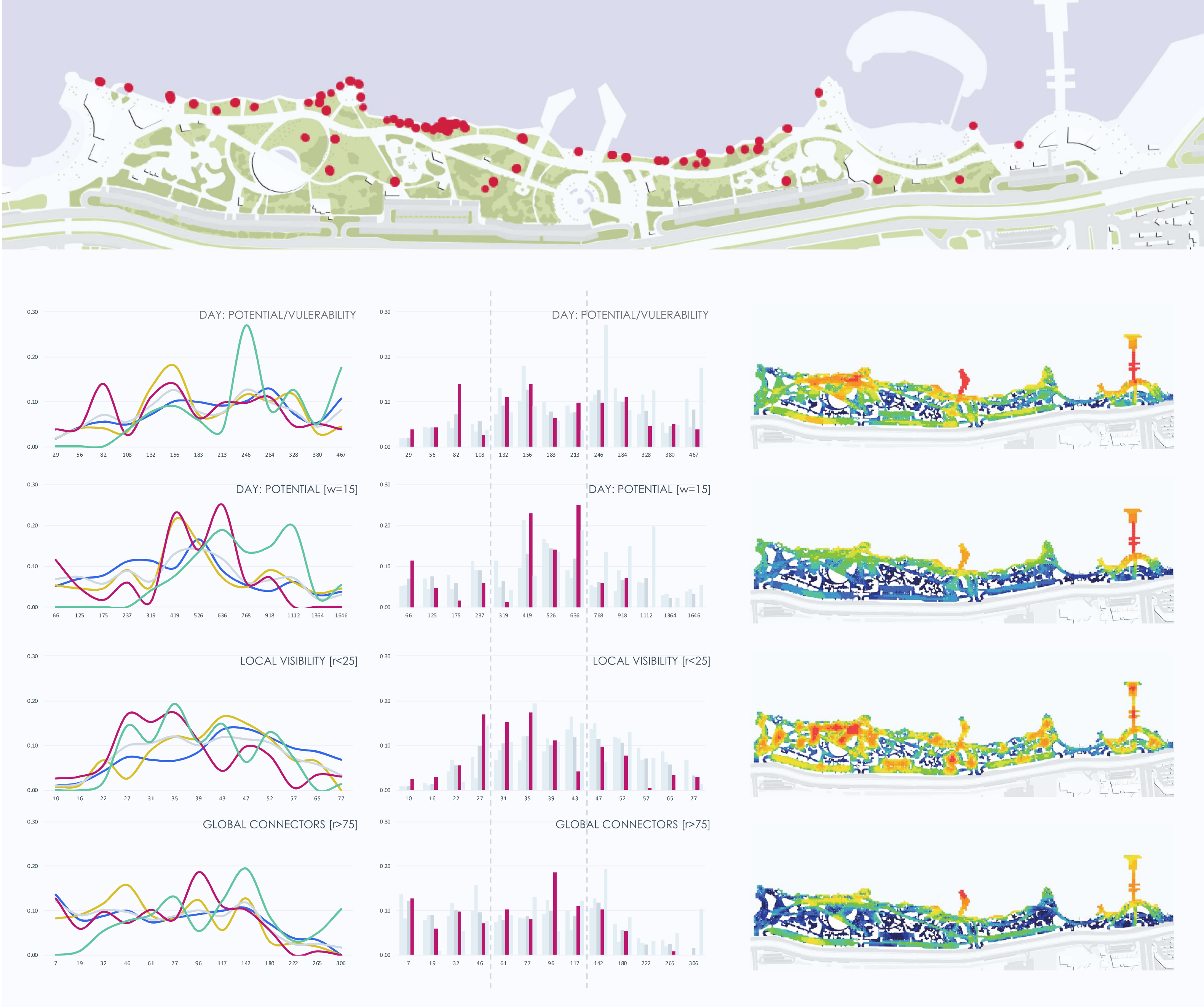
Figure 5.11. Spatial metrics by activity sub-group: observing (night).



5.4.2.1. Settling

Those settling are often found in groups on grassy areas near obstacles that keep them shielded from the views of others. These obstacles might be shrubs, benches (using the bench as a wall), or the dark, at night. Pedestrians who settle tend to stay in one spot for an extended period of time, often surpassing an hour. This activity subgroup tends to indirectly interact with the sea, often choosing areas near the waterfront edge, but also close enough to the integrated regions of the waterfront. The standard daytime graph doesn't provide much insight, but of all the other subgroups, this activity has the least amount of people in areas with high *visual value* on the second graph. They also rank the lowest on both local and global visibility, avoiding areas that are visually vulnerable to those near and far.

Figure 5.12. Spatial metrics by activity sub-group: settling (day).



At night this subgroup has the highest percentage increase, doubling from 17% of the total visitor subset to 33% (the largest activity subset) at night. During the night, those settling are located in areas with low vulnerability, and a non-remarkable visual potential. The graph shifts slightly to the right when introducing the weight of the sea, implying a higher interest in the visual potentials of the attractor, as opposed to the visual potential the waterfront has to offer as a space. However, when the plotted along the daytime weighted graph – the values decrease again. This implies that the sea is not the biggest attraction factor according to this subgroup, and they do not necessarily prioritise the sea view over external visibility conditions, unlike the previous category of observers.

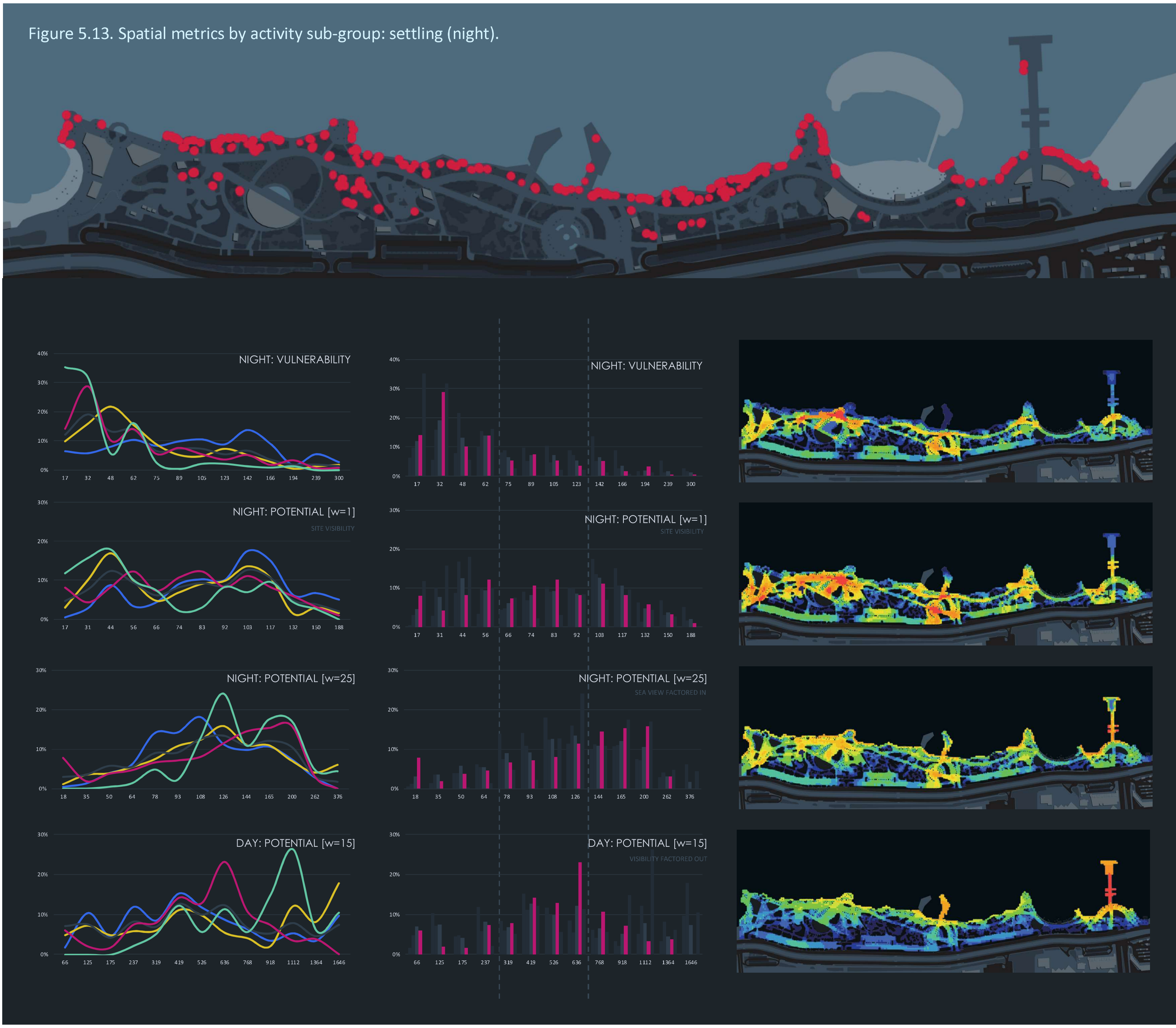
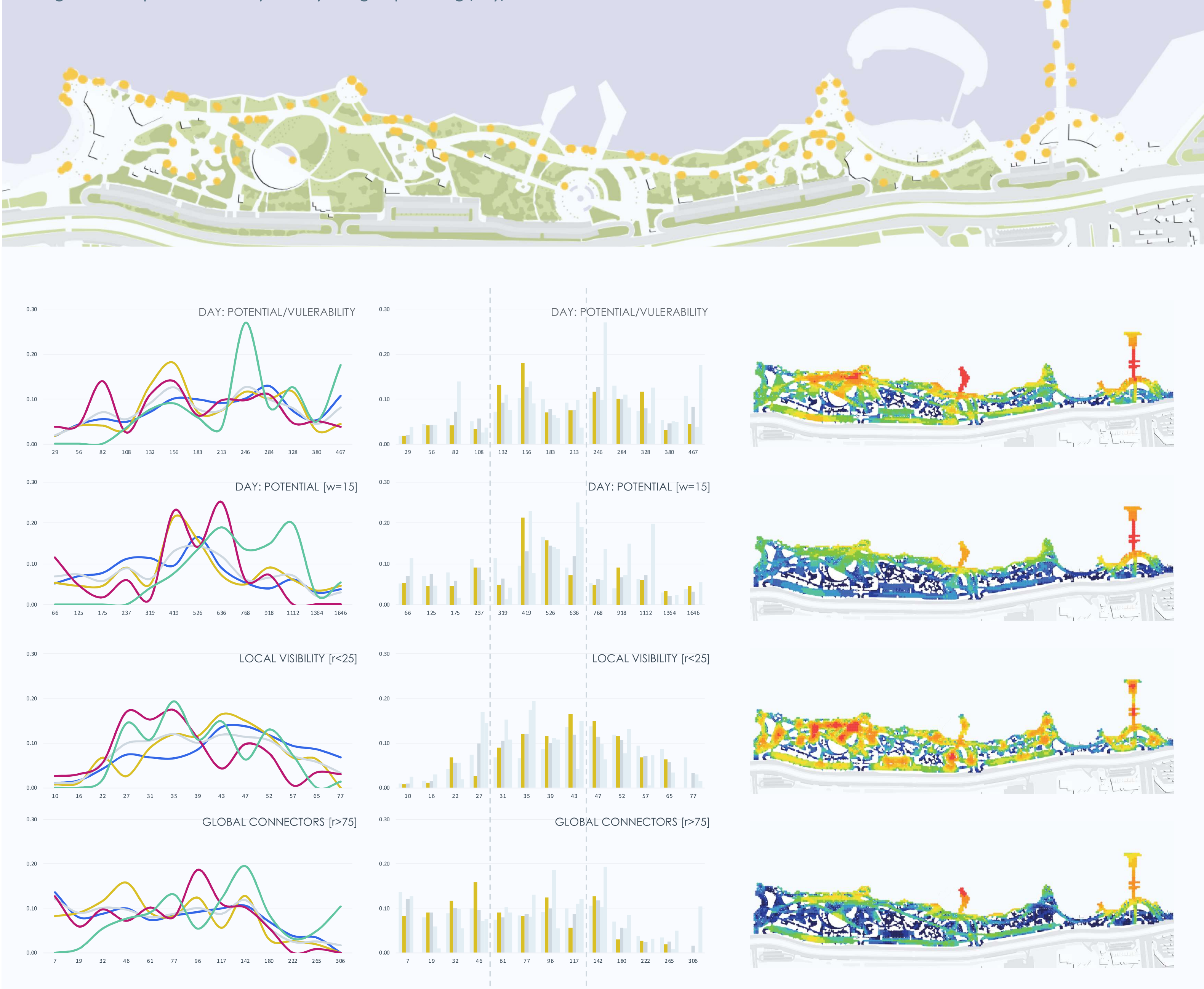


