SCHOOL OF SCIENCE
AND TECHONOLOGY

Final Year Project

Indoor Map Construction and Navigation

by

Marcus Wong Yew Hon 15024490

BSc (Hons) in Computer Science

Supervisor: Dr. Chia Wai Chong

Date: 16 November 2018

Abstract

GPS navigation systems for outdoor navigation has been around for decades. However, indoor navigation systems are still in its developmental stages when compared to its outdoor counterpart. Conventional positioning techniques for indoor navigation requires prior preparation and installation of devices for buildings to be navigation ready. An indoor navigation system approach without prior knowledge or device requirements was proposed and implemented in this paper. Users of the system need only a taken photo of a directory map and directory lot listing for navigation to take place. The system is achieved through implementation of perspective correction, text recognition, lot segmentation with labelling, and routing techniques.

KEYWORDS – Indoor Navigation, OCR Text Extraction, Perspective Correction, A* Routing Algorithm, Lot Segmentation.

Table of Contents

List of Figures

List of Tables

1. Introduction

Advancements made during the last decade in navigational technology is nothing short of amazing with the advent of smartphones and the concept of ubiquitous computing. Navigation from one place to another has never been easier with services such as Google Maps and Waze. However, consumers are now more likely to lose their way within large enclosed buildings such as stadiums, educational institutes and shopping malls. This is because GPS signals are not accurate nor reliable enough for indoor navigation usage. Several solutions and methods have been used to combat the difficulties faced by indoor navigation. One of them includes installation of beaming devices all around indoor establishments that sends signals within its maximum radius range. Mobile devices are then used to receive said signals and broadcast said signals back to the beaming devices to pinpoint the exact location of user within indoor establishments in real time. While this method is effective, it presents various inconveniences and cost incurred just to install the required beaming devices. A similar approach to the beaming devices can be applied to WIFI signals as well and is preferable to beaming devices as WIFI are usually readily available in most indoor establishment nowadays. However, the above approaches require the users to be within range of the broadcasted signals or WIFI in order for indoor navigation to be performed. Besides, having indoor navigation to require pre-installed beaming devices and WIFI signals means that not all indoor establishments are navigation-ready. For example, a shopping mall that have not prepared a mobile application for indoor navigation or installed any beaming devices for indoor navigation means that no indoor navigation can be done by users.

Thus, a new approach to indoor navigation devoid of signal requirements is needed for users to navigate through their intended location indoors without relying on external means to pinpoint their location. As most large indoor establishments have a floor plan map and directory, it could be used as the base for the establishment's indoor navigation. Users could simply take a photo of an establishment's floor map and directory to provide the necessary information needed for navigation to take place. The taken floor map and directory images can then be processed to produce a navigation-ready interactive map for the user to perform indoor navigation. This approach allows users to navigate through any indoor establishments without needing special preparations (beaming devices installation, custom applications) from establishment owners.

2. Literature Review

There are not many reading materials available in regards of our indoor navigation system in general. However, there are major components in our indoor navigation system that are thoroughly discussed in its respective literatures such as routing algorithms, perspective correction and floorplan lot segmentation with lot labelling that can be studied and reviewed.

Routing algorithms are used to calculate the shortest possible distance from a target location to a destination location in a map. Floorplan lot segmentation is needed for future operations such as directory labelling and routing nodes placement to be implemented into the system. Lot labelling is required for our system to link each physical location names from a directory into the individual lot segments in a floorplan for navigation purposes. Lastly, perspective correction of floorplan images must be implemented to provide a more accurate and precise base of operations for all the above stated components to operate properly.

2.1 Routing Algorithms

Dijkstra's algorithm is a pathfinding algorithm that finds the shortest path between points, also known as nodes, in a graph [1]. It was concocted by Edsger W. Dijkstra in 1956 and remains one of the most popular shortest path algorithms to date.

The basic idea of Dijkstra's algorithm are as follows: in Dijkstra's algorithm, there will be a starting node where the algorithm begins which will be called a source node and an ending node which will be called a destination node. All nodes besides the source node are labelled with its respective distance value from the source node. A value of infinity will first be labelled for all unvisited neighboring nodes of the source node and a value of 0 will be assigned to the source node. This is not an indication of infinite distance, but to note that those nodes remain to be visited. Dijkstra's algorithm creates and maintains sets of nodes called visited vertex and unvisited vertex. While traversing through the nodes, all unvisited neighboring nodes of the current node will be considered and have its tentative distance calculated simultaneously. The calculated distance value will then be compared to the current node's labelled distance value and have it replaced if the calculated distance is of smaller value to the labelled distance value. After all unvisited neighboring nodes have been calculated and considered, the current node the algorithm is traversing will be added into the visited vertex set and removed from the unvisited vertex set. The algorithm will then proceed by selecting the next unvisited node with the least distance value as the new current node and the steps above will then be repeated for it. The algorithm will only stop when the destination node has already been added in the visited vertex set or when the smallest calculated distance in the unvisited vertex set is infinity (this will only occur when there are no connections that exists between the source node and its remaining unvisited nodes after a full traversal). [2]

While Dijkstra's algorithm was designed to be simple to understand [4] and is easy to implement, it is important to note that the algorithm does have its deficiencies in terms of performance and memory requirements. [5][6] This is due to the nature of Dijkstra's algorithm where all the nodes are considered and processed at some point even when it has little to no contribution in finding the shortest path to the destination node. This causes an increase not just in processing power but memory requirements as well [6].

The A* algorithm on the other hand, aims to correct and improve upon the disadvantages present in Dijkstra's algorithm. The A* algorithm is also a shortest path finding algorithm that is seeing widespread usage in routing applications due to its impressive accuracy and performance. The A* algorithm is an extension of Dijkstra's algorithm with the extension being a heuristic function being incorporated into its distance calculating operations on nodes. [6][7] Unlike Dijkstra's algorithm, it serves to provide the most optimal and acceptable short path with less time taken provided a good heuristic function is being used. [7]

A* algorithm works by traversing through all possible nodes and routes to the destination node for the path that has the least incurring cost while maintaining a priority queue of visited nodes. It does so by considering the routes that *seems* to lead to the solution the quickest. This is where the heuristic component of the algorithm comes to play.

$$f(n) = g(n) + h(n)$$

Above illustrates the equation used by the A* algorithm to determine the order of the nodes in a priority queue to be used for traversing later. The equation can be interpreted as two separate parts, where *g(n)* denotes the cost of a path from the source node to *n*, the destination node, and *h(n)* denotes the heuristic function that allows the algorithm to make estimates of the least cost path from the destination node, *n*, to the source node. However, the heuristic function must be admissible, whereby the actual cost of the path will not be overestimated to reach the destination node, for the algorithm to properly determine the shortest route. [6] The A* algorithm will create two tables namely OPEN and CLOSE, whereby OPEN records all currently inaccessible but detected nodes in the graph and CLOSE records all visited nodes. The contents of the OPEN table are sorted through the heuristic function *h(n)* and the algorithm selects the least costing node to traverse one at a time, to arrive at the destination node. [7]

Due to the heuristic function incorporated into A* algorithm, it is theoretically more efficient compared to Dijkstra's algorithm as it explores and traverses through lesser nodes to reach the destination node. However, it might require just as much, if not more computational power and memory requirements as Dijkstra's algorithm if more operations and calculations have to be done per node. Besides, the A* algorithm's optimality is heavily dependent on the heuristic function it receives. A mediocre heuristic function could lead to sub-optimal results in terms of speed and accuracy and at the same time the better the heuristics, the better the results.

2.2 Perspective Correction

As the procurement of floor plans for our system will be through images taken by users using their mobile phones, an angle skew of the floor plan is to be expected. Therefore, perspective correction must be done on the image prior to floor plan segmentation.

There have been various implementation methods suggested by researchers to rectify the perspective of a target object in an image. Murali S. and Geetha Kiran A. [8] suggested using edge detection for target object, dilate the object to fill empty holes within target object, then erode the target object for distinct object segmentation. Once target object has been extracted, the corner points of the extracted target object are detected and is inputted into a homography matrix. While the results do show successful perspective correction as accordance to the algorithm proposed, it leaves much to be desired as the objects are not scale accurate after perspective correction which lead to stretched out or blurry outputs. This is due to the proposed algorithm not utilizing any prior information about camera lens specification parameters or real-world reference objects.

Ryan Baumann, Christopher Blackwell, and W. Brent Seales [9] proposed a somewhat similar approach to the system in [8] except for using Hough transform for line detection on target objects instead of dilation and erosion for segmentation. In [9], the image is subjected to Canny edge detection to produce an image comprising of only edges. The edge image is then dilated and subject to a Hough Transform to detect all possible lines within the image. A line classification algorithm is then implemented to determine the positions of the lines in the image. The authors then suggest averaging and extending the lines till the edge of the image and utilize the newly acquired lines to compute a destination matrix for perspective correction. Similar to [8], though the results are up to expectations of the algorithm, the issue of scale and aspect ratio of the final results still persist due to the exact reason the researchers at [8] suffers, which is the lack of priori knowledge in regards of camera parameters and object scale.

Lastly, in [10], Shijian Lu, Ben M. Chen and C.C. Ko proposed a method of perspective correction utilizing fuzzy set and morphological operations which, unlike [8] and [9], requires no priori knowledge to produce satisfactory results. The input image is first subjected to global thresholding for binarization of image. Next, four sets of structuring elements are constructed for extraction of tip points of each word in the document image. For stroke boundary extraction, four more sets of structuring elements are constructed. Once the tip points and stroke boundaries have been extracted, an algorithm to classify tip point location and tip point tracing will be executed. This will help form the horizontal lines that correspond to the top and bottom of each rows of text with the implementation of the least square method. A destination quadrilateral matrix is then constructed in regards of the two orthogonal lines obtained from the classified tip points and stroke boundaries whereas the target quadrilateral matrix will be constructed based on the character amount and height-width ratio of characters. As this will create multiple homographies, an optimal one will be decided using classified tip points. Rectification of perspective is then achieved with the optimal homography. This algorithm proposed in [10] proves to be extremely robust as compared to [8] and [9] as the result are well rectified with no traces of aspect ratio distortion whatsoever. However, it is important to note that [10] is only tested on text documents and might not work as well in images containing miniscule amounts of text.

2.3 Lot Segmentation with Lot Labelling

Sheraz Ahmed and Marcus Liwicki [11] [12] [13] have proposed an amalgamation of methods and algorithms in regards of floorplan room/lot segmentation as well as lot labelling techniques. In [11], the authors presented an automated way of detecting and labelling rooms in a floorplan raster image by combining techniques presented in their own works [12] [13]. The proposed system first extracts all text information within the floorplan image by separating the structural information of the floorplan such as walls and borders, leaving behind only the text information. This is done by morphological erosion and dilation to remove thick wall lines, and morphological opening operation to remove thin wall lines. The leftover texts are then subjected to connected components analyzing and smearing techniques to remove additional noise before final extraction of texts. The resultant extraction of the wall lines has allowed the authors to perform structural analysis. As the extracted walls are incomplete, the authors closed the gaps between walls and used the end result to determine the overall boundary of the floorplan. The boundary and extracted walls with the closed gaps are then combined to create the final complete floorplan image and is ready for room segmentation. The authors [11] used SURF matching technique to detect door symbols to close the final small gaps within the floorplan image. This should be unnecessary to our proposed system as mall directories do not typically have gaps separating between lots due to doors symbols. However, the SURF matching technique could prove useful for directory labelling if applied alongside OCR (using Tesseract). Next, the authors detect each individual room's boundary by inverting the floorplan image and subject it to connected component analysis. Once the room boundaries are found, the extracted texts obtained earlier are superimposed upon the segmented room floorplan. All text information is then subjected to OCR and is compared with a predefined dictionary of room titles. The closest OCR match based on Levenshtein distance will be assigned its respective room. The authors implemented an ingenious method of room splitting based on room labels by calculating the horizontal and vertical distance of each label to its neighboring labels. If horizontal distance is greater than its vertical distance, a horizontal boundary will be drawn between the labels. Otherwise, a vertical boundary will be drawn between labels. The results of the proposed system by Sheraz Ahmed, Marcus Liwicki et al [11] [12] [13] are astounding with room labelling and segmentation accuracy coming at a rate exceeding 80% based on 80 floorplan images datasets. This is impressive considering no priori knowledge regarding the test images are used during testing.

3. Aims and Objectives

The aim of this project is to develop a system that would allow users to navigate through indoor establishments and allow users to reach their intended destination quickly and efficiently by simply providing floorplan and directory images. This would prove significant to users who are not familiar of the interior of said establishments to be able to navigate around with ease.

To achieve the aim stated above, several objectives for the project must be fulfilled:

- To develop an algorithm to automate perspective correction process for user captured images of mall plans and directories.
- To develop an algorithm to segment and differentiate between lots and shop name labels in a mall plan and directory list.
- To develop an algorithm to match lots in mall plan according to the mall directory provided by user.
- To implement a routing algorithm for user to navigate from one location to another that returns route in an acceptable amount of time (less than 3 seconds).

4. System Overview

The proposed indoor navigation system will be running on a computer with Windows platform and is developed using Python programming language. The proposed system would automatically convert floorplan and directory images into navigation ready map interfaces and allows users to navigate from one lot to another within an indoor establishment through directions offered in the interactive map interface within the system. The following are the list of programming languages, libraries and APIs that would be used to develop the system:

- Python 2.7.13 (Programming Language)
- Google Cloud Platform Vision API (Optical Character Recognition Library)
- OpenCV3 3.1.0 (Computer Vision Library)

The following are the list of functionalities and features offered by the system:

- Converting floorplan images to navigation-ready map interface
- Automated linking of directory labels to floorplan map interface.
- Provides navigation routes from user selected source location to user selected destination location for indoor establishments

5. Methodology

In this segment, detailed discussion of a proposed methodology to achieve the objectives set in the previous segment is written. In a nutshell, the chronological implementation methods summarizing the core components of the indoor navigation system are as follows: image preprocessing, perspective correction, lot segmentation with directory labelling and routing algorithm implementation.

First, the system receives user captured map and directory images as source images. Then, preprocessing of said images are done to remove noise before perspective correction process can take place, which rectifies the images into a frontal facing orientation. This is done to improve the accuracy of future operations such as text extraction and lot labelling. The extracted lots and labels can then be employed in conjunction with the routing algorithm to calculate the route based on user selected locations. Figure 1 illustrates the steps mentioned and would be further discussed in its respective sub-section with individual flowcharts going into further detail for each core component.

