Intelligent Lighting for Wide Spaces

João Correia
Instituto Superior Técnico
joaocruzcorreia@tecnico.ulisboa.pt

Abstract

The worldwide energy consumption has been increasing in recent years and since much of the energy comes from non-renewable sources, this increase in consumption may have serious environmental consequences in the near future. Electricity is the form of energy whose consumption has grown the most, so it is necessary to find more efficient technologies. This paper proposes an intelligent lighting system for indoor large-scale spaces, such as underground car parks. The system will consist of intelligent lighting luminaires, which will operate autonomously, and a configuration application. The proposed solution follows a distributed approach and has the particularity to only illuminate the areas where people are present, being able to determine the direction of movement of users and illuminate those areas in advance. The goal is to reduce energy consumption without decreasing the quality of lighting, offering a solution that is easy to install and offers high scalability. Since it is not feasible to implement a prototype of this system due to the high number of necessary luminaires, it was decided to develop and test the system using a simulator. In this thesis it is described the architecture of the proposed solution and are described the developments made, with emphasis on the simulator, the algorithm present in the luminaires and the configuration application. We also present the tests performed and their results, which demonstrate that the proposed solution is efficient and allows greater energy savings than other systems.

Keywords: Intelligent Luminaires, Lighting System, Energy Saving, Distributed System

1. Introduction

The world's energy consumption has been growing and is estimated to increase by 46% by 2040. This is due largely to developing countries [1]. With environmental problems and 87% of the energy generated from non-renewable energy sources [2], it is necessary to find alternatives. Renewable energy is one of the alternatives, however their growth is not fast enough [1].

Electricity is the form of energy whose consumption has grown the most [3]. In recent years there has been an increased focus in reasearching technologies that use electricity more efficiently. One application of electricity is to light indoor large-scale spaces. Many of those spaces are using very inefficient lighting systems, that are usually turned on from opening time to closing time. Energy is wasted by lighting a much larger area than the required.

This paper proposes an intelligent lighting system for indoor large-scale spaces that allows energy savings without compromising lighting quality. To this end, smart luminaires are used, consisting of dimmable LED lamps, a motion sensor, a radio communication module and a microprocessor. There is also a configuration application that allows to manage and configure the various parameters of intelligent luminaires. The proposed system is a distributed solution, which facilitates the installation and system scalability.

The operation of the system depends on the algorithm present in each intelligent luminaire. The system operates based on motion sensors and the information received from the surrounding luminaires. When it detects motion, all luminaires within a certain radius light up. The system also seeks to determine the user's travel direction and lights the lamps in this direction, in anticipation, to a certain distance. After a few seconds, if no movement is detected, the lights will dim until they reach a minimum intensity or go out. A key objective of this paper is to explore the system's ability to illuminate the area around the user and correctly identify the user's direction. Since the system is intended for large spaces, it would be necessary to acquire a large number of components, which would have significant costs. Alternatively, the solution will be developed and tested using a simulator.

2. Related Work

Yoshiura et al. [4] propose the concept of a street lighting system where the lights turn on before pedestrians or vehicles pass and turn off or reduce the intensity otherwise. This system consists of three components. The lamp unit consisting of a LED array, light sensor, motion sensor, controller and communication device. The LED lamp is turned on if the sensors detect motion and sends a message to other units. The sensor unit consists of the motion sensor, controller and communication device. If it detects motion, it sends a message to other units. This unit is installed at several points to ensure that the lamps are turned on before the pedestrian or vehicle passes through them. The access point, which consists of the communication and control device, is used in case the distance between the lamp units and the sensor units is too large to communicate with each other.

The Tvilight [5] is an intelligent lighting control company that specializes in outdoor sensors, wireless lighting control and lighting management software for street lighting applications. Of the various products available, the ones that standout are CityManager and Citysense. The CityManager [6] is a lighting management software that allows to monitor, manage and control the whole city lighting infrastructure. The Citysense [7] is a sensor that adjusts the intensity of light according the presence of pedestrians or vehicles. The lights dim when no one is detected and light up, illuminating the area around the area where there was detected a presence.