Figure 5.13. Spatial metrics by activity sub-group: settling (night).

5.4.2.3. Resting

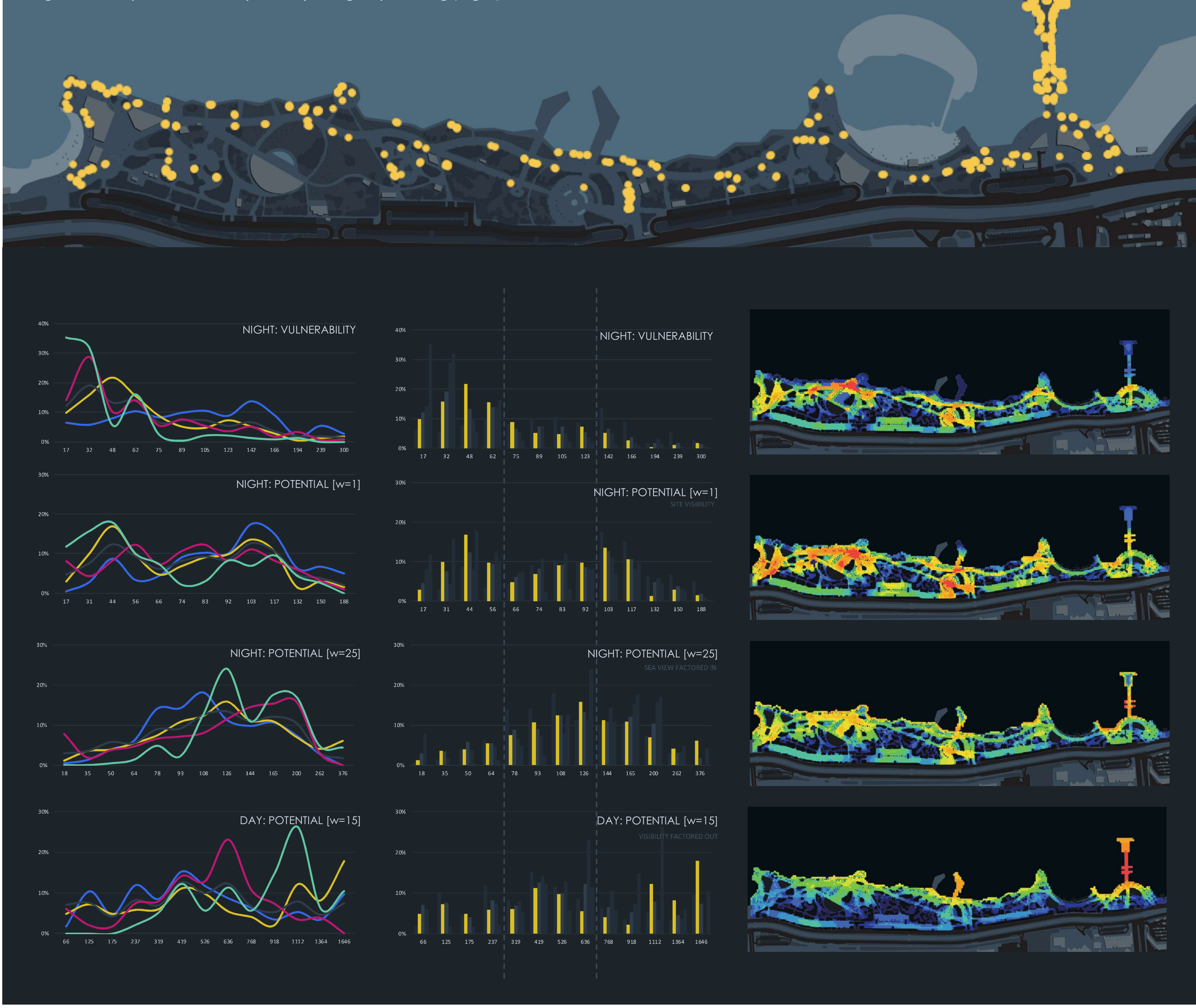
Those involved in the activity of resting are found sitting on benches along walkways or piers. Those on benches are less clearly defined in their intents – but they generally fall under two distinct sets; while some are resting temporarily, others act as observers that choose to sit. As such the data set contains two different subgroups driving the numbers on the graph, where conclusions are less clear and somewhat distorted. The graphs below do not give conclusive information on this subgroup, as even the high local integration values are motivated by the design allocation of benches. Perhaps the only indication of value is the discrepancy between the lines representing those resting and those travelling; as the assumption that both are accessing designed paths and seating, allocated in the same areas – and are engaged in temporary activities implies that their spatial patterns should align. Any disjunction in the pattern could then be indicative of the observing group that is simply observing the sea from a seated position – and motivated by attractors.

Figure 5.14. Spatial metrics by activity sub-group: resting (day).



This is made clearer in the night time graphs, where two distinct peaks are shown to represent each subgroup within the resting category. Most of those sitting rank low in vulnerability, implying darker areas (most likely the pier), with a small second peak ranking slightly higher. The distinction is clearer yet in the second graph – where there are two distinct peaks, one ranking lower in waterfront visibility metrics (those assumed to be observing), and one ranking higher (those resting along the pathways), which again is parallel to the metrics of those travelling. Justification for this theory lies in the unexpected peak along the final daytime graph – comparable (higher even) only to the observer subgroup. Where the graph once again has two peaks, one lower in value and aligned to a peak in the traveller subgroup (those sitting along integrated areas), and one large peak in areas that rank high in visual value despite lighting conditions (those observing the sea).