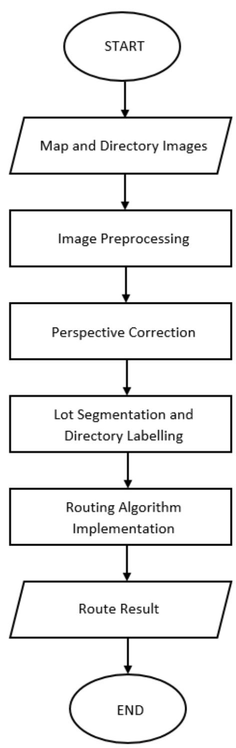


Figure 1 Overview of Methodology for Indoor Navigation System

5.1 Image Preprocessing

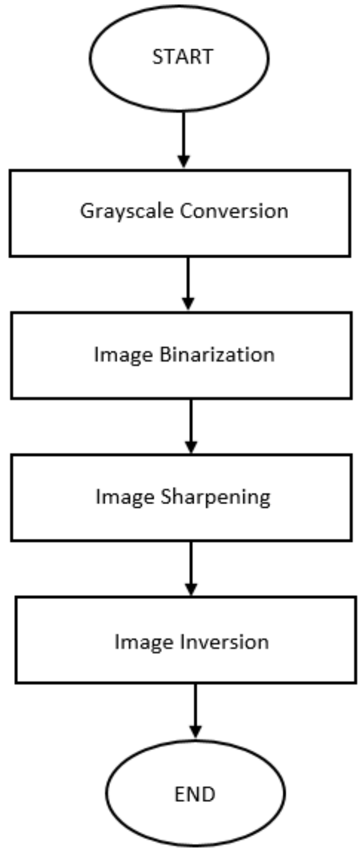


Figure 2 Overview of Image Preprocessing

As the floorplan and directory images are acquired through user's camera, the images are most likely to be riddled with noise and inconsistent illumination. Therefore, preprocessing of said images such as grayscale conversion would have to be done to ease future processing load. Then, thresholding of the grayscale image would allow clearer distinction of target object within the input image (floorplan and directory). Thresholding would not only provide a clearer visual for future processes but also increases efficiency of future operation by binarizing input image as there are now less information to process. An adaptive threshold method, specifically Gaussian adaptive threshold method, is used for binarizing the input images as the user captured images are most likely to contain different illumination intensities in different areas. This is because Gaussian adaptive thresholding calculates the threshold value based on the weighted sum of the neighboring pixel values using a gaussian window (figure 2) to achieve uniform binarization results regardless of light conditions. Besides, the usage of the gaussian window during binarization would simultaneously reduce the amount of noise present in the image as well due to a blurring effect that comes with the implementation of the window. A sharpening process for the thresholded image is implemented as the blurring process prior might have affected legibility of texts in the floorplan image. A sharpening structuring element as shown in figure 3 is used in conjunction with a 2D filter upon the thresholded image to yield sharper edges that defines text information in the floorplan image more distinctly to ensure future operations such as Optical Character Recognition (OCR) will operate more effectively. Lastly, the thresholded image is inverted for clearer distinction between text, walls and routes of the map image.

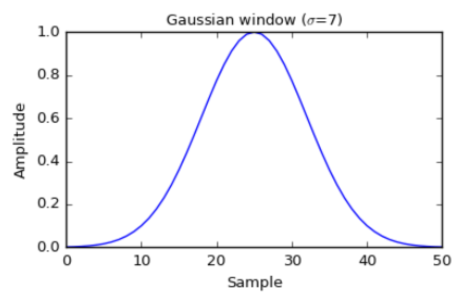


Figure 3 Visualization of a Gaussian window

-1	-1	-1
-1	9	-1
-1	-1	-1

Figure 4 Structuring Element used for Image Sharpening Process

5.2 Perspective Correction

Once preprocessing of floorplans and directory images are complete, the images are subjected to perspective correction as the user taken images are not angled perpendicularly towards the camera lens. Two modes of perspective correction, automated and manual perspective correction, are implemented for map images while directory images are subjected to manual perspective correction. The manual perspective correction mode is introduced for users to rectify images manually if the automatic perspective correction process returns unsatisfactory rectification of source images. Figure 5 illustrates the overall flow of the perspective correction process.

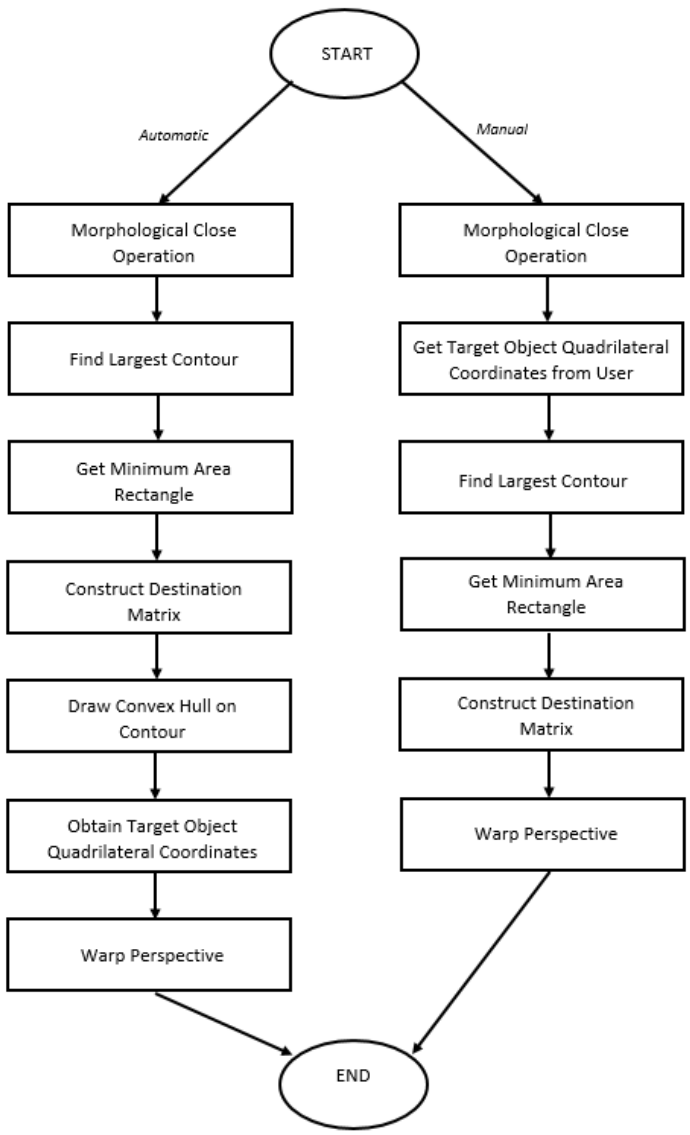


Figure 5 Overview of Perspective Correction Process

Automatic Perspective Correction

The image will first be morphologically closed using a 4x4 sized rectangular structuring element to fill in all empty holes within the floorplan. The image will then be subjected to a contour finding algorithm that can be found in OpenCV's expansive computer vision library of functions. The largest contour will be regarded as the target object and a minimum area rectangle is constructed around the largest contour. The minimum area rectangle will provide with the dimension information of the target object (width, height, x-coordinate and y-coordinates). A destination matrix is constructed based on the obtained dimensions of the minimum area rectangle. Then, a convex hull will be drawn and used to approximate the relative shape of the map image. This is because perspective correction is usually done on a quadrilateral target object, and not on an (n>4)-sided polygon target object, a trait mall directory maps share. This results in a shape approximation with reduced vertices and more accurate representation of a quadrilateral shape being done on an n-sided polygon. The quadrilateral corner points coordinate obtained from the convex hull will be utilized along with the coordinate of the destination matrix to calculate the perspective transform matrix. The result of the perspective transform is then used to warp the perspective of the target object into the intended orientation and view (figure 6).

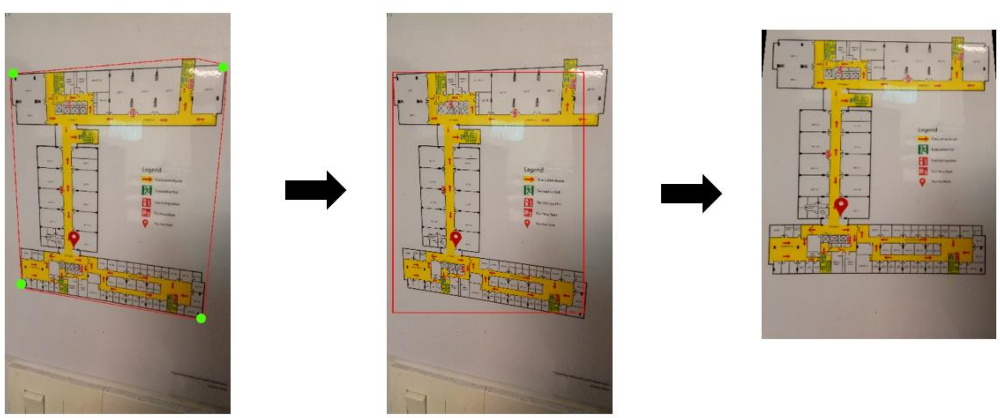


Figure 6 Corner points of Convex Hull (left), Bounding Box of Map (middle), Perspective Corrected Map Image (right)

Since the dimensions for the target object obtained from the minimum enclosing rectangle is used for perspective transform, it will simultaneously segment the target object (mall directory map structure) from the image. The segmented floorplan image will be returned in binary form through the implementation of adaptive thresholding, similar to the preprocessing phase.

Manual Perspective Correction

A similar approach to the automated perspective correction can be employed to manual perspective correction of images with the image first being morphologically closed to fill in all empty holes within the floorplan. Next, manually provided geometric coordinates of the map image by the user are used for the source array of the perspective correction process. Similar to the automated perspective correction process, the image is subjected to a contour finding algorithm and a minimum area rectangle is constructed around the largest contour to obtain the dimension information of the target object. A destination matrix is also constructed based on the obtained dimensions of the minimum area rectangle. The source array, along with the destination matrix, are then used to calculate the transformation matrix, which could be used again to finally obtain the perspective correction transformation results.

The same process can be applied to the directory image as well to correct the perspective and segment out the words in the directory image for text extraction process in the future. Similar to automatic perspective correction, a binarized form of the resultant directory and floorplan image are returned through implementation of adaptive thresholding procedures. This is done to increase efficiency of processing the images in future operations, especially during the routing phase of the system.

5.3 Lot Distinction and Directory Labelling

Using the extracted images, some methods proposed in [11] such as OCR can be implemented for lot segmentation as well as directory linking and labelling. Lot segmentation with lot labelling is required for the system to serve as the foundation for routing algorithm implementation (explained in next segment). The segmented directory image from the perspective correction phase is subjected to OCR (optical character recognition) [11] through Google Cloud Platform's Vision API, an online OCR library by Google that utilizes machine learning for text recognition, to recognize the texts within the directory image and store each line of directory text (which includes lot number and outlet name) into a dictionary of lot labels.

Directory Text Processing

In directory images, information for each lot are represented in a line containing the lot number and lot name. Therefore, extraction of text based on directory lines is required. As Google's Vision API does not recognize texts that are far away from each other as being in the same line or paragraph, an algorithm must be implemented to group texts that are of distance from one another into a single line. By grouping relevant lots into individual lines, distinction between lot names and lot numbers can be achieved. Figure 7 illustrates the difference between the built-in text grouping mechanism of Google's Vision API and the intended grouping results for the shop lot name and lot number of our directory image. Figure 8 illustrates the steps required to achieve the distinction between outlet name and lot number within a directory text image.

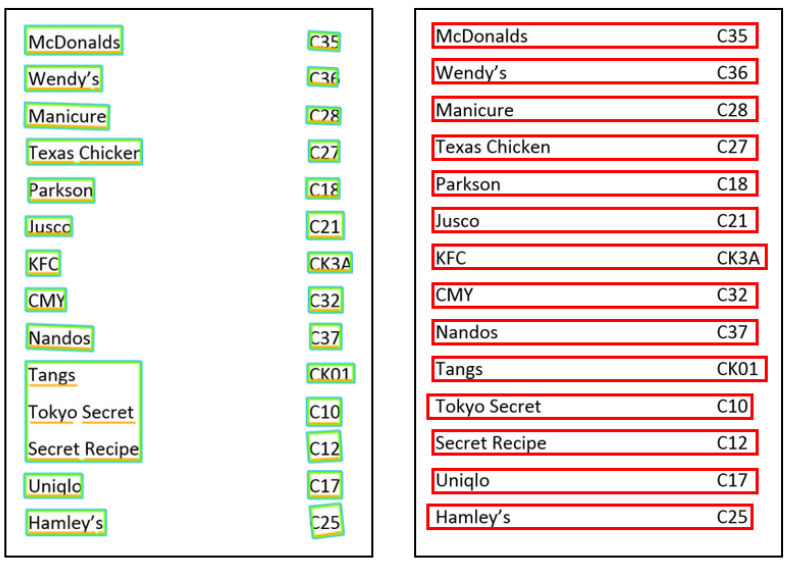


Figure 7 Grouping of Texts from Vision API (left) and intended grouping of texts (right)

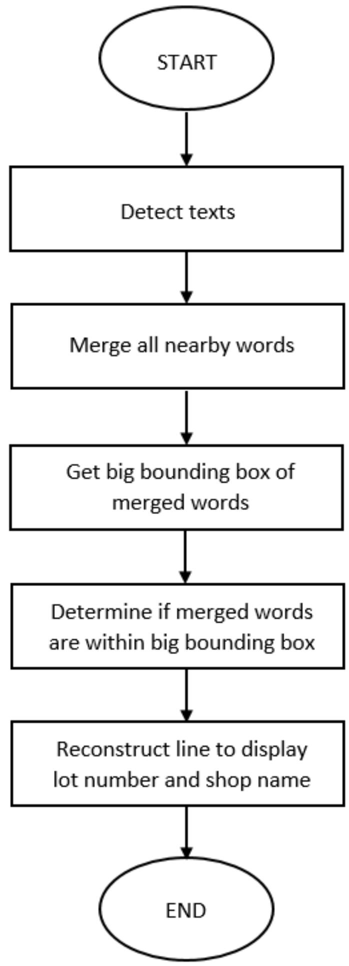


Figure 8 Basic Steps for Lot Number and Lot Name Distinction in a Directory

Text coordinate information of merged words is required in order to proceed with future text processing operations. Firstly, the perspective corrected directory image is subjected to OCR to obtain the raw text information which includes text descriptions and text coordinates (within the image) for each individual word. A list containing only text descriptions within the directory image is stored as well. While the list containing only text descriptions already provides merged word texts courtesy of Vision API's auto merged word identification system, it however, does not contain the coordinates for the merged words. Using the raw text information and text description list, a comparison can be made between the word indexes of the two sets of information to chain multiple nearby words into a single line that contains text coordinate information as well. The resultant merged word information is then appended into an array.

$$gradient = \frac{y2 - y1}{x2 - x1} \quad (1)$$

$$yMin = (y1) - (gradient * (x1 - xMin)) \quad (2)$$

$$yMax = (y1) + (gradient * (xMax - x1)) \quad (3)$$

A large bounding box is constructed around each individual merged word based on the coordinate information of each merged word. The bounding box determines an average height value by extracting and averaging the y-axis component with the x-axis component of each individual merged word to calculate a gradient value (denoted in 1). The gradient value is used to determine a maximum and minimum y axis value (denoted in 2 and 3) of the big bounding box to allow optimum text grouping of distanced words even in skewed conditions (figure 9).

Figure 9 Example of a constructed bounding box in skewed conditions

Each individual merged text coordinate point is compared against a point checking algorithm to determine if a coordinate point is within bounds of a polygon. In this case, all four coordinate points of the merged text will be iterated through the point checking algorithm against each instances of the constructed big bounding box. If all four coordinate points are within bounds of the big bounding box, the merged word will be regarded as a match and the position of the matched merged word within the merged word array will be recorded for future reference. All merged words that are regarded as a match are further merged together based on the recorded index of each matched word recorded prior. Since all merged words that are within the same big bounding box instance will be labelled with the same index, a merged line containing the lot name and lot number can be obtained. The process continues until no more matched merged words are left.

All merged lines containing lot number and lot name info will be stored in an array. An arbitrary symbol will be inserted in between the lot names and lot number of each merged line for easier identification and separation. The lot number and lot names will be inserted into a dictionary with the lot names being the key and the lot number being the value. This is done to allow users to search for matching lots in the future through shop lot names. The result of the shop name and lot number distinction from directory image can be seen in figure 10.