Smart Street Lighting (SSL) [9] is a system that turns on and off street lighting dynamically, based on the location of pedestrians and its "safety zone". The safety zone is the radius of the illuminated area. The location of each user is obtained from his smartphone, which has an application to change the size of the safety zone and periodically sends location information and configuration for the SSL server. Each lamp post is equipped with a radio device which receives information from the SSL server through a radio base station.

On paper [10] LiLong et al. present a road lighting control system. This system is based on wireless sensor networks and allows to monitor and control in real time the lighting of public roads. Each lamp has a wireless communication module, a control module and a microprocessor. The lamps are controlled remotely through a management software that runs on the host computer. It is possible to monitor the lamps, control the intensity and program the lights to change their intensity at a certain time.

On paper [11] it is presented to us an embedded video processing system that controls the light intensity in a parking lot by motion detection. The lighiting intensity is increased when moving persons or vehicles are detected, and otherwise decreased, thereby reducing power consumption. In the proposed solution dimmable LED lamps and an intelligent video sensor that can control the intensity of the lamp are used. This system uses a real-time video analysis algorithm to detect motion.

3. Proposed Solution

3.1 Functional Description

It is intended to develop an intelligent distributed lighting and control system. The proposed system targets wide spaces with a large number of lamps.

The system's intelligence resides in the luminaires. Each luminaire is an intelligent luminaire that has processing capabilities, can detect movement, exchanges information with neighboring luminaires and controls the intensity of light. They represent a node in the system and are constituted by an LED lamp, adjustable in intensity, a motion sensor, a radio communication module and a microprocessor. They have four intensities: high, medium, low and minimum. All luminaires start at minimum intensity. Intelligent luminaires are exactly equal to each other and have a matrix arrangement. Since the system works with intelligent luminaires, it simplifies its installation and changes that may be needed in the future. Another advantage is that there is no single point of failure, as in centralized systems.

The system operation is based on the use of motion sensors which detect the movement of users and works as described below:

- When motion is detected, the lamps within a certain radius from the zone when there is movement detection are lit up. This radius will be designated lighting radius. The node that detects motion is with high intensity, the nodes in the lighting radius edge are with low intensity and the remaining nodes are with medium intensity;
- When the system determines that the user is moving in one direction, the lamps that are in the predicted direction light up along a certain distance with the lighting radius. This distance will be designated anticipation distance. The nodes over this distance are with high intensity;
- If there is no movement detected, the lamps will dim until minimum intensity.

The system should work as described with multiple users in a space, which can move individually or in groups and in any direction.

3.2 System Main Components

The basic component of this system is the intelligent luminaire, which has been described above. Present in each luminaire is an algorithm responsible for system behavior. The algorithm is the main part of the system and defines the behavior of a lamp, being responsible for interpreting the information received, change the lamp state and send information to other intelligent fixtures.

There are parameters that can be changed, such as the lighting radius. For this a configuration application is used. It allows to manage and monitor the system and it runs on a computer. It is used to configure the system during installation and change system parameters.

The communication between the application configuration and the luminaires is provided by a gateway element. The gateway is responsible for sending information from the configuraion applicatio to the luminaires and vice-versa.

3.3 Implementation

The objective of this paper is the development of the the algorithm and the configuration application. The development of an actual prototype would require a large number of components, which would have high costs. Instead a simulator will be used to support the development and testing the algorithm.

A. Simulator

The simulator is where the system will be tested. With the simulator it is possible to create test environments with different dimensions and any number of nodes and users.

A simulator was developed specifically for this project. The simulator was developed in Java using the Swing framework for the GUI. Its main features are:

- Has a GUI;
- Setup the test environment;
- Set the users' paths;
- Pause and resume the simulation at any time;
- Run the simulation step-by-step;
- Change the simulation speed of execution;
- Show the obtained energy savings;
- Simulate the failure of a node;
- Show all messages exchanged between nodes and related statistics.

The developed simulator is shown in Figure 1. The simulator is controlled through various buttons and features the map area (rectangular area), where we can observe the simulation.