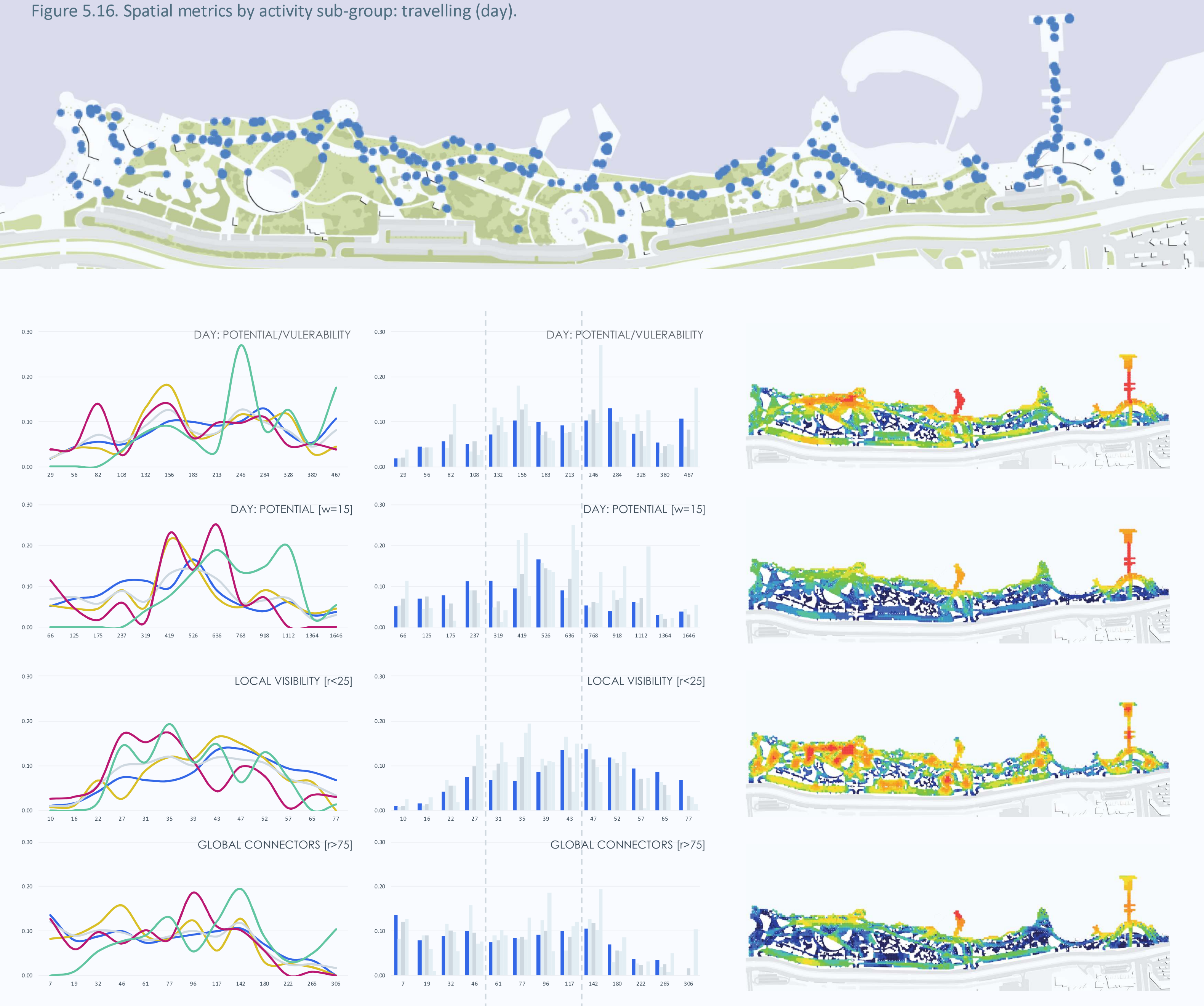
Figure 5.15. Spatial metrics by activity sub-group: resting (night).



5.4.2.4. Travelling

Those travelling walk along integrated routes of movement. It is unknown how many of them are engaged in a temporary activity of moving from one place to another, and how many are walking recreationally. In either case, their activity patterns are dictated more by spatial characteristics, and less by attractors when compared to the other subgroups. The distribution of the travellers is slightly skewed towards areas with high visibility, the numbers *retract* however, when the sea factor is included. As the majority are not found along viewing docks or piers, they rank lower on global connectors, but high on local visibility as they are found in places integral to the spatial structure of the waterfront.

Figure 5.16. Spatial metrics by activity sub-group: travelling (day).



At night, vulnerability is not a deciding factor to this activity type – as the numbers are evenly distributed. The activity does take place in areas with high visual potential, however, with only a small subgroup allocated in areas with low visual potential (probably attributed to those walking back from the pier). Here it is even clearer that this activity subgroup is less influenced by attractors, as the introduction of a weighted model shifts the balance to the left, with the daytime model shifting it back even further. Those travelling then rank low on areas with high *visual value*, but high in areas with high *visibility* in general.

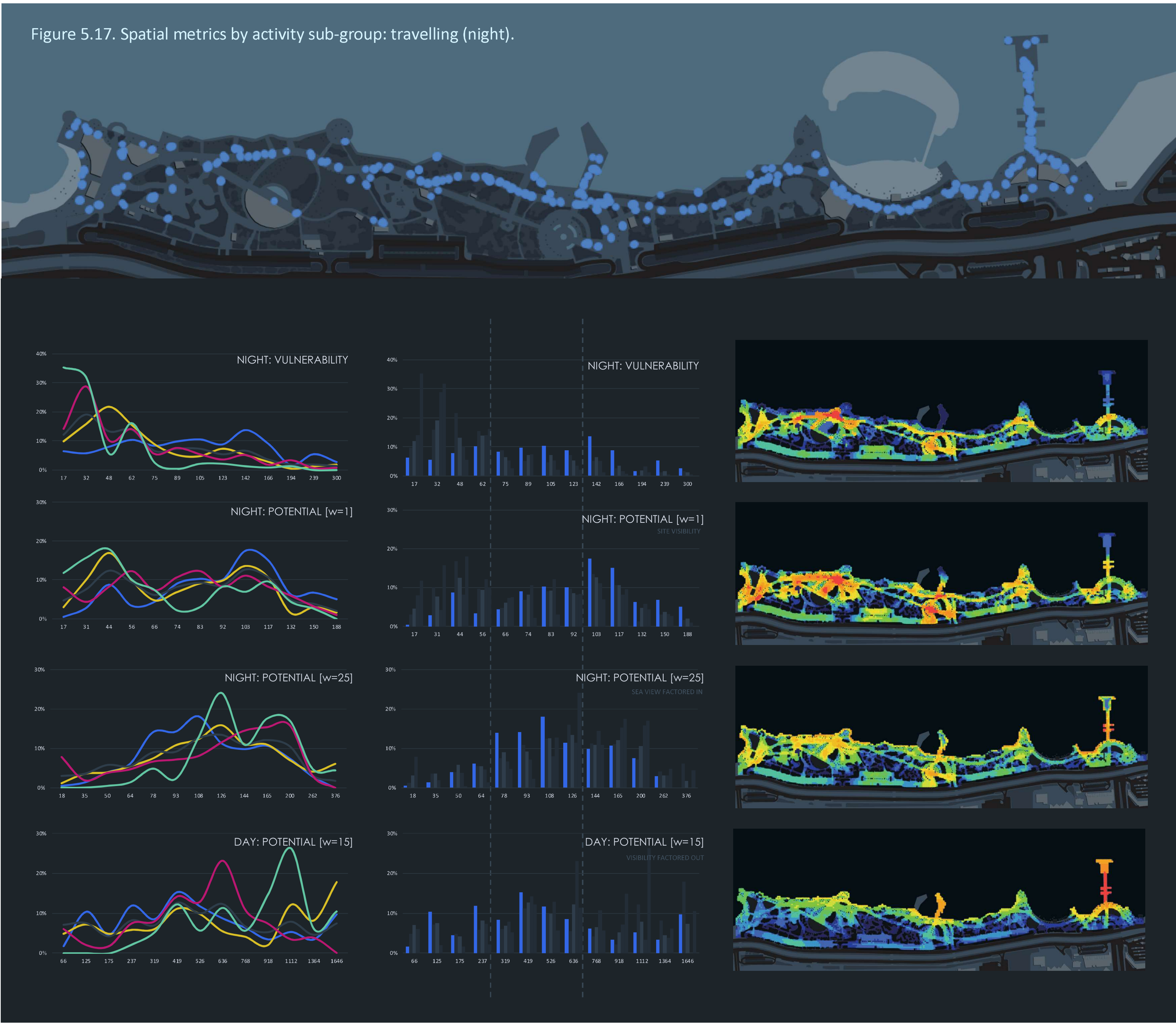


Figure 5.17. Spatial metrics by activity sub-group: travelling (night).

5.5. DISCUSSION

The results raise a couple of discussion points, the first linking back to the theory of attractors, and the second on pedestrian activity and theories on prospect and refuge.

On a macro scale, activity levels seemed highest around the northern area of the waterfront, with an even distribution of services – larger areas in other subsections, equal facilities, maintenance and outside factors, the question remains: why is the general public gravitating towards this particular sub-region? As there are no concrete answers the best speculative response, based on the activity level seen on a micro and meso scale, would be the sea.