McDonalds	C35
Wendy's	C36
Manicure	C28
Texas Chicken	C27
Parkson	C18
Jusco	C21
KFC	CK3A
CMY	C32
Nandos	C37
Tangs	CK01
Tokyo Secret	C10
Secret Recipe	C12
Uniqlo	C17
Hamley's	C25

```
Parkson - C18
CMY - C32
Nandos - C37
Tokyo Secret - C10
Secret Recipe - C12
Texas Chicken - C27
KFC - CK3A
Manicure - C28
McDonalds - C35
Hamley's - C25
Jusco - C21
```

Figure 10 Directory Image (left) and Result of Directory Text Processing (right)

Map Text Labels Processing

The segmented floorplan image obtained during the perspective correction phase will be subjected to OCR as well and have its text information recognized. The texts from the map image, which consists of lot number text information and coordinate information, will be stored in an array and will be compared against the dictionary of lot labels defined prior. As proposed in [11], the closest OCR match will be assigned its respective lot in the floorplan image. Levenshtein distance will not be used in this instance as comparison through a dictionary keyword requires an exact match, which contrasts the usage of the Levenshtein distance method.

As the lot label dictionary defined prior utilizes shop names as key values, users will be able to search for shop names which in turn returns the corresponding dictionary value, the lot number. The returned lot number value will then be matched with the map text array. With this, the outlets listed in the directory will be successfully linked to its respective individual lot in the map image and will be the base of operations for users to search and navigate between lots in our indoor navigation system. Once the floorplan images have its lots labelled accordingly to its directory, users will now be able to choose their desired shops from the directory and have their choices reflect accordingly in the floorplan image. The next step is to provide navigation functionality to the labelled floorplan image.

5.4 Routing Algorithm Implementation

The system will be utilizing the A* algorithm as the routing algorithm for the system's navigation function. A* algorithm is a shortest path finding algorithm that is an extension of Dijkstra's algorithm with the extension being a heuristic function being incorporated into its distance calculating operations on nodes. The pseudocode for the implementation of the A* algorithm [3] is illustrated in figure 11.

```
Input: Source vertex s
OPEN.insert(s)
while OPEN ≠ ∅ do
    u = OPEN.extract_min()
    foreach vertex v ∈ Adj(u) do
        g(v) = g(u) + w(u, v)
        v′ = check_for_duplicates(v)
        OPEN.insert(v′)
        CLOSED.insert(u)
```

Figure 11 Pseudocode for implementation of A algorithm*

As no prior knowledge in regards of the map input images infrastructure and routes are known, the conventional approach of implementing nodes in the form of a tree graph based on the map's infrastructure that could then be superimposed upon the map image cannot be used. Therefore, a different approach of node implementation was taken for our indoor navigation system, whereby each individual pixel in the map input image will be regarded as a node. Therefore, each node pixel can be represented or identified by its corresponding coordinates.

As presented in the previous segment, users are now able to search for their desired lots from the directory and have their choices reflect accordingly in the floorplan image by simple comparison between dictionary key value and map label text value. Once a match has been found, the corresponding map label text value, which contains the coordinate points information, can be used to pinpoint the exact pixel or node location within the map input image. This technique is applicable for finding the starting node (starting destination) and goal node (ending destination) to be used in our path finding algorithm. An 8-connectivity neighbor traversing method is implemented for the routing algorithm of the indoor navigation system as it is sufficient in providing satisfactory routes from starting node to ending node.

As mentioned before, the lack of prior knowledge in regards of the map image’s layout requires automatic detection of walls and non-passable sections of the map to be implemented. Utilizing the binarized properties of the process input map image, pixels that are below a certain threshold value are deemed as a wall or a non-traversable pixel. This is because the adaptive thresholding binarization method during the image preprocessing phase returns map images that can be characterized as having black outlines for walls and white portions as traversable routes (figure 12). This will allow our routing algorithm to avoid traversing through perceived walls in an input map image when finding the shortest path to goal destination by implementing the above technique during neighbor pixel traversal. Before traversal, the wall detection technique will be utilized in conjunction with neighbor pixel return phase to only return non-wall neighbor pixels.

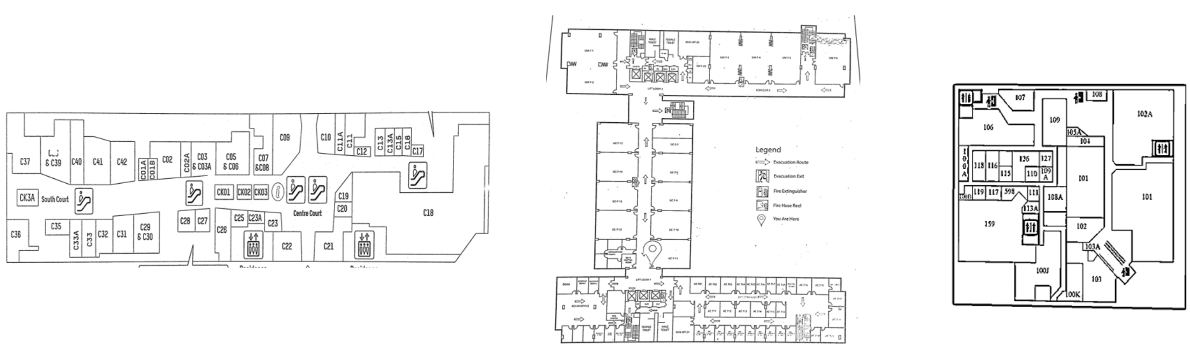


Figure 12 Processed Map Input Images Share Common Characteristics

The technique comes with a disadvantage however, as the starting node and ending node are placed within the corresponding lot label within the map image itself, the traversable routes will be confined within the walls of the starting and ending location lots. This will result in the algorithm being stuck and returning no available routes. Therefore, relocation of starting and ending nodes to a location that is fully traversable must be done for proper navigation to take place. The approach used to achieve this is by traversing the image pixel by pixel from the starting and ending node's coordinates in four directions (north, south, east, west). The values of each traversed pixel are referenced, and traversal stops when a black pixel is encountered (border of the bounding lot walls). The point of traversal halt will have its coordinates recorded and have a new starting or ending node to be assigned right outside of the lot border (figure 13). This presents the routing algorithm with a total of 4 starting nodes and 4 ending nodes overall.

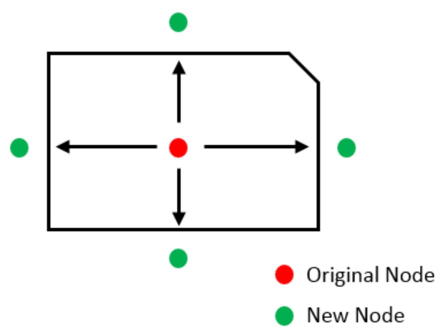


Figure 13 Reassignment of Starting and Ending Nodes

Before employing the A Star algorithm, a heuristic must be decided. Two types of heuristics have been employed for the path finding algorithm of the system, which is the Manhattan Distance and the Squared Euclidean Distance heuristic. It is important to note that heuristics used in an A Star algorithm must never overestimate the distance cost, else the path returned will be suboptimum, voiding the advantages of the A Star algorithm among breadth first search algorithms. The Manhattan Distance returns an estimated cost by addition of the absolute difference of the subtraction of two corresponding points (denoted in 4). The Euclidean Distance returns an estimated cost by summing the squared subtractions between two corresponding points (denoted in 5).

$$manhattan\ distance = |x1 - x2| + |y1 - y2| \quad (4)$$

$$squared\ euclidean = (x1 - x2)^2 + (y1 - y2)^2 \quad (5)$$

The Manhattan distance is restricted in only horizontal and vertical distance estimation whereas the Squared Euclidean metric returns the shortest distance between two points in a plane (figure 14). This makes the Squared Euclidean metric to theoretically return a shorter path in terms of cost as compared to the Manhattan Distance metric. Since the system utilizes an 8-connectivity neighborhood traversal method, the nature of the Manhattan metric being a vertical and horizontal movement only metric might present a jagged route pattern as opposed to the Squared Euclidean method, which provides diagonal movement for representing routes. Thus, the Squared Euclidean method is more suitable provided the neighborhood are of 8-connectivity, otherwise, a 4-connectivity neighborhood approach will more likely befit the Manhattan metric.

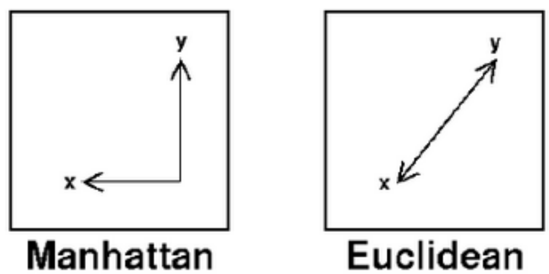


Figure 14 Comparison Between Manhattan and Euclidean Distance

Referencing the pseudocode in Figure 11, the implementation of the A Star algorithm for the indoor navigation system bear much similarity except for some minor changes to accommodate the relocated starting and ending node coordinates. Each starting coordinate will be iterated in conjunction with each of the four ending coordinates when executing the path finding process. This means that the system will potentially go through the A Star path finding algorithm at a maximum of 16 times. This approach is computationally taxing as the algorithm will traverse through a significant number of nodes for potentially multiple iterations. Therefore, traversal of fewer nodes is necessary to increase performance as the system currently regards each pixel as a node. Before initiation of path finding, the map input image, starting and ending coordinates are reduced by a factor of four. Reduction of image will reduce the number of pixels within the image, thus reducing the number of nodes required to traverse as well. The starting and ending nodes are also reduced to reflect the position of the shrunk map input image. This resulted in a vastly reduced computational load during the path finding process and improved time consumption by a significant margin. The shortest path returned by the algorithm will be plotted on the input map image. Figure 15 below shows the result of the system's routing algorithm from start to goal.

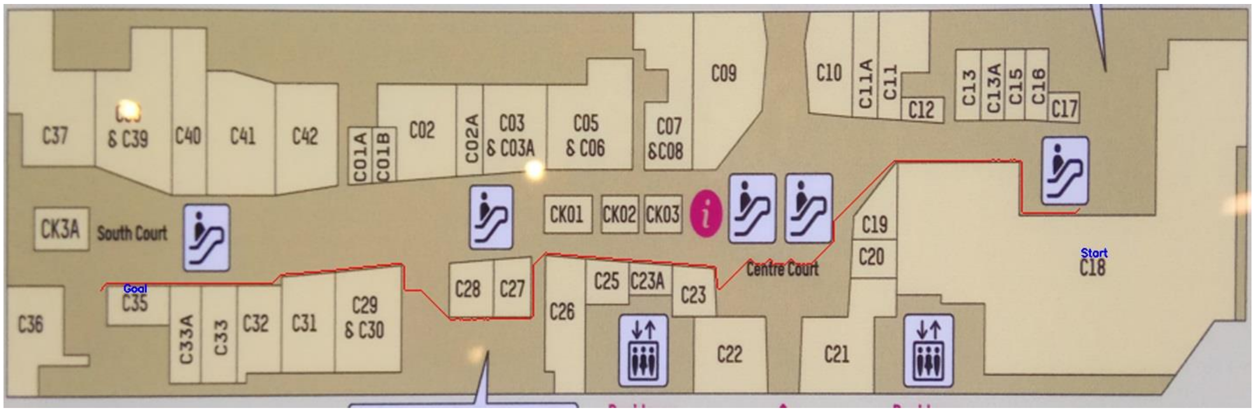


Figure 15 Routing Result in red

Testing

Our system went through four testing phases to assess its legibility in four components based on the objectives stated prior to achieve a universal indoor navigation system. A total of 7 test map images and 7 directory images will be used for system testing. The four phases includes automatic map perspective correction results, directory text segmentation accuracy, lot number and mall plan linkage accuracy and efficiency of routing algorithm. The system is tested on a computer with a dual core Intel i5-4210H processor.

Phase 1 – Automatic Map Perspective Correction Results:

As a rectified map image will be the basis for all future routing operations for our indoor navigation system, perspective correction shall be tested. Manual perspective will not be tested as the rectified image are assumed to be satisfactory as per the user's judgement. A distorted or source map input image will be displayed alongside the rectified image through automatic perspective correction process in the next section (table 1).

Phase 2 - Directory Text Segmentation Accuracy:

This phase will assess the accuracy of text detection done on directory images in regards of shop name and lot number distinction and extraction. Two criteria will be assessed in this testing phase, first being the number of lines detected overall against the original number of directory text lines within the image (denoted in 6). This is to assess the reliability of the text segmentation algorithm of the indoor navigation system. The second criteria is the number of correct extracted lot number in each detected line of a directory image (denoted in 7). Accurate extraction of lot name is unnecessary as node placement is based on lot number information. This is to assess the accuracy of the text segmentation algorithm of the indoor navigation system. The results will be tabulated in table 2 and 3 in the next segment.

$$Lines\ detected = \frac{no.\ of\ detected\ lines}{total\ lines\ in\ image}\ x\ 100\% \quad (6)$$

$$Lot\ distinction\ accuracy = \frac{no.\ of\ correct\ lines\ segmented}{total\ detected\ lines}\ x\ 100\% \quad (7)$$

Phase 3 – Lot Number and Mall Plan Linkage Accuracy

Evaluation of the number of linkage between the detected lot labels in the directory image with the lot labels within the map image will be performed in this phase. This is important as the linkage of lots between directory and map images are the basis for routing and navigation to take place in the system. As a simple matching method will be used to link the map label and the lot name in the lot number dictionary defined during directory text segmentation, a map label detection accuracy test will suffice as linkage accuracy. A function to calculate the linkage accuracy percentage is denoted in 8. 3 test images will be used as the map labels are distinct and present.

$$Linkage\ accuracy = \frac{no.\ of\ correct\ map\ label\ detection}{total\ number\ of\ map\ labels}\ x\ 100\% \quad (8)$$

Phase 4 – Routing Algorithm Efficiency

The efficiency of the routing implementation approach used in the system can be caused by many factors which includes routing algorithm execution iterations, heuristics used, distance between starting and ending nodes and the dimensions of the map image itself (due to higher amount of pixels, thus more nodes to traverse through). To reduce the erratic factors that might affect consistency of results for the system's routing algorithm efficiency, the testing for this phase will be limited to one map image. This is to assure the results from testing will not be affected by image dimension differences, and the best and worst case scenarios for the routing algorithm can be tested more consistently. The best case scenario in this case would be a single iteration of the routing algorithm's execution before a path is found, whereas worst case scenario would be an iteration of 16 times for the routing algorithm to find a path. In conjunction with the scenario metrics, a long distance between starting and ending nodes as well as a short distance between start and end nodes will be tested. Lastly, with the implementation of the A* algorithm, the heuristics plays the largest role in routing efficiency. Two heuristics namely Manhattan distance and Squared Euclidean Distance metrics will be used in testing to encapsulate the above test factors.

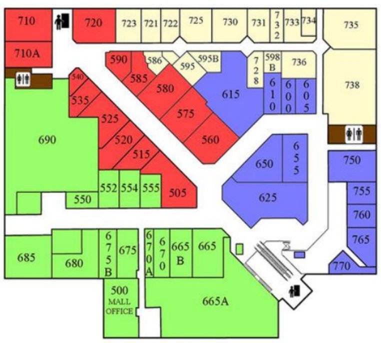
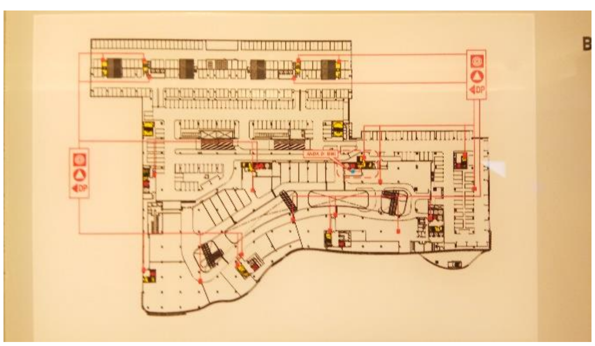
In conclusion, this testing phase will have two test instances (short and long distances) for each scenario cases (best case and worst case) tested under two different heuristics. Execution time will be the deciding measure used to determine the efficiency of the routing algorithm. The results will be tabulated in table 5 and 6 in the next segment.

6. Results and Discussion

This section details the test results from the explained testing procedures explained in the previous segment for the indoor navigation system. The results for 4 test phases are presented in tabulated forms with an in depth analysis and discussion of the obtained outcomes in its respective sub-section.

6.1 Phase 1 Results

Table 1 tabulates the automatic perspective correction results on source map images. The source image is depicted at the left column of the table and the rectified results are presented in the right column of the table. A total of 7 map images are used for testing in this phase. The map images consists of user taken map images from mobile phones and raster map images taken from the internet.

<table>
<tr><th>Source Images</th><th>Automatically Rectified Images</th></tr>
<tr><td>Image 1
</td><td></td></tr>
<tr><td>Image 2

</td><td>
</td></tr>
<tr><td>Image 3
</td><td></td></tr>
<tr><td>Image 4</td><td></td></tr>
</table>

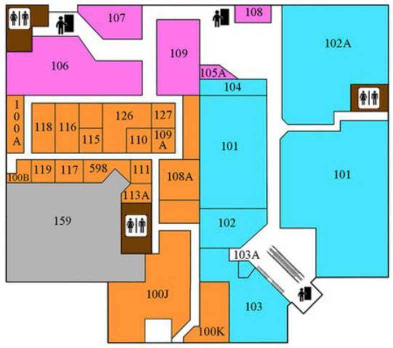
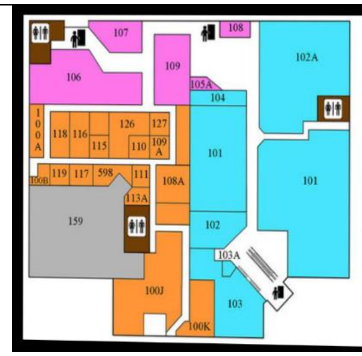
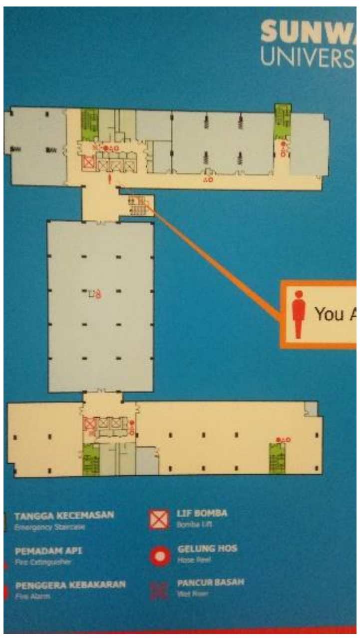
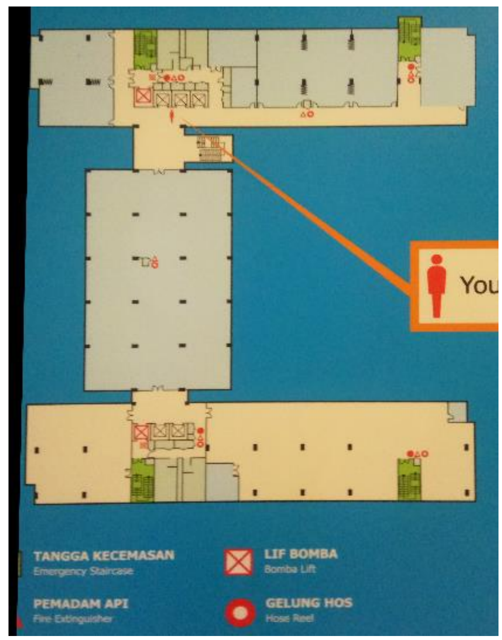
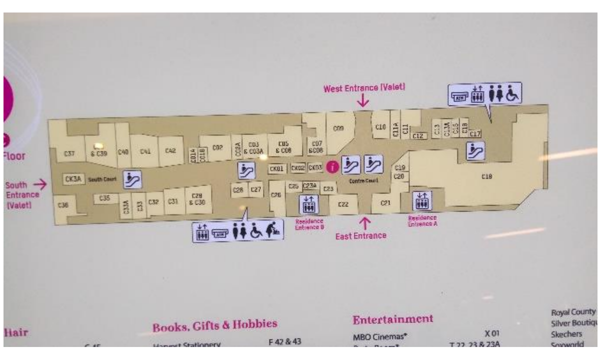
<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>Image 5

</td>
<td>
</td>
</tr>
<tr>
<td>Image 6

</td>
<td>
</td>
</tr>
<tr>
<td>Image 7</td>
<td></td>
</tr>
</table>

Table 1 Automatic Perspective Correction Results on Map Images

The results in phase 1 illustrates the rectified map images through automatic perspective correction process. It can be seen in test images 1, 2, 4, 5 and 6 that map images that are less arbitrary in shapes produces a more appropriate rectification results whereas test images 3 and 7, which presents a more arbitrary structure, yielded an even more distorted rectification result. This is because a less arbitrarily shaped map structure allows closer shape approximation predictions based on the convex hull obtain on the contours of the map structure, which results in better rectification results. Besides, the arbitrary shapes typically present the automatic perspective correction process with a higher number of vertices after convex hull identification due to the higher number of gaps needed to be filled by the convex hull. This causes the shape approximation method to choose vertices that are not representative of the intended or closest shape of the map structure to achieve good perspective correction. As for test image 7, only a small portion of the overall map image was detected, and perspective corrected (inaccurately). This is because the perspective correction method uses the largest contour found within an image to identify target map structure for rectification. Unlike other test images, which displays target map structure as a whole, test image 7 featured a map image with disjointed components to represent overall structure of the map. This causes many contours to be detected within the image. The resultant perspective corrected image for test image 7 represents the largest contour of test image 7 to be rectified (figure 17). Therefore, the automatic perspective correction algorithm for our indoor navigation system can only be done on map images that are less arbitrary in shape without disjointed map components. However, if user insists on utilizing incompatible map images to be perspective corrected, users can opt for manual perspective correction method offered by the system to pick and choose bounding corner coordinates for optimum perspective correction regardless of shape or structure.

Figure 16 Largest Contour Falsely detected as Map Structure

6.2 Phase 2 Results

Table 2 tabulates the directory line detection results done on a series of directory test images (refer to appendix). The results depicted in percentage for each test image is located at the furthermost right column and an overall average percentage of all test results are tabulated at the bottom right corner of the table. For table 3, the successfully segmented lines in table 2 is measured for lot number extraction accuracy. The results are tabulated in a similar way as table 2, showing individual image results and total average results in percentage form. A total of 6 directory images are used for testing in this phase. The directory images consists of user taken images from mobile phones and raster directory images taken from the internet.

Test Image	Total Directory Lines	Successful Segmented Lines	Results
1	222	95	42.79%
2	287	176	61.32%
3	287	212	73.86%
4	123	96	78.04%
5	48	33	68.75%
6	14	11	78.57%
	Total		Average
	981	623	63.50%

Table 2 Directory Line Detection Accuracy

Test Image	Total Detected Lines	Correct Segmented Lines	Results
1	95	39	41.05%
2	176	24	13.63%
3	212	193	91.03%
4	96	84	87.50%
5	33	31	93.93%
6	11	11	100.00%
	Total		Average
	623	623	63.50%

Table 3 Lot Number Extraction Accuracy

For directory text segmentation, table 2 of phase 2 shows the results for successfully segmented directory lines within a directory image. There appears to be a slightly lower detection accuracy rate for the first two test directory images with a 42.79% and 61.32% directory lines detection accuracy. The low accuracy readings can be identified by a number of factors. One of them being the perspective corrected directory image itself. As all directory test images goes through a mandatory manual perspective correction done by the user, a subpar perspective correction on the directory image will yield vastly different directory text segmentation results. Another factor that might have affected the results are the resolution of the test images. The lower resolution of the first two test directory images further cements the hypotheses as the other test images all have vastly superior line text segmentation accuracy rates averaging above 70%. Lastly, the final factor that might affect text detection accuracy in directory images are the line spacing of each line within the directory image. As the directory text segmentation algorithm to identify and achieve lot number and lot name distinction uses bounding boxes to merge words that are separated by a far horizontal distance, a narrow line spacing may cause the bounding boxes to overlap with one another. The overlapping bounding boxes will interfere with the point checking algorithm (elaborated in methodology) and cause false negatives, thus disregarding a detected line even though all relevant merged words are within the bounding box. The implementation of the bounding box to be skew tolerant have further contributed to the overlapping phenomenon as a narrow line spacing enhanced by potential skewness introduced in imperfect manual perspective correction may have caused the bounding boxes to have merged with lines below its intended y-axis. Figure 18 illustrates the overlapping bounding boxes in a compact directory listing.

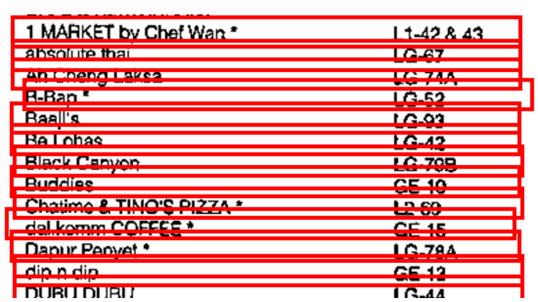


Figure 17 Overlapping Bounding Boxes of Directory Lines

As for table 3 of phase 2, the higher resolution directory test images (3,4,5 and 6) have returned an impressive lot number accuracy reading exceeding 90% on average. However, the low-resolution images (1 and 2) returned an abysmal reading of only 41.05% and 13.63% respectively. This is due in part of the illegibility of texts returned by test directory images 1 and 2 in a state of low resolution (figure 19). The higher accuracy reading returned by test directory image 2 was because the lot number of that particular test directory image contains only numerical figures, which are vastly more legible in comparison to test directory image 1, which contains alphanumerical lot numbers that are less clear for text extraction (figure 20). These results show that clarity of text in an image is crucial for text recognition accuracy.

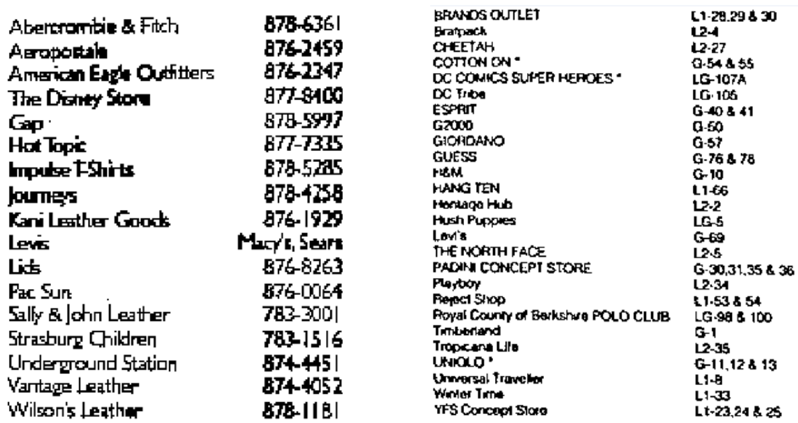


Figure 18 Low Resolution Directory Images

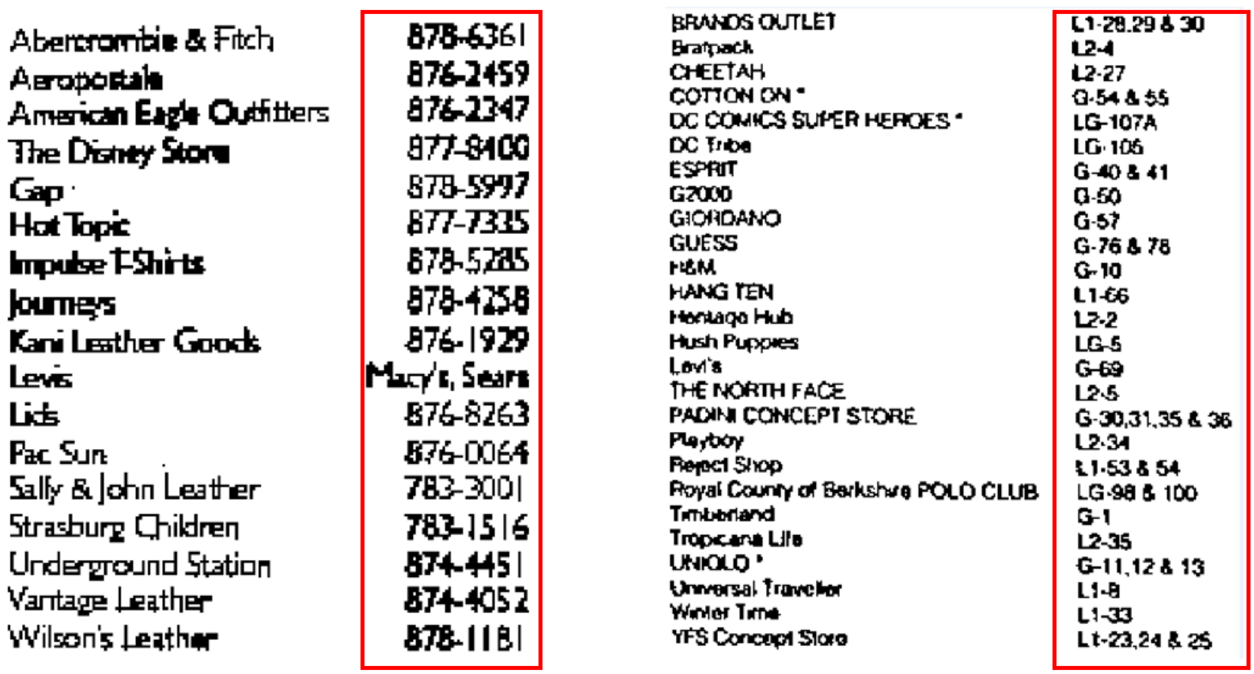


Figure 19 Legibility of Numeric (left) is better than Alphanumeric (right)

6.3 Phase 3 Results

Phase 3 details the results of map label text detection accuracy. The results will reflect the linkage accuracy of the indoor navigation system as well because a correct detection of a map label is a confirm link to the directory text information within the directory dictionary defined after directory text extraction provided the key values for the dictionary is accurate too. A total of 3 map images are used for testing in this phase (refer to appendix). The images consists of user taken map images from mobile phones and raster map images taken from the internet.