On a smaller scale, the sea is clearly driving pedestrian movement, while some activity subgroups are less influenced than others, in a more total sense – pedestrian distribution alone the waterfront is affected by the sea. Areas too far from the waterfront edge are dead zones, despite being fully serviced and well maintained. This study, then, adds to the literature supporting the role of attractors in driving pedestrian movement. As such it is fair to speculate, that even on a broader more macro level, the sea acts as an attractor as well. The one thing available in the three regions (the Sand strip, the Pearl strip, and the Fisherman strip) that is not available in all others is direct contact with the sea. While all other strips have viewing docks, the Sand strip (now closed) has the most direct access, as it is a beach that allows swimming and water activities. The Pearl strip and Fisherman strip both have extending piers allowing for a more immersive seaside experience. Additionally, these piers are almost always filled with pedestrians – with many of the Snapchat posts featuring content from on pier (Fisherman strip).

In comparison, the least active spot (Nawras strip) has a historical landmark and large landscaped areas, but its area to perimeter ratio is large. It is less about the waterfront and more a recreational space – and similarly to the highly integrated spaces within the study area, it is at times neglected. Additionally, the Nawras strip neighbours a roundabout, is exposed to the main road, and is linked to several other parts (Nawras Resort, nearby park, and the old Corniche waterfront extensions, south of the renovated Jeddah Waterfront). It is too exposed. The theory of prospect and refuge might be the most applicable in Saudi Arabia, a country that values privacy above all. In that sense, the Nawras strip, and the neighbouring Shell strip are all low ranking in the prospect of refuge – with direct access to the busy main road.

The idea of a place being too exposed might also be a factor in why the paved plaza, ranking high on all spatial metrics, failed to attract footfall. It is also directly open to the main road – and extremely well-lit at night. As the different activity subgroups often have different defining spatial characteristics, for a space to succeed in the waterfront region, it has to satisfy the needs of at least one of the defined subgroups (unless it is a playground, which creates and attracts its own activity subgroup). While those resting and travelling are less extreme on the scale of wanting prospect and refuge, observers and settlers are very much aligned with the individual concepts, separated. Where observers seem to seek visual vistas and prospect, despite a lack of refuge at times. Their spatial distribution is not dependant on how seen they are, only on what they can see. On the opposite end of the scale are the settlers. The driving force is refuge, seeking areas of low visual vulnerability, near obstacles that shield them from the vison of others. Even creating barriers where there are none (unconventional use of benches), which inaccurately places them on a higher visual set on the map, as the areas are visible, but they are not. They choose spaces based on refuge, despite a lack of prospect at times. They still choose to settle near the sea, so prospect is still of value – but refuge comes first. Areas like the paved plaza, then fail because they are too far from the edge for observers, too exposing for those wishing to settle, and have to external activity to attract a different set of visitors that would then create a new dynamic of engagement.

CHAPTER 6
CONCLUSION

6.1. SUMMARY

In conclusion, this research aimed to study the pedestrian patterns along the Jeddah Waterfront, specifically in relation to spatial characteristics of the waterfront. To do so, several custom tools had to be developed. This included a code to gather and interpret Snapchat heat map data, a code to interpret data from Google traffic, and a custom graph builder used to accurately model the waterfront visibility during the day and night. These tools, along with the standard DepthMap tools and observational methods, were then used to analyse both the waterfront as a space, and how visitors interacted with the different waterfront regions. As per the research findings, the Pearl and Fisherman strip were selected as the core areas of this study due to their high levels of pedestrian activity. These areas were then visited at 6 am and 12 am respectively, thrice during the day, and twice during the night for snapshot observations. The spatial characteristics of the area were mapped out pointing out areas with high potential for interaction and highly integrated routes of movement. While locally, pedestrians frequented the integrated areas (smaller scale), globally integrated regions were neglected for the most part, globally accessible paths as well. Pedestrian movement was found to be influenced by the sea as an attractor, and less by spatial characteristics of the waterfront on a global level. However, on a micro scale – pedestrian subgroups acted differently, with some groups responding more to attractors, some groups to spatial characteristics, and others to a mixture of both. Specifically, those engaged in observing were driven mostly by attractors, despite spatial characteristics; those engaged in settling prioritised areas with low visual vulnerability, only slightly influenced by the sea as an attractor; those engaged in resting consisted of two separate subgroups – some acting as observers and some acting as travellers; and those travelling were driven by waterfront spatial characteristics and not attractor points.

6.2. DESIGN RECOMMENDATIONS

In conclusion, for a more optimized use of space – future waterfront design should take the following into consideration:

1. Create more extruding observational decks and piers, macro traffic is driven by the promise of direct access to sea.

2. Implement shading devices across walkways. The waterfront is only active during the night and very early morning, as the heat drives away visitors. The Jeddah Waterfront initiative is a mega-project, costing hundreds of millions, for it to then be vacant every day from around 10 am to 5 pm is inefficient and could be amended by the addition of canopies.

3. Design more linear forms. The most unused areas of the waterfront have been those distant from the waterfront edge. Creating more linear parks increases the use of the individual spaces, areas further away can be redirected to serve different functions.

4. Create more activity points. While the sea was a major attractor driving pedestrian movement, playgrounds were not affected – where playground close to the sea and distant from the sea had an equal distribution of visitors throughout the day. More activity points create local attractors that could revitalise the area.

5. Allocate areas of visual refuge for those looking to settle. Custom benches that act as short walls could be allocated in grassy areas alongside the existing seating areas – where they can serve either subgroup.

6.3. LIMITATIONS AND FUTURE RESEARCH

This research faced many modelling limitations; while creating the adapted visibility model, many educated guesses had to be made due to limited literature on several topics. These topics include pedestrian visibility at night and how far the average pedestrian can see during the day. As such site visits were used to estimate a base radius for the visibility models. Similarly, defining lit and dim areas was done by a set of created standards based on visual experiences. However, these values are adjustable, and a new graph could easily be created with adjusted values if new literature on pedestrian visibility metrics arises.

The most pressing limitation, within the course of this research, however, was time. The implementation of the obstacle hit system described in the methodology chapter was left out but might be elaborated upon in future research. The system could be used to assess the myriad of vacant kiosks for sale at the waterfront to assess which kiosk has the most spatially attractive qualities, what subgroup each object would be seen by the most, and how that would affect the subsequent visitor interaction. Alternatively, future work could expand upon the current research by increasing the number of cases – testing out the visibility model on several waterfronts to see whether all waterfronts have a similar attractor effect. Finally, more in depth work on the current waterfront could add more insight into the pedestrian patterns – with movement traces and more detailed study of pedestrian isovists.

REFERENCES

Alawi, G., Mostafa, L. (2019). Users' Satisfaction in City Waterfront: The Case of Jeddah Corniche. *The Academic Research Community Publication, 3*(2), 100. https://doi.org/10.21625/ARCHIVE.V3I2.504

AlGhamdi, N., AlRajebah, N., AlMegren, S. (2019). Crowd Behavior Analysis Using Snap Map: A Preliminary Study on the Grand Holy Mosque in Mecca [Poster Abstract]. 22nd ACM Conference on Computer-Supported Cooperative Work and Social Computing, Texas, USA.

Ali, A., Ameur, F. (2018). An Assessment of Jeddah's Hydraulic Protection and Management Systems of Flood. *Open Access Library Journal, 5*(2).

Arab News. (2018). The Place: Jeddah Waterfront. *Arab News*. https://www.arabnews.com/node/1379451/saudi-arabia

Arab News. (2021). Hajj 2021: How Jeddah Earned its Reputation as the City of Hospitality. *Arab News*. https://www.arabnews.com/node/1895921/saudi-arabia

Araguez, M., Psarra, S. (2017). Spatial and Social Patterns of an Urban interior: The Architecture of SANAA. 10th International Space Syntax Symposium, London, UK.

Baarma, B. (2017). A Brand New Corniche Experience. *Destination Jeddah*. https://destinationksa.com/the-jeddah-waterfront/

BBC News. (2019). Saudi Arabia to open up to foreign tourists with new visas. *BBC News*. https://www.bbc.com/news/business-49848068

Belarem, M., Hamza, M., Jamil, A., Ajmi, M. (2018). Mapping and Analysis of the School Network of Makkah Al-Mukarramah (Saudi Arabia), Jeddah Girls' Secondary Schools as Example. *Current Urban Studies, 6*(1), 102-120.

Benedikt, M. L. (1979). To Take Hold of Space: Isovists and Isovist Fields. In: 6.1. doi 10.1068/b060047. URL: http://dx.doi.org/10.1068/b060047.

Botta, F., Moat, H., Preis, T. (2020). Measuring the Size of a Crowd Using Instagram. *Urban Analytics and City Science, 47*(9), 1690–1703. https://doi.org/10.1177/2399808319841615

Dosen, A. S., Ostwald, M. J. (2013). Prospect and Refuge Theory: Constructing a critical Definition for Architecture and Design. *International Journal of Design in Society, 6*.

Dosen. A. S., Ostwald, M. J., (2012). Testing Prospect-Refuge Theory: A Comparative Methodological Review. In The 46th Annual Conference of the Architectural Science Association.

Douglas, M. (1991). The Idea of a Home: A Kind of Space. *Social Research, 58*(1), 287–307.

Dwimirnani, P. (2016). Space Syntax at Night: Measuring the Spatial Configuration in Green Open Spaces by the Influence of Artificial Light (Master's Thesis, UCL, London).

Figueiredo, L., Amorim, L. (2005). Continuity Lines in the Axial System [Paper Presentation]. 5th International Space Syntax Symposium, Delft, Holland.