Test Images	Total Map Labels	Correct Map Labels	Results
1	57	45	78.94%
2	30	29	96.67%
3	58	41	70.68%
	Total		Average
	145	115	79.31%

Table 4 Map Label Detection Accuracy

It can be seen that the test yielded an average accuracy reading of 79.31%, meaning that 79.31% of the time on average, a lot name searched by the user will yield a match and reflect within the map image. Similar to phase 2, the accuracy of map label text detection can be influenced by the same factors that have plagued phase 2. Imperfections in the perspective correction of map image be it automatic or manual could present the system with different accuracy readings. The resolution of the image will affect the legibility of the map labels within the image as well, causing a dip in accuracy for map label detection. It is interesting to note that there is a fairly significant dip in accuracy for test images 1 and 3. Upon closer inspection of the test images, it was found that the dip in accuracy was due to some of the map labels being positioned in a vertical orientation (figure 21). As Vision API can only accommodate text detection with slight to moderate skews, a vertical orientation of texts will yield an inconclusive detection. Therefore, the dip in accuracy for test images 1 and 3 are to be expected.

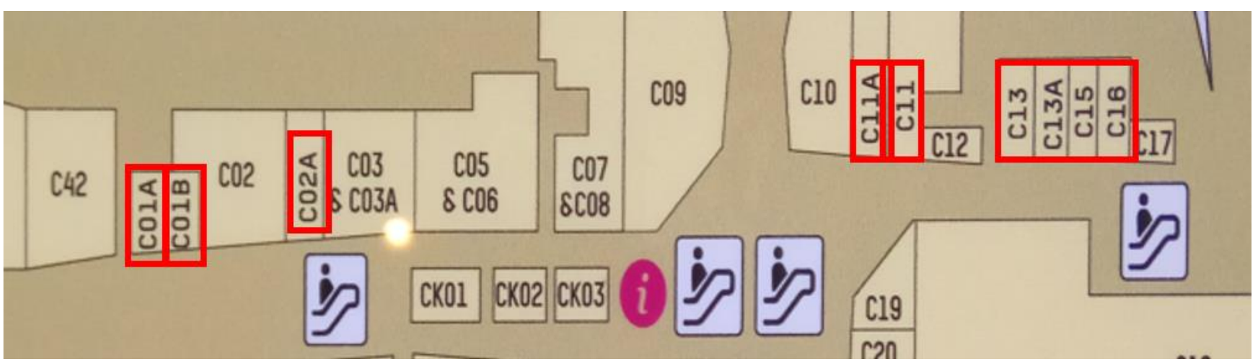


Figure 20 Vertically Placed Map Labels

6.4 Phase 4 Results

Phase 4 results details the results of routing algorithm efficiency using execution time as the measurement metric. Table 5 tabulates execution time results of the A* routing algorithm using the Manhattan Distance Heuristics separated into best case scenarios and worst case scenarios. The same applies to results in table 6 with the exception of the A* routing algorithm running using the Squared Euclidean Distance heuristics. As mentioned in the previous section (refer to Testing sub-section in Methodology), the best case scenario is a 1 time iteration of the algorithm in finding the route whereas the worst case scenario is a 16 time iteration of the algorithm before finding a path.

Manhattan Distance Heuristic

Distance	Execution Time (seconds)	
	Best Case Scenario	Worst Case Scenario
Short	0.25	0.56
Long	2.49	16.8

Table 5 Execution Time of Routing Algorithm Using Manhattan Distance Metric

Squared Euclidean Distance Heuristic

Distance	Execution Time (seconds)	
	Best Case Scenario	Worst Case Scenario
Short	0.03	0.56
Long	0.11	22.62

Table 6 Execution Time of Routing Algorithm Using Squared Euclidean Distance Metric

For routing algorithm performance, tables 5 and 6 shows the execution time for the system to locate a route from starting node to ending node for two types of heuristics. The Manhattan Distance heuristics is run with the use of a 4-neighborhood traversal system whereas the Squared Euclidean is run with the 8-neighborhood traversal system. The difference in path traversal pattern between an 8 and 4-neighborhood routing algorithm is illustrated in figure 22. It can be seen in table 5 that for short distances, the system objective of obtaining a route from starting point to end in less than 3 seconds have been achieved, with an execution time of 0.25 seconds for best case scenario and 0.56 seconds for worst case scenario. The long-distance category on the other hand, managed to achieve an execution time of 2.49 seconds for best case scenario and a lengthy 16.80 seconds for worst case scenarios. As for table 6, the short distance metric yielded an execution time of 0.03 seconds for best case scenario and 0.56 seconds for worst case scenario. The long-distance metric yielded 0.11 seconds in execution time in comparison with the whopping 22.62 execution time of its worst-case scenario counterpart.

A trend can be observed by comparison of the execution time between tables 5 and 6. The Manhattan Distance heuristic returned a generally slower execution time for best case scenarios when compared to the best-case scenario execution time of the Squared Euclidean distance heuristic. The difference is upwards of a factor of almost 10 for short distances and around 20 for long distances. However, a reversal trend is observed for worst case scenarios as there appears to be no difference in execution time (0.56 seconds) for short distances between the two heuristics. In fact, the Manhattan Distance heuristic outperformed the Squared Euclidean Distance heuristic on long distanced worst-case scenarios by a significant amount of time. This may be because the Euclidean Distance metric returns the absolute shortest path between two points, whereas the Manhattan Distance returns the shortest route based on a limited 2D movement on the x and y axis. This contributed to the Squared Euclidean approach benefitting from short distances. However, as a path gets more convoluted and complex, the Manhattan Distance's precedes the Squared Euclidean Distance metric. This is because the squaring mathematical operation present in the Squared Euclidean approach is more computationally taxing compared to the linear computations of the Manhattan Distance approach, thus increasing execution time when covering longer and more complicated distances.

Figure 21 Difference in Traversal Patterns between Manhattan (top) and Squared Euclidean (bottom)

7. Conclusion

In this paper, an indoor navigation system has been proposed and implemented that does not require external means such as priori knowledge, beaming devices and GPS for indoor navigation purposes. Through techniques such as automatic or manual perspective correction for map and directory images, directory line text segmentation, lot and directory linking and A Star algorithm routing implementation, the system is able to not only show routes to and from user's desired location, but also be able to do it below 3 seconds under favorable circumstances. However, accuracy for directory line text segmentation leaves much to be desired. Future enhancement must be done to improve the results of directory line text segmentation in order to enhance the reliability of the indoor navigation system in terms of lot searching, since it is directly influenced by the system's ability to distinct between lot number and name.

References

[1] *Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs" (PDF). Numerische Mathematik.* ***1****: 269–271. doi:10.1007/BF01386390.*

[2] *Mehlhorn, Kurt; Sanders, Peter (2008). "Chapter 10. Shortest Paths" (PDF). Algorithms and Data Structures: The Basic Toolbox. Springer. doi:10.1007/978-3-540-77978-0. ISBN 978-3-540-77977-3.*

[3] *Felner, Ariel (2011). "Position Paper: Dijkstra's Algorithm versus Uniform Cost Search or a Case Against Dijkstra's Algorithm." Proc. 4th Int'l Symp. on Combinatorial Search.*

[4] *Dijkstra, Edsger W., Reflections on "A note on two problems in connexion with graphs (PDF)*

[5] *Chuang Ruan; Jianping, Luo; Yu, Wu (2011). "MAP NAVIGATION SYSTEM BASED ON OPTIMAL DIJKSTRA ALGORITHM" (PDF).*

[6] Chandak, Aniket; Bodhale, Rutika; Burad, Raveena (2016). "Optimal shortest path using HAS, A star and Dijkstra algorithm" (PDF). Imperial Journal of Interdisciplinary Research (IJIR), Vol-2, Issue-4, 2016. ISSN: 2454-1362.

[7] Haifeng, Wang; Jiawei, Zhou; Guifeng, Zheng; Liang, Yun (2014). " HAS:Hierarchical A-Star algorithm for big map navigation in special areas " (PDF). 2014 International Conference on Digital Home

[8] Kiran A, Geetha; S, Murali (2013). "AUTOMATIC RECTIFICATION OF PERSPECTIVE DISTORTION FROM A SINGLE IMAGE USING PLANE HOMOGRAPHY" (PDF). International Journal on Computational Sciences & Applications (IJCSA), Vol.3, No.5.

[9] Baumann, Ryan; Blackwell, Christopher; Scales, W. Brent. "Automatic Perspective Correction of Manuscript Images" (PDF).

[10] Shijian Lu, Ben M. Chen, C.C. Ko (2005). "Perspective rectification of document images using fuzzy set and morphological operations" (PDF). ScienceDirect Image and Vision Computing 23 (2005) 541–553*

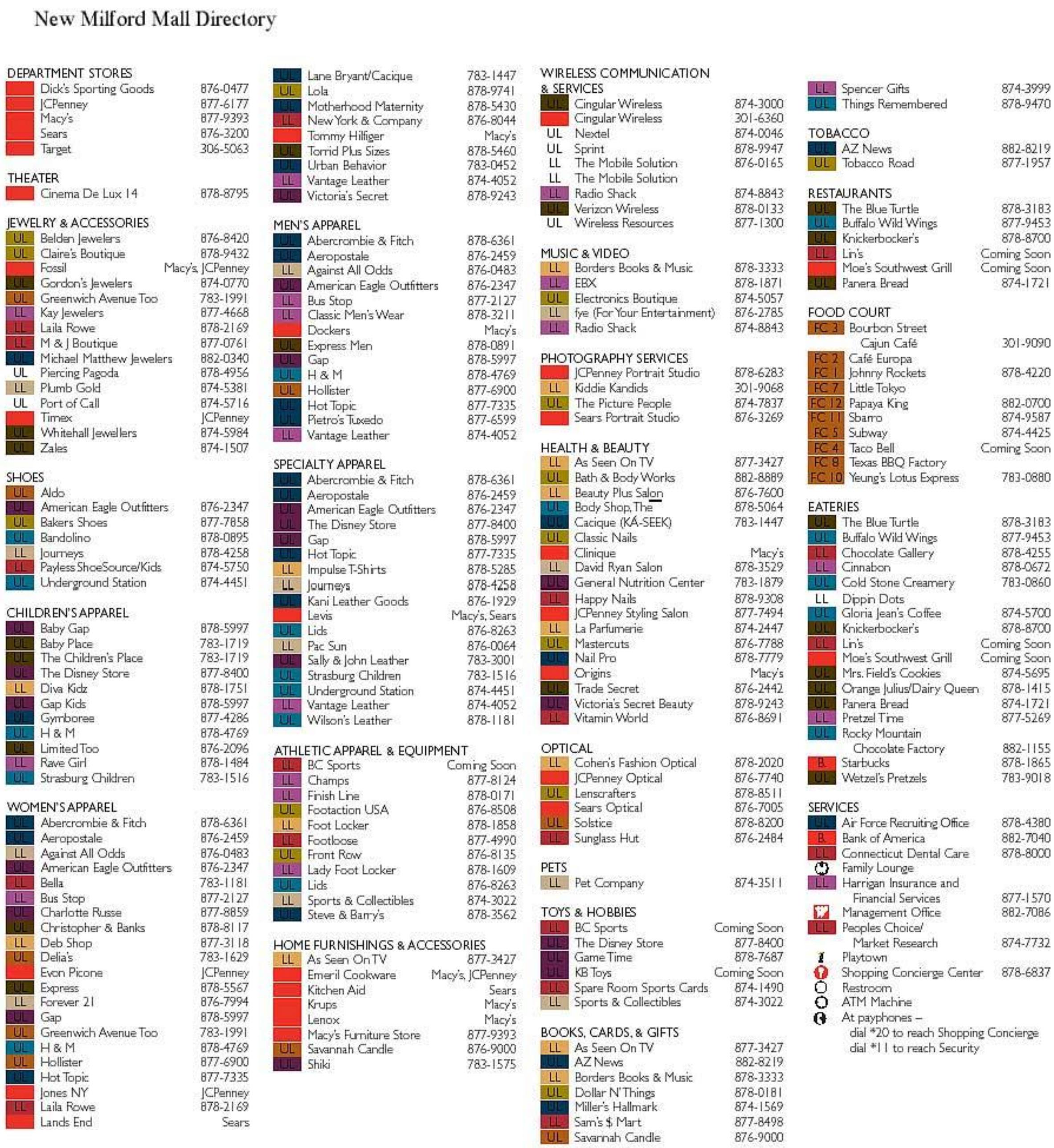
*[11] Sheraz Ahmed, Marcus Liwicki, Markus Weber, Andreas Dengel (2012). "*Automatic Room Detection and Room Labeling from Architectural Floor Plans*" (PDF). 2012 10th IAPR International Workshop on Document Analysis Systems*

[12] Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, Gemma Sánchez (2013). "Statistical segmentation and structural recognition for floor plan interpretation" (PDF). DOI 10.1007/s10032-013-0215-2

[13] Sheraz Ahmed, Marcus Liwicki, Andreas Dengel (2012). "Extraction of Text Touching Graphics using SURF" (PDF). 2012 10th IAPR International Workshop on Document Analysis Systems

Appendix

Directory Test Images (Phase 2)



New Milford Mall Directory

DEPARTMENT STORES

	Dick's Sporting Goods	876-0477
	JCPenney	877-6177
	Macy's	877-9393
	Sears	876-3200
	Target	306-5063

THEATER

	Cinema De Lux 14	878-8795

JEWELRY & ACCESSORIES

UL	Belden Jewelers	876-8420
UL	Claire's Boutique	878-9432
	Fossil	Macy's, JCPenney
UL	Gordon's Jewelers	874-0770
UL	Greenwich Avenue Too	783-1991
LL	Kay Jewelers	877-4668
LL	Laila Rowe	878-2169
LL	M & J Boutique	877-0761
UL	Michael Matthew Jewelers	882-0340
UL	Piercing Pagoda	878-4956
LL	Plumb Gold	874-5381
UL	Port of Call	874-5716
	Timex	JCPenney
UL	Whitehall Jewellers	874-5984
UL	Zales	874-1507

SHOES

UL	Aldo	
UL	American Eagle Outfitters	876-2347
UL	Bakers Shoes	877-7858
UL	Bandolino	878-0895
LL	Journeys	878-4258
LL	Payless ShoeSource/Kids	874-5750
UL	Underground Station	874-4451

CHILDREN'S APPAREL

UL	Baby Gap	878-5997
UL	Baby Place	783-1719
UL	The Children's Place	783-1719
UL	The Disney Store	877-8400
LL	Diva Kidz	878-1751
UL	Gap Kids	878-5997
UL	Gymboree	877-4286
UL	H & M	878-4769
UL	Limited Too	876-2096
LL	Rave Girl	878-1484
UL	Strasburg Children	783-1516

WOMEN'S APPAREL

UL	Abercrombie & Fitch	878-6361
UL	Aeropostale	876-2459
LL	Against All Odds	876-0483
UL	American Eagle Outfitters	876-2347
LL	Bella	783-1181
LL	Bus Stop	877-2127
UL	Charlotte Russe	877-8859
UL	Christopher & Banks	878-8117
LL	Deb Shop	877-3118
UL	Delia's	783-1629
	Evon Picone	JCPenney
UL	Express	878-5567
LL	Forever 21	876-7994
UL	Gap	878-5997
UL	Greenwich Avenue Too	783-1991
UL	H & M	878-4769
UL	Hollister	877-6900
UL	Hot Topic	877-7335
	Jones NY	JCPenney
LL	Laila Rowe	878-2169
	Lands End	Sears
UL	Lane Bryant/Cacique	783-1447
UL	Lola	878-9741
UL	Motherhood Maternity	878-5430
LL	New York & Company	876-8044
	Tommy Hilfiger	Macy's
UL	Torrid Plus Sizes	878-5460
UL	Urban Behavior	783-0452
LL	Vantage Leather	874-4052
UL	Victoria's Secret	878-9243

MEN'S APPAREL

UL	Abercrombie & Fitch	878-6361
UL	Aeropostale	876-2459
LL	Against All Odds	876-0483
UL	American Eagle Outfitters	876-2347
LL	Bus Stop	877-2127
LL	Classic Men's Wear	878-3211
	Dockers	Macy's
UL	Express Men	878-0891
UL	Gap	878-5997
UL	H & M	878-4769
UL	Hollister	877-6900
UL	Hot Topic	877-7335
UL	Pietro's Tuxedo	877-6599
LL	Vantage Leather	874-4052

SPECIALTY APPAREL

UL	Abercrombie & Fitch	878-6361
UL	Aeropostale	876-2459
UL	American Eagle Outfitters	876-2347
UL	The Disney Store	877-8400
UL	Gap	878-5997
UL	Hot Topic	877-7335
LL	Impulse T-Shirts	878-5285
LL	Journeys	878-4258
UL	Kani Leather Goods	876-1929
	Levis	Macy's, Sears
UL	Lids	876-8263
LL	Pac Sun	876-0064
UL	Sally & John Leather	783-3001
UL	Strasburg Children	783-1516
UL	Underground Station	874-4451
LL	Vantage Leather	874-4052
UL	Wilson's Leather	878-1181

ATHLETIC APPAREL & EQUIPMENT

LL	BC Sports	Coming Soon
LL	Champs	877-8124
LL	Finish Line	878-0171
UL	Footaction USA	876-8508
LL	Foot Locker	878-1858
LL	Footloose	877-4990
UL	Front Row	876-8135
LL	Lady Foot Locker	878-1609
UL	Lids	876-8263
LL	Sports & Collectibles	874-3022
UL	Steve & Barry's	878-3562

HOME FURNISHINGS & ACCESSORIES

LL	As Seen On TV	877-3427
	Emeril Cookware	Macy's, JCPenney
	Kitchen Aid	Sears
	Krups	Macy's
	Lenox	Macy's
	Macy's Furniture Store	877-9393
UL	Savannah Candle	876-9000
UL	Shiki	783-1575

WIRELESS COMMUNICATION & SERVICES

UL	Cingular Wireless	874-3000
	Cingular Wireless	301-6360
UL	Nextel	874-0046
UL	Sprint	878-9947
LL	The Mobile Solution	876-0165
LL	The Mobile Solution	
LL	Radio Shack	874-8843
UL	Verizon Wireless	878-0133
UL	Wireless Resources	877-1300

MUSIC & VIDEO

LL	Borders Books & Music	878-3333
LL	EBX	878-1871
UL	Electronics Boutique	874-5057
LL	fye (For Your Entertainment)	876-2785
LL	Radio Shack	874-8843

PHOTOGRAPHY SERVICES

	JCPenney Portrait Studio	878-6283
LL	Kiddie Kandids	301-9068
UL	The Picture People	874-7837
	Sears Portrait Studio	876-3269

HEALTH & BEAUTY

LL	As Seen On TV	877-3427
UL	Bath & Body Works	882-8889
LL	Beauty Plus Salon	876-7600
UL	Body Shop, The	878-5064
UL	Cacique (KÁ-SEEK)	783-1447
UL	Classic Nails	
	Clinique	Macy's
LL	David Ryan Salon	878-3529
UL	General Nutrition Center	783-1879
LL	Happy Nails	878-9308
	JCPenney Styling Salon	877-7494
LL	La Parfumerie	874-2447
UL	Mastercuts	876-7788
UL	Nail Pro	878-7779
	Origins	Macy's
UL	Trade Secret	876-2442
UL	Victoria's Secret Beauty	878-9243
LL	Vitamin World	876-8691

OPTICAL

LL	Cohen's Fashion Optical	878-2020
	JCPenney Optical	876-7740
UL	Lenscrafters	878-8511
	Sears Optical	876-7005
UL	Solstice	878-8200
LL	Sunglass Hut	876-2484

PETS

LL	Pet Company	874-3511

TOYS & HOBBIES

LL	BC Sports	Coming Soon
UL	The Disney Store	877-8400
UL	Game Time	878-7687
UL	KB Toys	Coming Soon
LL	Spare Room Sports Cards	874-1490
LL	Sports & Collectibles	874-3022

BOOKS, CARDS, & GIFTS

LL	As Seen On TV	877-3427
UL	AZ News	882-8219
LL	Borders Books & Music	878-3333
UL	Dollar N' Things	878-0181
UL	Miller's Hallmark	874-1569
LL	Sam's $ Mart	877-8498
UL	Savannah Candle	876-9000
LL	Spencer Gifts	874-3999
UL	Things Remembered	878-9470

TOBACCO

UL	AZ News	882-8219
UL	Tobacco Road	877-1957

RESTAURANTS

UL	The Blue Turtle	878-3183
UL	Buffalo Wild Wings	877-9453
UL	Knickerbocker's	878-8700
LL	Lin's	Coming Soon
	Moe's Southwest Grill	Coming Soon
UL	Panera Bread	874-1721

FOOD COURT

FC 3	Bourbon Street Cajun Café	301-9090
FC 2	Café Europa	
FC 1	Johnny Rockets	878-4220
FC 7	Little Tokyo	
FC 12	Papaya King	882-0700
FC 11	Sbarro	874-9587
FC 5	Subway	874-4425
FC 4	Taco Bell	Coming Soon
FC 8	Texas BBQ Factory	
FC 10	Yeung's Lotus Express	783-0880

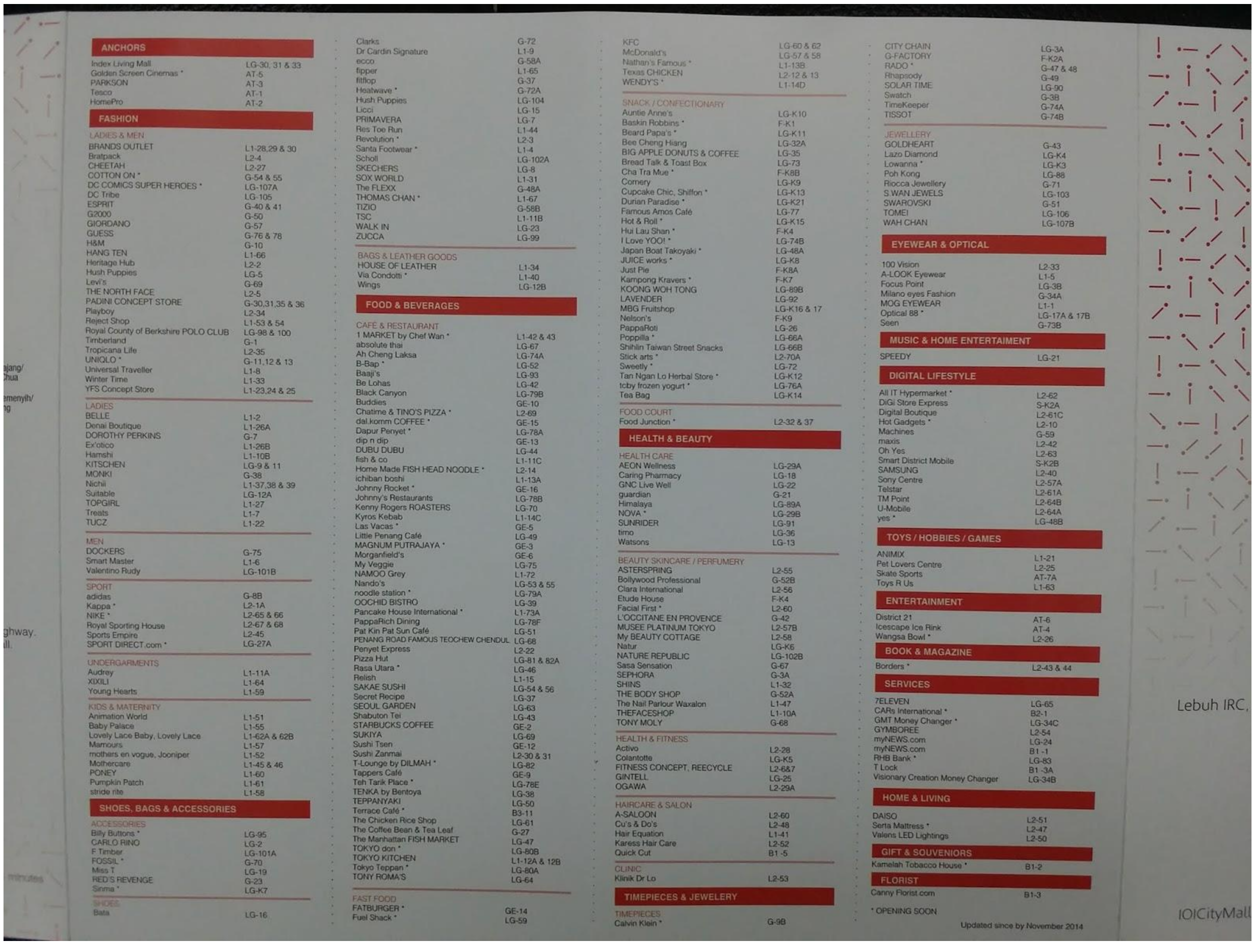
EATERIES

UL	The Blue Turtle	878-3183
UL	Buffalo Wild Wings	877-9453
LL	Chocolate Gallery	878-4255
LL	Cinnabon	878-0672
UL	Cold Stone Creamery	783-0860
LL	Dippin Dots	
UL	Gloria Jean's Coffee	874-5700
UL	Knickerbocker's	878-8700
LL	Lin's	Coming Soon
	Moe's Southwest Grill	Coming Soon
UL	Mrs. Field's Cookies	874-5695
UL	Orange Julius/Dairy Queen	878-1415
UL	Panera Bread	874-1721
LL	Pretzel Time	877-5269
UL	Rocky Mountain Chocolate Factory	882-1155
B	Starbucks	878-1865
UL	Wetzel's Pretzels	783-9018

SERVICES

UL	Air Force Recruiting Office	878-4380
B	Bank of America	882-7040
LL	Connecticut Dental Care	878-8000
	Family Lounge	
LL	Harrigan Insurance and Financial Services	877-1570
	Management Office	882-7086
LL	Peoples Choice/ Market Research	874-7732
	Playtown	
	Shopping Concierge Center	878-6837
	Restroom	
	ATM Machine	
	At payphones – dial *20 to reach Shopping Concierge dial *11 to reach Security	

Test Image 1



ANCHORS

Index Living Mall	LG-30, 31 & 33
Golden Screen Cinemas *	AT-5
PARKSON	AT-3
Tesco	AT-1
HomePro	AT-2

FASHION

LADIES & MEN

BRANDS OUTLET	L1-28,29 & 30
Bratpack	L2-4
CHEETAH	L2-27
COTTON ON *	G-54 & 55
DC COMICS SUPER HEROES *	LG-107A
DC Tribe	LG-105
ESPRIT	G-40 & 41
G2000	G-50
GIORDANO	G-57
GUESS	G-76 & 78
H&M	G-10
HANG TEN	L1-66
Heritage Hub	L2-2
Hush Puppies	LG-5
Levi's	G-69
THE NORTH FACE	L2-5
PADINI CONCEPT STORE	G-30,31,35 & 36
Playboy	L2-34
Reject Shop	L1-53 & 54
Royal County of Berkshire POLO CLUB	LG-98 & 100
Timberland	G-1
Tropicana Life	L2-35
UNIQLO *	G-11,12 & 13
Universal Traveller	L1-8
Winter Time	L1-33
YFS Concept Store	L1-23,24 & 25

LADIES

BELLE	L1-2
Denai Boutique	L1-26A
DOROTHY PERKINS	G-7
Ex'otico	L1-26B
Hamshi	L1-10B
KITSCHEN	LG-9 & 11
MONKI	G-38
Nichii	L1-37,38 & 39
Suitable	LG-12A
TOPGIRL	L1-27
Treats	L1-7
TUCZ	L1-22

MEN

DOCKERS	G-75
Smart Master	L1-6
Valentino Rudy	LG-101B

SPORT

adidas	G-8B
Kappa *	L2-1A
NIKE *	L2-65 & 66
Royal Sporting House	L2-67 & 68
Sports Empire	L2-45
SPORT DIRECT.com *	LG-27A

UNDERGARMENTS

Audrey	L1-11A
XIXILI	L1-64
Young Hearts	L1-59

KIDS & MATERNITY

Animation World	L1-51
Baby Palace	L1-55
Lovely Lace Baby, Lovely Lace	L1-62A & 62B
Mamours	L1-57
mothers en vogue, Jooniper	L1-52
Mothercare	L1-45 & 46
PONEY	L1-60
Pumpkin Patch	L1-61
stride rite	L1-58

SHOES, BAGS & ACCESSORIES

ACCESSORIES

Billy Buttons *	LG-95
CARLO RINO	LG-2
F Timber	LG-101A
FOSSIL *	G-70
Miss T	LG-19
RED'S REVENGE	G-23
Sinma *	LG-K7

SHOES

Bata	LG-16
Clarks	G-72
Dr Cardin Signature	L1-9
ecco	G-58A
fipper	L1-65
fitflop	G-37
Heatwave *	G-72A
Hush Puppies	LG-104
Licci	LG-15
PRIMAVERA	LG-7
Res Toe Run	L1-44
Revolution *	L2-3
Santa Footwear *	L1-4
Scholl	LG-102A
SKECHERS	LG-8
SOX WORLD	L1-31
The FLEXX	G-48A
THOMAS CHAN *	L1-67
TIZIO	G-58B
TSC	L1-11B
WALK IN	LG-23
ZUCCA	LG-99

BAGS & LEATHER GOODS

HOUSE OF LEATHER	L1-34
Via Condotti *	L1-40
Wings	LG-12B

FOOD & BEVERAGES

CAFÉ & RESTAURANT

1 MARKET by Chef Wan *	L1-42 & 43
absolute thai	LG-67
Ah Cheng Laksa	LG-74A
B-Bap *	LG-52
Baaji's	LG-93
Be Lohas	LG-42
Black Canyon	LG-79B
Buddies	GE-10
Chatime & TINO'S PIZZA *	L2-69
dal.komm COFFEE *	GE-15
Dapur Penyet *	LG-78A
dip n dip	GE-13
DUBU DUBU	LG-44
fish & co	L1-11C
Home Made FISH HEAD NOODLE *	L2-14
ichiban boshi	L1-13A
Johnny Rocket *	GE-16
Johnny's Restaurants	LG-78B
Kenny Rogers ROASTERS	LG-70
Kyros Kebab	L1-14C
Las Vacas *	GE-5
Little Penang Café	LG-49
MAGNUM PUTRAJAYA *	GE-3
Morganfield's	GE-6
My Veggie	LG-75
NAMOO Grey	L1-72
Nando's	LG-53 & 55
noodle station *	LG-79A
OOCHID BISTRO	LG-39
Pancake House International *	L1-73A
PappaRich Dining	LG-78F
Pat Kin Pat Sun Café	LG-51
PENANG ROAD FAMOUS TEOCHEW CHENDUL	LG-68
Penyet Express	L2-22
Pizza Hut	LG-81 & 82A
Rasa Utara *	LG-46
Relish	L1-15
SAKAE SUSHI	LG-54 & 56
Secret Recipe	LG-37
SEOUL GARDEN	LG-63
Shabuton Tei	LG-43
STARBUCKS COFFEE	GE-2
SUKIYA	LG-69
Sushi Tsen	GE-12
Sushi Zanmai	L2-30 & 31
T-Lounge by DILMAH *	LG-82
Tappers Café	GE-9
Teh Tarik Place *	LG-78E
TENKA by Bentoya	LG-38
TEPPANYAKI	LG-50
Terrace Café *	B3-11
The Chicken Rice Shop	LG-61
The Coffee Bean & Tea Leaf	G-27
The Manhattan FISH MARKET	LG-47
TOKYO don *	LG-80B
TOKYO KITCHEN	L1-12A & 12B
Tokyo Teppan *	LG-80A
TONY ROMA'S	LG-64