Fotios, S., Unwin, J., Farrall, S. (2015). Road Lighting and Pedestrian Reassurance After Dark: A Review. *Lighting Research & Technology, 47*, 449–469. https://doi.org/10.1177/1477153514524587

Francis, M. (2020). The Saudi Seasons: A Useful Overview. *Inside Saudi*. https://insidesaudi.com/the-saudi-seasons-a-useful-overview/#2_The_Jeddah_Season-_2_months

Gulf News. (2017). New Jeddah Waterfront Open to Visitors. *Gulf News*. https://gulfnews.com/world/gulf/saudi/new-jeddah-waterfront-open-for-visitors-1.2133433

Hacar, O., Gulgen, F., Bilgi, S. (2020). Evaluation of the Space Syntax Measures Affecting Pedestrian Density Through Ordinal Logistic Regression Analysis. *International Journal of GeoInformation, 9*(10), 589. https://doi.org/10.3390/ijgi9100589

Helmy, M. (2017). Artscaping in public places: Jeddah, the city of urban art. *Street Art and Urban Creativity, 3*. 42-52.

Hillier, B. (1996). Space is the Machine: A Configurational Theory of Architecture. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511597237

Hillier, B., Hanson, J. (1984). The Social Logic of Space. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511597237

Hillier, B., Iida, S. (2005). Network and Psychological Effects in Urban Movement. In: Cohn, A.G., Mark, D.M., (eds.) Proceedings of Spatial Information Theory: International Conference.

Hillier, B., Penn, A., Hanson, J., Grajewski, T., Xu, J. (1993). Natural Movement: Or Configuration and Attraction in Urban Pedestrian Movement. *Environment and Planning B: Planning and Design, 20*, 29-66.

Jahromi, M., Jafari, A. Rafiee, S., Mohtasebi, S. (2007). A Survey on Some Physical Properties of the Date Palm Tree. *Journal of Agricultural Technology, 3*(2), 317-322.

Jeddah Municipality. (2019). Jeddah Waterfront. *Jeddah Municipality*. https://www.jeddah.gov.sa/Projects/JW/index.php

Juhasz, L. Hochmair, H. (2018). Analyzing the Spatial and Temporal Dynamics of Snapchat [Paper Presentation]. VGI-ALIVE Conference, Lund, Sweden.

Krimly, R. (2019). 'Jeddah Season' festival offers entertainment, unique experiences to visitors. *AlArabiya News*. https://english.alarabiya.net/features/2019/06/11/-Jeddah-Season-festival-offers-entertainment-unique-experiences-to-visitors

Kwon, J., Sailer, K. (2015). Seeing and Being Seen Inside a Museum and a Department Store: A Comparison Study in Visibility and Co-Presence Patterns [Paper Presentation]. 10th International Space Syntax Symposium, London, UK.

McLane, Y., Kozints, N. (2019). Spatiality, Experiences, and The Formation of Place Attachment at Campus Student Life Centers. College Student Journal, 53(1), 78-98.

Monokrousou, K., Giannopoulou, M. (2016). Interpreting and Predicting Pedestrian Movement in Public Space Through Space Syntax. Social and Behavioral Sciences, 223, 509 – 514.

Mostafa, L. (2017). Urban and Social Impacts of Waterfronts Development, Case Study: Jeddah Corniche. Procedia Environmental Sciences. 37. 205-221. 10.1016/j.proenv.2017.03.035.

Omnicore. (2021). Snapchat by the Numbers: Stats, Demographics & Fun Facts. *Omnicore*. https://www.omnicoreagency.com/snapchat-statistics/

Peponis, J. (1993). Evaluation and Formulation in Design. Nordisk Arkitekturforskning.

Peponis, J., Dalton, R., Wineman, J., Dalton, N. (2004). Measuring the Effects of Layout Upon Visitors' Spatial Behaviours in Open Plan Exhibition Settings. Environment and Planning B: Planning and Design, 31, 453 – 473.

Psathiti, Chrystala & Sailer, K. (2017). A Prospect-Refuge Approach to Seat Preference: Environmental psychology and spatial layout[Paper Presentation]. 11th Space Syntax Symposium, Lisbon, Portugal.

Ramírez, Y. (2020). Developing a Traffic Congestion Model based on Google Traffic Data: A Case Study in Ecuador. Conference: 6th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS).

Rohloff, I., Psarra, S., Wineman, J. (2009). Experiencing Museum Gallery Layouts through Local and Global Visibility Properties in Morphology [Paper Presentation]. 7th Space Syntax Symposium, Stockholm, Sweden.

Sailer, K. (2007), Movement in Workplace Environments – Configurational or Programmed?. In: Kubat, A. S., Ertekin, Ö., Güney, Y. I. and Eyübolou, E. (eds.), Proceedings of the Sixth International Space Syntax Symposium, Istanbul, Turkey.

Sailer, K., Penn, A., (2009). Spatiality and Transpatiality in Workplace Environments. In Koch, D., Marcus, L. and Steen, J. (eds). Proceedings of the Seventh International Space Syntax Symposium, Stockholm, Sweden.

Saudi Gazette. (2017). Saudi Waterfront Project Starts Next Month. *Saudi Gazette.* https://saudigazette.com.sa/article/516862

Saudi Gazette. (2021). Formula 1 Saudi Arabia Grand Prix unveils Jeddah Street Circuit. *Saudi Gazette*. https://saudigazette.com.sa/article/604554/SAUDI-ARABIA/Formula-1-Saudi-Arabia-Grand-Prix-unveils-Jeddah-Street-Circuit

Thabit, A. (2021). The Experience of Non-locals among Locals: A Local Guide's Interview with Expats in Jeddah, KSA. *Local Guides Connect*. https://www.localguidesconnect.com/t5/General-Discussion/The-Experience-of-Non-locals-among-Locals-A-Local-Guide-s/td-p/2951509

Turner, A. (2000). Angular Analysis a Method for the Quantification of Space. *CASA Working Paper Series, 23*.

Turner, A. (2001). "Depthmap: A Program to Perform Visibility Graph Analysis". In: Proceedings of the 3rd International Symposium on Space Syntax, p. 9. URL: http://www.ucl.ac.uk/bartlett/3sss/ papers_pdf/31_turner.pdf.

Turner, A. (2005). Could a Road-Centre Line Be an Axial Line in Disguise? *Proceedings of the 5th International Symposium on Space Syntax*.

Tzortzi, K., Schieck, A. (2017). Rethinking Museum Space: Interaction Between Spatial Layout Design and Digital Sensory Environments [Paper Presentation]. 11th Space Syntax Symposium, Lisbon, Portugal.

Youssef, A., Sefry, S., Pradhan, B., Alfadail, E. (2015). Analysis On Causes Of Flash Flood In Jeddah City (Kingdom Of Saudi Arabia) Of 2009 And 2011 Using Multi-Sensor Remote Sensing Data And GIS. Geomatics, Natural Hazards and Risk, 7(3), 1018-1042.

Zalakeviciute, R., Bastidas, M., Buenano, A., Rybarczyk, Y. (2020). A Traffic-Based Method to Predict and Map Urban Air Quality. *Applied Sciences, 10*(6).

Zhai, Y., Baran, P. (2013). Application of Space Syntax Theory in Study of Urban Parks and Walking [Paper Presentation]. 9th International Space Syntax Symposium, Seoul, South Korea.

Zhai, Y., Baranb, P., Wuc, C. (2018) Can Trail Spatial Attributes Predict Trail Use Level in Urban Forest Park? An Examination Integrating GPS Data and Space Syntax Theory. *Urban Forestry & Urban Greening 29*, 171–182.

Zook, J. (2017). Spatial Schemata in Museum Floorplans [Paper Presentation]. 11th Space Syntax Symposium, Lisbon, Portugal.

APPENDIX

Code created on Java, Eclipse (version Mars.2), to create a flexible, weighted, directional graph for visibility analysis. Base components include a point class, a line class, an obstacle class, a zone class, a grid class, a graph class, and an isovist class – each with their own separate methods and functions. The Graph class adapts the Visual Integration [HH] metrics used in DepthMap, following the calculations by Turner (2001) expanded upon in the paper *Dissecting Visibility Graph Analysis* by Koutsolampros et al. (2019).