FAST FOOD

FATBURGER *	GE-14
Fuel Shack *	LG-59
KFC	LG-60 & 62
McDonald's	LG-57 & 58
Nathan's Famous *	L1-13B
Texas CHICKEN	L2-12 & 13
WENDY'S *	L1-14D

SNACK / CONFECTIONARY

Auntie Anne's	LG-K10
Baskin Robbins *	F-K1
Beard Papa's *	LG-K11
Bee Cheng Hiang	LG-32A
BIG APPLE DONUTS & COFFEE	LG-35
Bread Talk & Toast Box	LG-73
Cha Tra Mue *	F-K8B
Comery	LG-K9
Cupcake Chic, Shiffon *	LG-K13
Durian Paradise *	LG-K21
Famous Amos Café	LG-77
Hot & Roll *	LG-K15
Hui Lau Shan *	F-K4
I Love YOO! *	LG-74B
Japan Boat Takoyaki *	LG-48A
JUICE works *	LG-K8
Just Pie	F-K8A
Kampong Kravers *	F-K7
KOONG WOH TONG	LG-89B
LAVENDER	LG-92
MBG Fruitshop	LG-K16 & 17
Nelson's	F-K9
PappaRoti	LG-26
Poppilla *	LG-66A
Shihlin Taiwan Street Snacks	LG-66B
Stick arts *	L2-70A
Sweetly *	LG-72
Tan Ngan Lo Herbal Store *	LG-K12
tcby frozen yogurt *	LG-76A
Tea Bag	LG-K14

FOOD COURT

Food Junction *	L2-32 & 37

HEALTH & BEAUTY

HEALTH CARE

AEON Wellness	LG-29A
Caring Pharmacy	LG-18
GNC Live Well	LG-22
guardian	G-21
Himalaya	LG-89A
NOVA *	LG-29B
SUNRIDER	LG-91
timo	LG-36
Watsons	LG-13

BEAUTY SKINCARE / PERFUMERY

ASTERSPRING	L2-55
Bollywood Professional	G-52B
Clara International	L2-56
Etude House	F-K4
Facial First *	L2-60
L'OCCITANE EN PROVENCE	G-42
MUSEE PLATINUM TOKYO	L2-57B
My BEAUTY COTTAGE	L2-58
Natur	LG-K6
NATURE REPUBLIC	LG-102B
Sasa Sensation	G-67
SEPHORA	G-3A
SHINS	L1-32
THE BODY SHOP	G-52A
The Nail Parlour Waxalon	L1-47
THEFACESHOP	L1-10A
TONY MOLY	G-68

HEALTH & FITNESS

Activo	L2-28
Colantotte	LG-K5
FITNESS CONCEPT, REECYCLE	L2-6&7
GINTELL	LG-25
OGAWA	L2-29A

HAIRCARE & SALON

A-SALOON	L2-60
Cu's & Do's	L2-48
Hair Equation	L1-41
Karess Hair Care	L2-52
Quick Cut	B1-5

CLINIC

Klinik Dr Lo	L2-53

TIMEPIECES & JEWELERY

TIMEPIECES

Calvin Klein *	G-9B
CITY CHAIN	LG-3A
G-FACTORY	F-K2A
RADO *	G-47 & 48
Rhapsody	G-49
SOLAR TIME	LG-90
Swatch	G-3B
TimeKeeper	G-74A
TISSOT	G-74B

JEWELLERY

GOLDHEART	G-43
Lazo Diamond	LG-K4
Lowanna *	LG-K3
Poh Kong	LG-88
Riocca Jewellery	G-71
S.WAN JEWELS	LG-103
SWAROVSKI	G-51
TOMEI	LG-106
WAH CHAN	LG-107B

EYEWEAR & OPTICAL

100 Vision	L2-33
A-LOOK Eyewear	L1-5
Focus Point	LG-3B
Milano eyes Fashion	G-34A
MOG EYEWEAR	L1-1
Optical 88 *	LG-17A & 17B
Seen	G-73B

MUSIC & HOME ENTERTAIMENT

SPEEDY	LG-21

DIGITAL LIFESTYLE

All IT Hypermarket *	L2-62
DiGi Store Express	S-K2A
Digital Boutique	L2-61C
Hot Gadgets *	L2-10
Machines	G-59
maxis	L2-42
Oh Yes	L2-63
Smart District Mobile	S-K2B
SAMSUNG	L2-40
Sony Centre	L2-57A
Telstar	L2-61A
TM Point	L2-64B
U-Mobile	L2-64A
yes *	LG-48B

TOYS / HOBBIES / GAMES

ANIMIX	L1-21
Pet Lovers Centre	L2-25
Skate Sports	AT-7A
Toys R Us	L1-63

ENTERTAINMENT

District 21	AT-6
Icescape Ice Rink	AT-4
Wangsa Bowl *	L2-26

BOOK & MAGAZINE

Borders *	L2-43 & 44

SERVICES

7ELEVEN	LG-65
CARs International *	B2-1
GMT Money Changer *	LG-34C
GYMBOREE	L2-54
myNEWS.com	LG-24
myNEWS.com	B1 -1
RHB Bank *	LG-83
T Lock	B1 -3A
Visionary Creation Money Changer	LG-34B

HOME & LIVING

DAISO	L2-51
Serta Mattress *	L2-47
Valens LED Lightings	L2-50

GIFT & SOUVENIORS

Kamelah Tobacco House *	B1-2

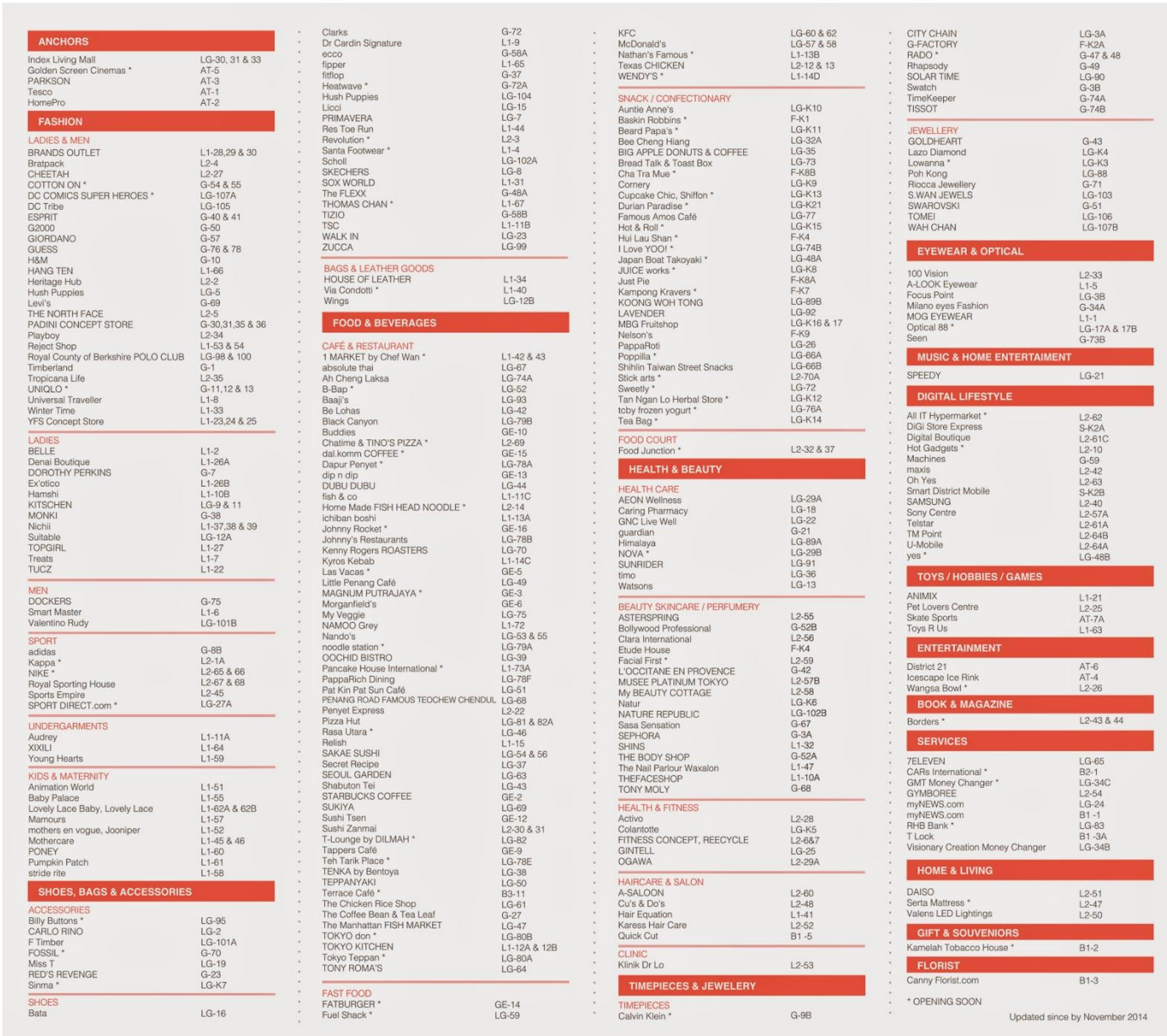
FLORIST

Canny Florist.com	B1-3

* OPENING SOON

Updated since by November 2014

Test Image 2



ANCHORS

Index Living Mall	LG-30, 31 & 33
Golden Screen Cinemas *	AT-5
PARKSON	AT-3
Tesco	AT-1
HomePro	AT-2

FASHION

LADIES & MEN

BRANDS OUTLET	L1-28,29 & 30
Bratpack	L2-4
CHEETAH	L2-27
COTTON ON *	G-54 & 55
DC COMICS SUPER HEROES *	LG-107A
DC Tribe	LG-105
ESPRIT	G-40 & 41
G2000	G-50
GIORDANO	G-57
GUESS	G-76 & 78
H&M	G-10
HANG TEN	L1-66
Heritage Hub	L2-2
Hush Puppies	LG-5
Levi's	G-69
THE NORTH FACE	L2-5
PADINI CONCEPT STORE	G-30,31,35 & 36
Playboy	L2-34
Reject Shop	L1-53 & 54
Royal County of Berkshire POLO CLUB	LG-98 & 100
Timberland	G-1
Tropicana Life	L2-35
UNIQLO *	G-11,12 & 13
Universal Traveller	L1-8
Winter Time	L1-33
YFS Concept Store	L1-23,24 & 25

LADIES

BELLE	L1-2
Denai Boutique	L1-26A
DOROTHY PERKINS	G-7
Ex'otico	L1-26B
Hamshi	L1-10B
KITSCHEN	LG-9 & 11
MONKI	G-38
Nichii	L1-37,38 & 39
Suitable	LG-12A
TOPGIRL	L1-27
Treats	L1-7
TUCZ	L1-22

MEN

DOCKERS	G-75
Smart Master	L1-6
Valentino Rudy	LG-101B

SPORT

adidas	G-8B
Kappa *	L2-1A
NIKE *	L2-65 & 66
Royal Sporting House	L2-67 & 68
Sports Empire	L2-45
SPORT DIRECT.com *	LG-27A

UNDERGARMENTS

Audrey	L1-11A
XIXILI	L1-64
Young Hearts	L1-59

KIDS & MATERNITY

Animation World	L1-51
Baby Palace	L1-55
Lovely Lace Baby, Lovely Lace	L1-62A & 62B
Mamours	L1-57
mothers en vogue, Jooniper	L1-52
Mothercare	L1-45 & 46
PONEY	L1-60
Pumpkin Patch	L1-61
stride rite	L1-58

SHOES, BAGS & ACCESSORIES

ACCESSORIES

Billy Buttons *	LG-95
CARLO RINO	LG-2
F Timber	LG-101A
FOSSIL *	G-70
Miss T	LG-19
RED'S REVENGE	G-23
Sinma *	LG-K7

SHOES

Bata	LG-16
Clarks	G-72
Dr Cardin Signature	L1-9
ecco	G-58A
fipper	L1-65
fitflop	G-37
Heatwave *	G-72A
Hush Puppies	LG-104
Licci	LG-15
PRIMAVERA	LG-7
Res Toe Run	L1-44
Revolution *	L2-3
Santa Footwear *	L1-4
Scholl	LG-102A
SKECHERS	LG-8
SOX WORLD	L1-31
The FLEXX	G-48A
THOMAS CHAN *	L1-67
TIZIO	G-58B
TSC	L1-11B
WALK IN	LG-23
ZUCCA	LG-99

BAGS & LEATHER GOODS

HOUSE OF LEATHER	L1-34
Via Condotti *	L1-40
Wings	LG-12B

FOOD & BEVERAGES

CAFÉ & RESTAURANT

1 MARKET by Chef Wan *	L1-42 & 43
absolute thai	LG-67
Ah Cheng Laksa	LG-74A
B-Bap *	LG-52
Baaji's	LG-93
Be Lohas	LG-42
Black Canyon	LG-79B
Buddies	GE-10
Chatime & TINO'S PIZZA *	L2-69
dal.komm COFFEE *	GE-15
Dapur Penyet *	LG-78A
dip n dip	GE-13
DUBU DUBU	LG-44
fish & co	L1-11C
Home Made FISH HEAD NOODLE *	L2-14
ichiban boshi	L1-13A
Johnny Rocket *	GE-16
Johnny's Restaurants	LG-78B
Kenny Rogers ROASTERS	LG-70
Kyros Kebab	L1-14C
Las Vacas *	GE-5
Little Penang Café	LG-49
MAGNUM PUTRAJAYA *	GE-3
Morganfield's	GE-6
My Veggie	LG-75
NAMOO Grey	L1-72
Nando's	LG-53 & 55
noodle station *	LG-79A
OOCHID BISTRO	LG-39
Pancake House International *	L1-73A
PappaRich Dining	LG-78F
Pat Kin Pat Sun Café	LG-51
PENANG ROAD FAMOUS TEOCHEW CHENDUL	LG-68
Penyet Express	L2-22
Pizza Hut	LG-81 & 82A
Rasa Utara *	LG-46
Relish	L1-15
SAKAE SUSHI	LG-54 & 56
Secret Recipe	LG-37
SEOUL GARDEN	LG-63
Shabuton Tei	LG-43
STARBUCKS COFFEE	GE-2
SUKIYA	LG-69
Sushi Tsen	GE-12
Sushi Zanmai	L2-30 & 31
T-Lounge by DILMAH *	LG-82
Tappers Café	GE-9
Teh Tarik Place *	LG-78E
TENKA by Bentoya	LG-38
TEPPANYAKI	LG-50
Terrace Café *	B3-11
The Chicken Rice Shop	LG-61
The Coffee Bean & Tea Leaf	G-27
The Manhattan FISH MARKET	LG-47
TOKYO don *	LG-80B
TOKYO KITCHEN	L1-12A & 12B
Tokyo Teppan *	LG-80A
TONY ROMA'S	LG-64