I. POINT CLASS

```java
public class Point {

    //Instance Variables
        private double x;
        private double y;
        private int ID;
        private int zoneID;

        private boolean edge;
        private double luminosity; //only applicable at night

        //Constructor
        public Point(double x, double y) //obstacle lines
        {
            this.x = x;
            this.y = y;

            //default
            this.ID = 0;
            this.edge = false;
            this.luminosity = 1;
            this.zoneID = -1;
        }
        public Point(double x, double y, int id) //basic VGA
        public Point(double x, double y, int id, boolean e) //points in adapted VGA grid
        public Point(double x, double y, int id, boolean e, double l) //night version
        {
            this.x = x;
            this.y = y;
            this.ID = id;
            this.edge = e;
            this.luminosity = l;

            //default
            this.zoneID = -1;
        }

        //Getters
        public double x()
        public double y()
        public int ID()
        public int zoneID()
        public boolean edge()
        public double luminosity()
        public String[] info()

        //Setters
        public void setX(double x)
        public void setY(double y)
        public void setID(int id)
        public void setZoneID(int zID)
        public void isEdge(boolean e)
        public void setLuminosity(double l)
```

```
    //Methods
    public void printSummary()
    {
        String[] pointSummary = this.info();
        for(int i=0; i<pointSummary.length; i++)
            System.out.print(pointSummary[i] + "  ");
        System.out.println();
    }
    public void getZone(Zone[] z) //test
    {
        double x = this.x();
        double y = this.y();

        double minX, minY, maxX, maxY;
        int i=0;
        int zone = -1;

        while(i<z.length && zone==-1)
        {
            minX = z[i].x();
            minY = z[i].y();
            maxX = z[i].endX();
            maxY = z[i].endY();

            if (x>=minX && x<maxX && y>=minY && y<maxY)
                zone = z[i].ID();

            i++;
        }

        if (zone==-1)
            System.out.println("Point ("+ this.x + ", " + this.y + ") is not in any listed zone.");

        this.zoneID = zone;
    }
```

II. LINE CLASS

```
public class Line {

    //Instance Variables
        private Point p;
        private Point q;

        private double m; //gradient (if = infinity, line is vertical)
        private double c; //y intercept
        private double length;

        //Constructor
        public Line(Point p, Point q)
        {
            this.p = p;
            this.q = q;
            calculateProperties();

        }

        //Getters
        public Point p()..
        public Point q()..
        public double x1()..
        public double x2()..
        public double y1()..
        public double y2()..
        public double m()..
        public double c()..
        public double length()..
        public String[] info()..

        //Setters
        public void setP(Point p)..
        public void setQ(Point q)..
```

```java
//Methods
public void printSummary()
private void calculateProperties()
{
    //if vertical, let m be infinity and c be the x value
    double px, py, qx, qy;

    px = this.p.x();
    py = this.p.y();
    qx = this.q.x();
    qy = this.q.y();

    if(px==qx){
    this.m = Double.POSITIVE_INFINITY;
    this.c = px;}

    //otherwise get line equation
    else{
    this.m = ((py - qy)/(px - qx));
    this.c = (py - this.m*px);}

    this.length = Math.sqrt( Math.pow((px-qx), 2) + Math.pow((py-qy),2) );
}
public double min(double a, double b)
public double max(double a, double b)
public double intersectionPoint(Line l)
{
    double x;

    if(this.m()==l.m() && this.c()==l.c())
        x = -2;//same line
    else if(this.m()==l.m())
        x = -1;//parallel lines
    else if(this.m()== Double.POSITIVE_INFINITY)
        x = this.c(); //no y intercept, line equation is just x
    else if(l.m()== Double.POSITIVE_INFINITY)
        x = l.c();
    else
    {x = (l.c()-this.c()) / (this.m()-l.m());}

    return x;
}

public boolean includesPoint(double x, double y)
public boolean intersects(Line l)
{
    boolean intersects=false;

    //intersection point
    double x = this.intersectionPoint(l);
    double y;

    if(x==-1)//parallel lines
        return intersects;

    if(x==-2)//same line
        {intersects = true;
        return intersects;}

    if(this.m()!=Double.POSITIVE_INFINITY)
        y = (this.m()*x)+this.c();
    else if(l.m()!=Double.POSITIVE_INFINITY)
        y = (l.m()*x)+l.c();
    else
        {intersects = false; //unnecessary, if both = +inf then x=-1; but just in case...
        return intersects;}

    //find out if (x,y) point of hypothetical intersection lies within the given segments
    if(this.includesPoint(x, y) && l.includesPoint(x, y))
            intersects = true;

    return intersects;
}
```

```java
//Collision Methods
public List<Integer> zones(Zone[] z) //returns list with zones that line crosses
{
    List<Integer> accessedZones = new ArrayList<Integer>();

    //change zone ID from default(-1) to actual zone
    this.p.getZone(z);
    this.q.getZone(z);


    accessedZones.add(this.p().zoneID());

    //if both points lie in same zone
    //then obviously line is contained within single zone
    if(this.p().zoneID()==this.q().zoneID())
        return accessedZones;
    else if(!accessedZones.contains(this.q().zoneID()))
        accessedZones.add(this.q().zoneID());


    Line[] boundaries = new Line[4];
    boolean intersects=false;
    int linePointer;

    for(int i=0; i<z.length; i++){

        //get boundaries of zone[i] and reset counters
        linePointer = 0;
        intersects = false;
        boundaries = z[i].getBoundaries();

        //check if line intersects any of the zone borders
        while(linePointer<boundaries.length && intersects==false)
        {
            intersects = this.intersects(boundaries[linePointer]);
            linePointer++;
        }

        //if it does, add it to the list of zones crossed by line
        if(intersects && !accessedZones.contains(z[i].ID()))
            accessedZones.add(z[i].ID());

    }

    return accessedZones;
}
public boolean collides(Obstacle o)
{

    boolean collides = false;
    List<Line> lines = o.obstacleLines();

    int i = 0;
    while(i<lines.size() && collides==false)
    {
        if(this.intersects(lines.get(i)))
        {
            collides = true;
            o.addHit();
        }
        i++;
    }

    return collides;
}
```

III. OBSTACLE CLASS

```java
public class Obstacle{

    //Instance Variables
        private int ID;
        private List<Line> lines;
        private List<Integer> zoneID;
        private int hits;

        //Constructor
        public Obstacle(List<Line> obsLines, int id)
        {
            this.ID = id;
            this.lines = obsLines;

            //default
            this.hits = 0;
        }

        //Getters
        public int ID()
        public List<Line> obstacleLines()
        public List<Integer> zoneID()
        public int hits()
        public String[] info()

        //Setters
        public void addLine(Line l)
        public void setHits(int h)
        public void addHit()

        //Methods
        public void printSummary()
        public void getZone(Zone[] allZones)
        {

            List<Integer> zoneList = new ArrayList<Integer>();

            for(int i=0; i<this.lines.size(); i++)
            {
                List<Integer> z = this.lines.get(i).zones(allZones);

                for(int j=0; j<z.size(); j++)
                    if(!zoneList.contains(z.get(j)))
                        zoneList.add(z.get(j));
            }

            this.zoneID = zoneList;

        }
```

IV. ZONE CLASS

```java
public class Zone {

    //Instance Variables
        private double startX;
        private double startY;
        private double endX;
        private double endY;
        private double size;
        private int ID;

        private List<Integer> pointList;
        private List<Integer> obstacleList;

        //Constructor
        public Zone(int id, double x, double y, double s)
        {
            this.ID = id;
            this.startX = x;
            this.startY = y;
            this.size = s;
            this.endX = x+s;
            this.endY = y+s;

            //will set both lists later on in grid
        }

        //Getters
        public double x()
        public double y()
        public double endX()
        public double endY()
        public int ID()
        public double size()
        public String[] info()

        public int numberPoints()
        public int numberObstacles()
        public List<Integer> pointList()
        public List<Integer> obstacleList()


        //Setters
        public void setX(double x)
        public void setY(double y)
        public void startingPoint(double x, double y)
        public void setSize(double s)
        public void setPointList(List<Integer> list)
        public void setObstacleList(List<Integer> list)

        //Methods
        public void printSummary()
        public void printPoints()
        public void createPointList(Point[] p)
        {

            List<Integer> pointList = new ArrayList<Integer>();
            for(int i=0; i<p.length;i++)
                if(p[i].zoneID()==this.ID())
                    pointList.add(p[i].ID());

            this.pointList = pointList;
        }
        public void createObstacleList(Obstacle[] o)
        {

            List<Integer> obstacleList = new ArrayList<Integer>();
            for(int i=0; i<o.length;i++)
                if(o[i].zoneID().contains(this.ID()))
                    obstacleList.add(o[i].ID());

            this.obstacleList = obstacleList;
        }
```

```
public Line[] getBoundaries()
{
    double x0 = this.x();
    double x1 = this.endX();

    double y0 = this.y();
    double y1 = this.endY();

    Point a = new Point(x0, y0, -1);
    Point b = new Point(x0, y1, -1);
    Point c = new Point(x1, y0, -1);
    Point d = new Point(x1, y1, -1);

    Line top = new Line(b,d);
    Line bottom = new Line(a,c);
    Line left = new Line(a,b);
    Line right = new Line(c,d);

    Line[] boundingBox = new Line[]{top, bottom, left, right};
    return boundingBox;
}
```