FAST FOOD

FATBURGER *	GE-14
Fuel Shack *	LG-59
KFC	LG-60 & 62
McDonald's	LG-57 & 58
Nathan's Famous *	L1-13B
Texas CHICKEN	L2-12 & 13
WENDY'S *	L1-14D

SNACK / CONFECTIONARY

Auntie Anne's	LG-K10
Baskin Robbins *	F-K1
Beard Papa's *	LG-K11
Bee Cheng Hiang	LG-32A
BIG APPLE DONUTS & COFFEE	LG-35
Bread Talk & Toast Box	LG-73
Cha Tra Mue *	F-K8B
Cornery	LG-K9
Cupcake Chic, Shiffon *	LG-K13
Durian Paradise *	LG-K21
Famous Amos Café	LG-77
Hot & Roll *	LG-K15
Hui Lau Shan *	F-K4
I Love YOO! *	LG-74B
Japan Boat Takoyaki *	LG-48A
JUICE works *	LG-K8
Just Pie	F-K8A
Kampong Kravers *	F-K7
KOONG WOH TONG	LG-89B
LAVENDER	LG-92
MBG Fruitshop	LG-K16 & 17
Nelson's	F-K9
PappaRoti	LG-26
Poppilla *	LG-66A
Shihlin Taiwan Street Snacks	LG-66B
Stick arts *	L2-70A
Sweetly *	LG-72
Tan Ngan Lo Herbal Store *	LG-K12
tcby frozen yogurt *	LG-76A
Tea Bag *	LG-K14

FOOD COURT

Food Junction *	L2-32 & 37

HEALTH & BEAUTY

HEALTH CARE

AEON Wellness	LG-29A
Caring Pharmacy	LG-18
GNC Live Well	LG-22
guardian	G-21
Himalaya	LG-89A
NOVA *	LG-29B
SUNRIDER	LG-91
timo	LG-36
Watsons	LG-13

BEAUTY SKINCARE / PERFUMERY

ASTERSPRING	L2-55
Bollywood Professional	G-52B
Clara International	L2-56
Etude House	F-K4
Facial First *	L2-59
L'OCCITANE EN PROVENCE	G-42
MUSEE PLATINUM TOKYO	L2-57B
My BEAUTY COTTAGE	L2-58
Natur	LG-K6
NATURE REPUBLIC	LG-102B
Sasa Sensation	G-67
SEPHORA	G-3A
SHINS	L1-32
THE BODY SHOP	G-52A
The Nail Parlour Waxalon	L1-47
THEFACESHOP	L1-10A
TONY MOLY	G-68

HEALTH & FITNESS

Activo	L2-28
Colantotte	LG-K5
FITNESS CONCEPT, REECYCLE	L2-6&7
GINTELL	LG-25
OGAWA	L2-29A

HAIRCARE & SALON

A-SALOON	L2-60
Cu's & Do's	L2-48
Hair Equation	L1-41
Karess Hair Care	L2-52
Quick Cut	B1 -5

CLINIC

Klinik Dr Lo	L2-53

TIMEPIECES & JEWELERY

TIMEPIECES

Calvin Klein *	G-9B
CITY CHAIN	LG-3A
G-FACTORY	F-K2A
RADO *	G-47 & 48
Rhapsody	G-49
SOLAR TIME	LG-90
Swatch	G-3B
TimeKeeper	G-74A
TISSOT	G-74B

JEWELLERY

GOLDHEART	G-43
Lazo Diamond	LG-K4
Lowanna *	LG-K3
Poh Kong	LG-88
Riocca Jewellery	G-71
S.WAN JEWELS	LG-103
SWAROVSKI	G-51
TOMEI	LG-106
WAH CHAN	LG-107B

EYEWEAR & OPTICAL

100 Vision	L2-33
A-LOOK Eyewear	L1-5
Focus Point	LG-3B
Milano eyes Fashion	G-34A
MOG EYEWEAR	L1-1
Optical 88 *	LG-17A & 17B
Seen	G-73B

MUSIC & HOME ENTERTAIMENT

SPEEDY	LG-21

DIGITAL LIFESTYLE

All IT Hypermarket *	L2-62
DiGi Store Express	S-K2A
Digital Boutique	L2-61C
Hot Gadgets *	L2-10
Machines	G-59
maxis	L2-42
Oh Yes	L2-63
Smart District Mobile	S-K2B
SAMSUNG	L2-40
Sony Centre	L2-57A
Telstar	L2-61A
TM Point	L2-64B
U-Mobile	L2-64A
yes *	LG-48B

TOYS / HOBBIES / GAMES

ANIMIX	L1-21
Pet Lovers Centre	L2-25
Skate Sports	AT-7A
Toys R Us	L1-63

ENTERTAINMENT

District 21	AT-6
Icescape Ice Rink	AT-4
Wangsa Bowl *	L2-26

BOOK & MAGAZINE

Borders *	L2-43 & 44

SERVICES

7ELEVEN	LG-65
CARs International *	B2-1
GMT Money Changer *	LG-34C
GYMBOREE	L2-54
myNEWS.com	LG-24
myNEWS.com	B1 -1
RHB Bank *	LG-83
T Lock	B1 -3A
Visionary Creation Money Changer	LG-34B

HOME & LIVING

DAISO	L2-51
Serta Mattress *	L2-47
Valens LED Lightings	L2-50

GIFT & SOUVENIORS

Kamelah Tobacco House *	B1-2

FLORIST

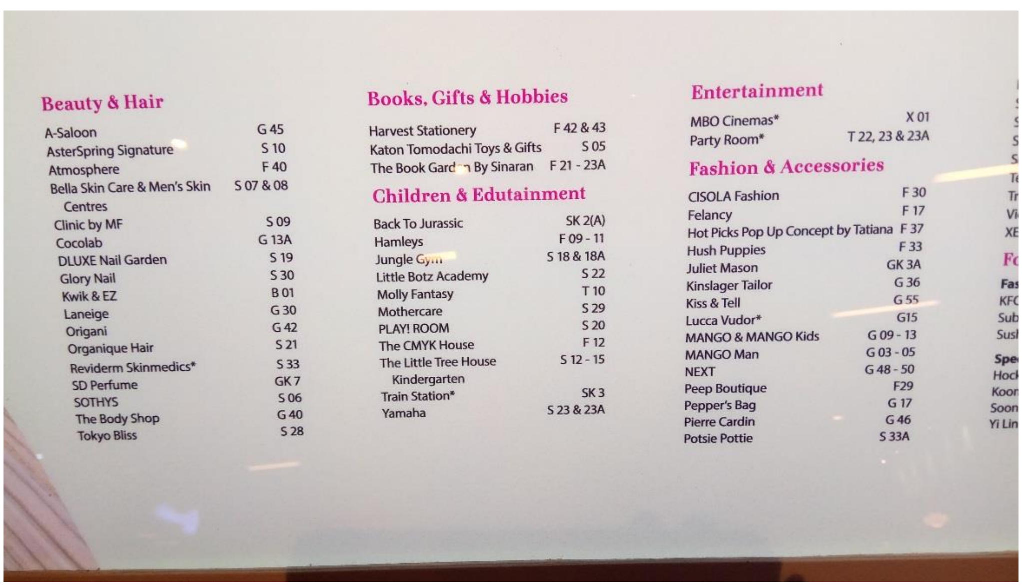
Canny Florist.com	B1-3

* OPENING SOON

Updated since by November 2014

Test Image 3

Test Image 4

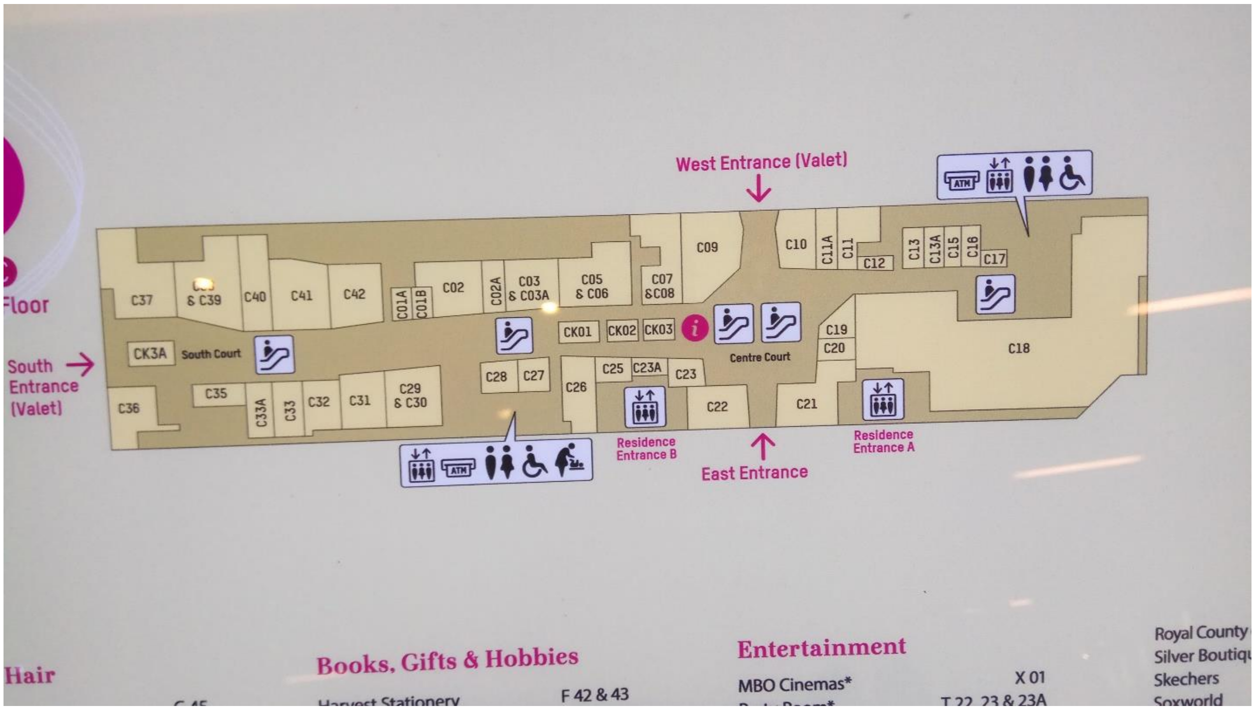


Test Image 5

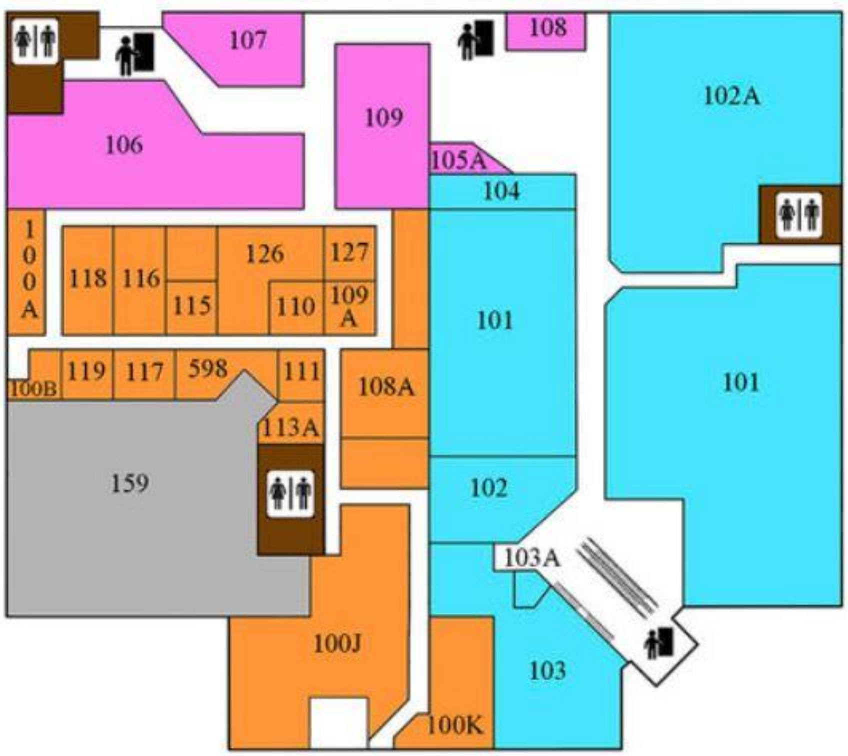
McDonalds	C35
Wendy’s	C36
Manicure	C28
Texas Chicken	C27
Parkson	C18
Jusco	C21
KFC	CK3A
CMY	C32
Nandos	C37
Tangs	CK01
Tokyo Secret	C10
Secret Recipe	C12
Uniqlo	C17
Hamley’s	C25

Test Image 6

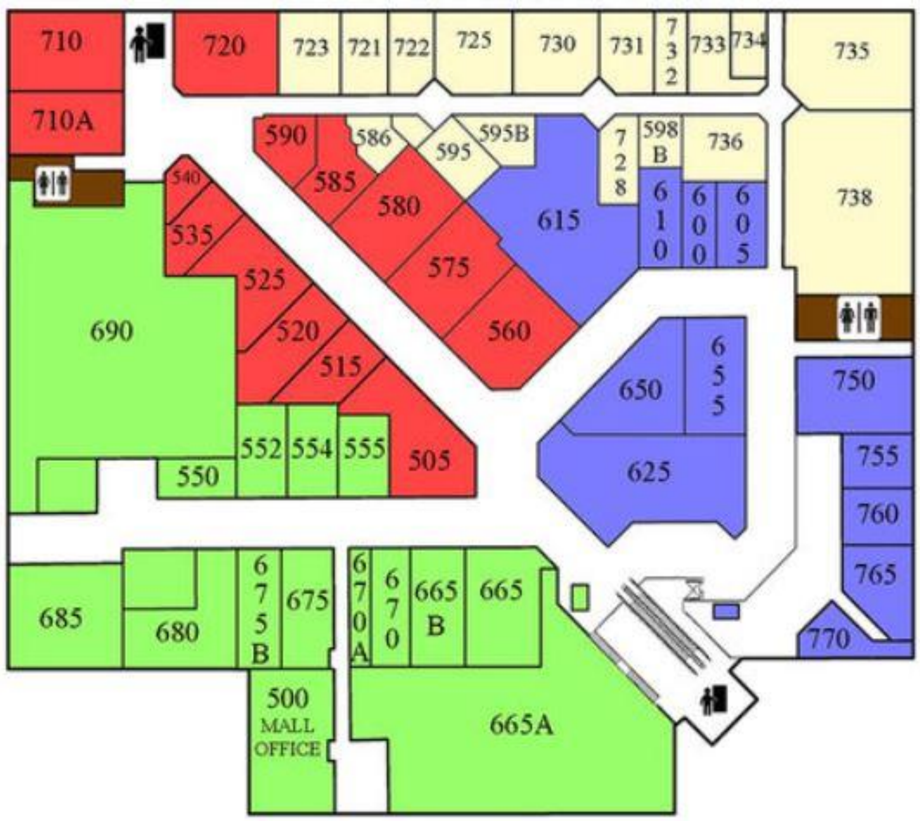
Map Test Images (Phase 3)



Test Image 1



Test Image 2



710
720
723
721
722
725
730
731
732
733
734
735
710A
590
586
595
595B
728
598 B
736
738
540
585
580
615
610
600
605
535
525
575
560
690
520
515
650
655
750
552
554
555
505
550
625
755
760
765
685
680
675 B
675
670 A
670
665 B
665
770
500
MALL
OFFICE
665A

Test Image 3