V. GRID CLASS

```
public class Grid {

    //Instance Variables
        private Point[] grid;
        private Zone[] zones;
        private Obstacle[] obstacles;


        //Constructor
        public Grid(Point[] points, Obstacle[] obstacles, int minZones)
        {
            this.grid = points;
            this.zones = createZones(minZones);
            this.obstacles = obstacles;

            assignZoneID();
            assignPointList();
            assignObstacleList();
        }
        public Grid(Point[] points, Obstacle[] obstacles, Obstacle border, int minZones)
        {
            this.grid = points;
            this.zones = createZones(minZones);

            //add border to list of obstacles
            Obstacle[] o = new Obstacle[obstacles.length+1];
            for(int i =0; i<obstacles.length; i++)
                o[i] = obstacles[i];
            o[o.length-1]=border;

            this.obstacles = o;

            assignZoneID();
            assignPointList();
            assignObstacleList();
        }


        //Getters
        public Point[] allPoints()
        public int getNumberPoints()
        public Zone[] allZones()
        public int getNumberZones()
        public Obstacle[] allObstacles()
        public int getNumberObstacles()
        public double[] getZeroPoint()
        public int[] resetHits()
```

```java
//Methods
public double[] gridBounds()
private Zone[] createZones(int minZones)
{
    double[] bounds = this.gridBounds();

    //grid coordinates plus added buffer on end points
    //relevant to Point>>getZones method
    double xMin = bounds[0]-50;
    double yMin = bounds[1]-50;
    double xMax = bounds[2]+50;
    double yMax = bounds[3]+50;

    //getting zone metrics based on user input
    double rangeX = xMax-xMin;
    double rangeY = yMax-yMin;
    double totalArea = rangeX*rangeY;
    double zoneArea = totalArea/minZones;
    double zoneIncrement = Math.sqrt(zoneArea);


    int numberZonesX = (int)Math.ceil(rangeX/zoneIncrement);
    int numberZonesY = (int)Math.ceil(rangeY/zoneIncrement);

    //necessarily larger because current code has square zones (adjust in future)
    int numberZones = numberZonesX*numberZonesY;

    //create zones
    Zone[] allZones = new Zone[numberZones];

    double xPointer = xMin;
    double yPointer = yMin;
    int zonePointer = 0;


    for(int i=0; i<numberZonesY; i++)
    {

        xPointer = xMin;
        for(int j=0; j<numberZonesX; j++)
        {
            Zone zone = new Zone(zonePointer,xPointer,yPointer,zoneIncrement);
            allZones[zonePointer] = zone;

            xPointer = xPointer+zoneIncrement;
            zonePointer++;
        }

        yPointer = yPointer+zoneIncrement;
    }

    return allZones;
}
private void assignZoneID()
{
    //assign zone id to each point
    for(int i=0; i<this.grid.length; i++)
        this.grid[i].getZone(this.zones);

    //assign zone id to each obstacle
    for(int j=0; j<this.obstacles.length; j++)
        this.obstacles[j].getZone(this.zones);

}
private void assignPointList() //assign point list to each zone
{
    for(int i=0; i<this.zones.length; i++)
        this.zones[i].createPointList(this.grid);
}
private void assignObstacleList() //assign obstacle list to each zone
{
    for(int i=0; i<this.zones.length; i++)
        this.zones[i].createObstacleList(this.obstacles);

}
```

VI. GRAPH CLASS

```
public class Graph {

    //Instance Variables
        private Grid base;
        private int wieght;
        private int radiusMax;
        private int radiusMin;

        private int[][] adjacencyMatrix;
        private double[] connectivity;
        private double[] connectivityII;
        private double[] nodeCount;
        private double[] totalDepth;
        private double[] meanDepth;
        private double[] intHH;

        //Constructor
        public Graph(Grid g, int w, int rMax, int rMin)
        {
            this.base = g;
            this.wieght = w;
            this.radiusMax = rMax;
            this.radiusMin = rMin;

            //weight represents the number of other symbolic nodes an edge node represents
            //i.e. if you can see this node, you can also see x others
            //let weight = 1 for regular graph

            //very large max radius = vga graph(n)
            //local measures can be observed by defining small max radius
            //the addition of a minimum radius is optional

            createGraph();
            calculateConnectivity();
        }
        public Graph(Grid g, int w, int rMax)[..]


        //Getters
        public Grid base()[..]
        public int[][] adjacencyMatrix()[..]
        public double[] connectivity()[..]
        public double[] connectivityII()[..]
        public double[] nodeCount()[..]
        public double[] totalDepth()[..]
        public double[] meanDepth()[..]
        public double[] integrationHH()[..]


        //Setters
        public void importMatrix(int[][] adj)[..]

        //Methods
        private void initializeMatrix(int size) [..]
        private int calculateWeight(Point p, Point q)
        {

            int maxDistance = this.radiusMax;

            Line l = new Line(p,q);
            int weight = (int) Math.round((1-(l.length()/maxDistance))*this.wieght);

            if(weight>1)
                return weight;
            else
                return 1;
        }
```

```
private void createGraph()
{
    int numberPoints = this.base.allPoints().length;
    initializeMatrix(numberPoints);

    Point p;

    for(int i=0; i<numberPoints; i++)
    {
        p = this.base.allPoints()[i];
        if(!p.edge())
            createLines(p);
    }
}
private void createLines(Point p)
{
    int numberPoints = this.base.allPoints().length;
    Point q;
    Line l;
    boolean collision = false;

    for(int j=0; j<numberPoints; j++)
    {
        q = this.base.allPoints()[j];

        if(p.luminosity()==1)
        {if(this.adjacencyMatrix[q.ID()][p.ID()]==1)
            this.adjacencyMatrix[p.ID()][q.ID()]=1;}
        //to save time, if lum=1 then daytime visibility
        //any non-edge connection is a two way connection

        else if(p.ID()!=q.ID())
        {
            l = new Line(p,q);
            double r = this.radiusMax*q.luminosity();

            //if daytime this value would equal radiusMax (lum = 1)
            //if night time this value would be a % of radiusMax

            if(l.length()<=r && l.length()>=this.radiusMin)
            {
                collision = collisionDetection(l);

                if(!collision)
                    this.adjacencyMatrix[p.ID()][q.ID()]=1;
            }

        }}
}

}
private boolean collisionDetection(Line l)
{
    //references
    List<Integer> zones = l.zones(this.base.allZones());
    List<Integer> obstacleList;

    //return value
    boolean collision = false;

    //counters
    int zoneIndex = 0;
    int obstacleIndex;
    List<Integer> obstacleID = new ArrayList<Integer>();
    //so you don't check the same obstacle twice if it overlaps two/more zones

    //detect collision
```

```
        //detect collision
        while(collision==false && zoneIndex<zones.size()){

            obstacleList = this.base.allZones()[zones.get(zoneIndex)].obstacleList();
            obstacleIndex=0;

            while(collision==false && obstacleIndex<obstacleList.size()){

                    if(!obstacleID.contains(obstacleList.get(obstacleIndex)))
                    //if you haven't already checked this obstacle
                    {
                        collision = l.collides(this.base.allObstacles()[obstacleList.get(obstacleInde
                        obstacleID.add(obstacleList.get(obstacleIndex));
                    }
                        obstacleIndex++;}

            zoneIndex++;}

        return collision;
    }




    //Graph Traversal
    public int[] BFS(int source)
    {
        int[] pointStats = new int[2]; //return node count and total depth

        //initialize arrays
        boolean[] visited = new boolean[this.base.allPoints().length];
        int[] steps =  new int[visited.length];
        for(int i=0; i<steps.length;i++)
            steps[i]=-1; //assuming unweighted or no negative weights

        List<Integer> queue = new ArrayList<Integer>();
        List<Integer> stepCount = new ArrayList<Integer>(); //to keep track of how many steps away noc

        int d = 0; //current distance from real source node
        visited[source]=true;
        steps[source]=0;

        queue.add(source);
        stepCount.add(d);

        while(!queue.isEmpty())
        {
            source = queue.remove(0); //first in first out
            d = stepCount.remove(0); //initialize to how far current node is

            for(int i=0; i<visited.length; i++) //visited length used as shortcut for size of all poir
                if(this.adjacencyMatrix[source][i]>0 && !visited[i])//connected and unvisited
                {
                    queue.add(i);
                    stepCount.add(d+1);// + one extra step to get to node i

                    visited[i]=true;
                    steps[i]=d+1;

                }
        }

        //total node count
        int sum=0;
        for(int i=0; i<visited.length; i++)
        {   if(visited[i]==true)
            {
                if(this.base.allPoints()[i].edge())
                    sum=sum+this.wieght;
                else
                    sum++;
            }
        }

        pointStats[0]=sum;
```

```java
    //total depth
    int totalDepth=0;
    for(int i=0; i<steps.length; i++)
    {
        if(steps[i]!=-1)
        {
            if(this.base.allPoints()[i].edge())
                totalDepth=totalDepth+(steps[i]*this.wieght);
            else
                totalDepth=totalDepth+steps[i];
        }
    }
    pointStats[1]=totalDepth;

    return pointStats;
    }

public void printGraph(int[][] graph)
public void calculateConnectivity()
{
    int nodes = this.base().allPoints().length;

    double[] connectivity_I = new double[nodes];
    double[] connectivity_II = new double[nodes];

    //get connectivity I (how much each point can see)
    int sum;
    for(int i=0; i<nodes;i++)
    {
        sum = 0;
        for(int j=0;j<nodes;j++)
        {
            if(this.adjacencyMatrix[i][j]==1)
            {
                if(this.base.allPoints()[j].edge())
                {
                    int w = calculateWeight(this.base.allPoints()[i], this.base.allPoints()[j]);
                    sum=sum+w;
                }
                else
                    sum++;
            }
        }
        connectivity_I[i]=sum;
    }

    //get connectivity II (how 'seen' each point is)
    for(int i=0; i<nodes;i++)
    {
        sum = 0;
        for(int j=0;j<nodes;j++)
            if(this.adjacencyMatrix[j][i]==1)
                    sum++;

        connectivity_II[i]=sum;
    }

    this.connectivity = connectivity_I;
    this.connectivityII = connectivity_II;
}
```

```java
public void calculateProperties() //all done with assumption of nodeCount>2 at least
{
    int nodes = this.base().allPoints().length;

    double[] connectivity = this.connectivity;

    double[] totalDepth = new double[nodes];
    double[] totalNodeCount = new double[nodes];
    double[] meanDepth = new double[nodes];
    double[] intHH = new double[nodes];


    //get total node count, total depth, and mean depth
    int[] pointProperties;
    for(int i=0; i<nodes; i++)
    {
        if(!this.base().allPoints()[i].edge())
        {
            pointProperties = BFS(i);
            //System.out.println("BFS for node " + i + " done.");

            totalNodeCount[i] = pointProperties[0];
            totalDepth[i] = pointProperties[1];
            meanDepth[i] = totalDepth[i]/(totalNodeCount[i]-1);
        }
        else
        {
            totalNodeCount[i] = 0;
            totalDepth[i] = 0;
            meanDepth[i] = 0;
        }
    }



    double[] dValue = new double[nodes];
    double[] totalRA = new double[nodes];
    double[] totalRAA = new double[nodes];

    //get visual integration
    for(int i=0; i<nodes; i++)
    {
        if(!this.base().allPoints()[i].edge())
        {

            double a = 2*(totalNodeCount[i]*( log2((totalNodeCount[i]+2)/3)-1)+1);
            double b = ((totalNodeCount[i] - 1)*(totalNodeCount[i] - 2));
            dValue[i] = a/b;
            totalRA[i] = 2*(meanDepth[i] - 1)/(totalNodeCount[i] - 2);
            totalRAA[i] = totalRA[i]/dValue[i];

           if(totalRAA[i]>0)
               intHH[i]=1/totalRAA[i];
           else
               intHH[i]=0;
       }
       else
       {
           dValue[i] = 0;
           totalRA[i] = 0;
           totalRAA[i] = 0;
           intHH[i]=0;
       }
   }


   //assign values
   this.connectivity = connectivity;
   this.nodeCount = totalNodeCount;
   this.totalDepth = totalDepth;
   this.meanDepth = meanDepth;
   this.intHH = intHH;
}
```

VII. ISOVIST CLASS

```
public class Isovist {

        //Instance Variables
            private Grid base;
            private List<Point> visitors;
            private int[] count;
            private int maxRadius;

        //Constructor
            public Isovist(Grid g, List<Point> snapshots, int r)
            {
                this.base = g;
                this.visitors = snapshots;
                this.maxRadius = r;

                assignZoneID();
                initializeCount();
                createGraph();
            }


        //Getters
            public int[] getCount()..

        //Methods
            public void assignZoneID()..
            public void initializeCount()..
            private void createGraph()
            {
                int numberPoints = this.visitors.size();
                Point p;

                for(int i=0; i<numberPoints; i++)
                {
                    p = this.visitors.get(i);
                    createLines(p);
                }
            }
            private void createLines(Point p)
            {
                int numberPoints = this.base.allPoints().length;
                Point q;
                Line l;
                boolean collision = false;

                for(int j=0; j<numberPoints; j++)
                {
                    q = this.base.allPoints()[j];
                    l = new Line(p,q);
                    double r = this.maxRadius*q.luminosity();

                    //if daytime this value would equal radiusMax (lum = 1)
                    //if night time this value would be a % of radiusMax

                        if(l.length()<=r)
                        {
                            collision = collisionDetection(l);

                            if(!collision)
                                this.count[j]++;
                        }
                }
            }
```

```
private boolean collisionDetection(Line l)
{

    //references
    List<Integer> zones = l.zones(this.base.allZones());
    List<Integer> obstacleList;

    //return value
    boolean collision = false;

    //counters
    int zoneIndex = 0;
    int obstacleIndex;
    List<Integer> obstacleID = new ArrayList<Integer>();
    //so you don't check the same obstacle twice if it overlaps two/more zones


    //detect collision
    while(collision==false && zoneIndex<zones.size())
    {
        obstacleList = this.base.allZones()[zones.get(zoneIndex)].obstacleList();
        obstacleIndex=0;

        while(collision==false && obstacleIndex<obstacleList.size())
            {

                if(!obstacleID.contains(obstacleList.get(obstacleIndex)))//if you haven't al
                {
                    collision = l.collides(this.base.allObstacles()[obstacleList.get(obstacl
                    obstacleID.add(obstacleList.get(obstacleIndex));
                }
                obstacleIndex++;
            }


        zoneIndex++;
    }

    return collision;

}
```

VIII. MAIN CLASS [SAMPLE]

```java
public class Run {

    //read file
        public static List<String> readTextFileByLine(String fileName) throws IOException
        public static int[] getIntData(String fileName, int columnNumber) throws IOException
        public static double[] getDoubleData(String fileName, int columnNumber) throws IOException
        public static boolean[] getBoolData(String fileName, int columnNumber) throws IOException
        public static char[] getCharData(String fileName, int columnNumber) throws IOException

    //write file
        public static void initializeColumn(int[] data, String header, String fileName) throws IOExcepti
        public static void addColumn(int[] data, String header, String fileName) throws IOException
        public static void addColumn(double[] data, String header, String fileName) throws IOException

    //analysis
        public static Point[] createPoints(String pointFile, boolean day) throws IOException
        {
            int[] id = getIntData(pointFile, 0);
            double[] x = getDoubleData(pointFile, 1);
            double[] y = getDoubleData(pointFile, 2);
            boolean[] e = getBoolData(pointFile, 3);
            double[] lum = getDoubleData(pointFile, 4);

            Point[] pointList = new Point[id.length];

            for(int i=0; i<id.length; i++)
            {   if(day)
                    pointList[i] = new Point(x[i],y[i],id[i], e[i]);
                else
                    pointList[i] = new Point(x[i],y[i],id[i], e[i], lum[i]);
            }

            return pointList;
        }


        public static Obstacle[] createObstacles(String obstacleFile) throws IOException
        {
            int[] allIds = getIntData(obstacleFile,0);
            double[] pX = getDoubleData(obstacleFile, 1);
            double[] qX = getDoubleData(obstacleFile, 2);
            double[] pY = getDoubleData(obstacleFile, 3);
            double[] qY = getDoubleData(obstacleFile, 4);

            //get unique ids
            List<Integer> uniqueIDs = new ArrayList<Integer>();
            for(int i=0; i<allIds.length; i++)
                if(!uniqueIDs.contains(allIds[i]))
                    uniqueIDs.add(allIds[i]);

            Obstacle[] obstacleList = new Obstacle[uniqueIDs.size()];

            int linePointer = 0;
            for(int i=0; i<uniqueIDs.size(); i++)
            {
                int currentID = uniqueIDs.get(i);
                List<Line> obstacleLines = new ArrayList<Line>();

                while(linePointer<allIds.length && allIds[linePointer]==currentID)
                {
                    Point p = new Point(pX[linePointer],pY[linePointer]);
                    Point q = new Point(qX[linePointer],qY[linePointer]);
                    Line l = new Line(p,q);
                    obstacleLines.add(l);
                    linePointer++;
                }

                obstacleList[i] = new Obstacle(obstacleLines,currentID);
            }
```

```java
public static void main(String[] args) throws IOException {

    long start = System.nanoTime();

    //************************************************** INITIALIZATION ****************************
    //define main file paths
    String pointFile = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaPoints.csv";
    String obstacleFile = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaObstacles.csv";
    String obstacleHits = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaObstacleHits.csv";

    String pointFileAll = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaAll.csv";
    String pointFileIsovist = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaIsovist.csv";

    //define grid points
    Point[] pointList_day = createPoints(pointFile, true);
    Point[] pointList_night = createPoints(pointFile, false);

    //define obstacles
    Obstacle[] obstacleList = createObstacles(obstacleFile);

    //define site border
    String borderFile = "/Users/Lubaba/Desktop/UCL/Dissertation/Data/VGA/csv/vgaBorders.csv";
    Obstacle border = createSingleObstacle(borderFile);

    //create grids
    Grid day = new Grid(pointList_day, obstacleList, 3);
    Grid day_border = new Grid(pointList_day, obstacleList, border, 3);
    Grid night = new Grid(pointList_night, obstacleList, 3);


   //create graph: classic
   Graph classicVGA; //to ensure program is on the right track
   classicVGA = new Graph(day_border,1,1000);
   addColumn(classicVGA.connectivity(), "classic_1000", pointFileAll);
   addColumn(classicVGA.integrationHH(), "int_classic", pointFileAll);

   //create graph: adapted
   Graph adaptedVGA;

   adaptedVGA = new Graph(day,1,25);
   addColumn(adaptedVGA.connectivity(), "radius_25", pointFileAll);

   adaptedVGA = new Graph(day,1,50);
   addColumn(adaptedVGA.connectivity(), "radius_50", pointFileAll);

   adaptedVGA = new Graph(day,1,75);
   addColumn(adaptedVGA.connectivity(), "radius_75", pointFileAll);

   adaptedVGA = new Graph(day,1,100);
   addColumn(adaptedVGA.connectivity(), "radius_100", pointFileAll);

   adaptedVGA = new Graph(day,1,150);
   addColumn(adaptedVGA.connectivity(), "radius_150", pointFileAll);
   addColumn(adaptedVGA.integrationHH(), "int_150", pointFileAll);

   adaptedVGA = new Graph(day,1,200);
   addColumn(adaptedVGA.connectivity(), "radius_200", pointFileAll);
   addColumn(adaptedVGA.integrationHH(), "int_200", pointFileAll);

   adaptedVGA = new Graph(day,1,500);
   addColumn(adaptedVGA.connectivity(), "radius_500", pointFileAll);

   adaptedVGA = new Graph(day,1,750);
   addColumn(adaptedVGA.connectivity(), "radius_750", pointFileAll);

   adaptedVGA = new Graph(day,1,1000);
   addColumn(adaptedVGA.connectivity(), "radius_1000", pointFileAll);
```