Before running a simulation we need to set the test environment. For this two text files are used: the map script and the user script. The map script contains the commands that defines the test environment, i.e., the format and size, number of nodes, lighting radius and anticipation distance, among other things. The user script contains the commands that control the users, indicating its initial position and its path. The two files are selected when loaded in the Map Script buttons, and User Script and are loaded into the simulator through the Load button. Then, in the map area, the nodes and users appear in their initial positions. It is considered that the disposition of nodes is matrix and that the xx and yy axes have the orientation shown in Figure 1.

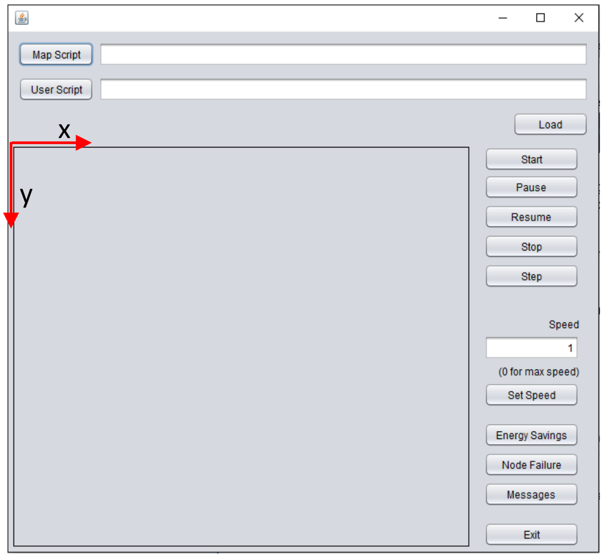

Figure 1 – Home screen of the simulador. The red arrows represent the orientation of the xx and yy axes.

It is possible to change the speed before the simulation begins. The speed value represents the time in seconds between each movement of the users. To change its value write the desired value in the Speed field and click on Set Speed. The value of the speed must be greater than or equal to 0. The value 0 is the maximum speed of the simulator, which corresponds to 0.25 seconds.

The simulation starts as soon as we click on the Start button. The simulation can be controlled using the Pause, Resume and Stop buttons, to pause, resume and stop the simulation, respectively. We can also run the simulation step by step. To do this, the simulation is paused and then Step button is clicked and the users move to their next position.

When the simulation ends a dialog box appears informing the end of the simulation. If we click on the Energy Savings button, a new window appears with information on the power units spent if the luminaires were with their max intensity, the energy spent using the system and the obtained savings. If we click the Messages button a new window appears where it shows information about the messaging and a table with all the messages sent. The information shown is the number of messages sent, the number of nodes that sent at least one message, the average number of messages sent by each node that sent at least one message and the average number of messages sent by each node that sent at least one message per unit of time. To restart the simulation, just click on Stop and Load buttons, if the simulation is still running, or just the Load button if the simulation already finished. To get out of the simulator press the Exit button.

To define the test environment, we use the map script. Although the simulator uses a matrix array, it is intended that the creation of test environments, with the available commands, had the fewest possible restrictions. It is possible to define test environments with any number of nodes and any form. It is possible to create obstacles like walls, which are represented in the simulator as a line segment, and do not affect the simulation result. It allows adding and removing nodes individually or together and change various system parameters and each node.

With the user script we can create users and set their path. Each user has an identifier, an initial position a path. The available commands allow users to move freely around the map, and even control their movement speed.

B. Algorithm

The algorithm is the main part of the system and will run in each intelligent luminaire. The algorithm must occupy the minimum resources as possible. Each node has the following characteristics:

- Has a unique identifier, defined by its coordinates;
- Each node communicates only with its neighbors, which are its adjacent nodes;
- The neighbors of each node are known *a priori*;
- The communication between nodes is made through messages.

As each node can only communicates with its neighbors, the number of directions that a node can detect are eight (up, down, left, right and diagonals). There are four intensities: minimum, low, medium and high. The minimum intensity is used when nodes do not detect movement or receive messages. The medium and low intensity are used when there was a detection, and the node is within the lighting radius. The high intensity is used when a node detects movement or is within the anticipation distance.

The communication between nodes is via messages. There are two types of messages: the LR messages (Lighting Radius) and AD messages (Anticipation Distance). The LR message is used to inform the nodes that they are inside the lighting radius and has the following fields:

- Origin: identifier of the node that detected movement;
- Timestamp of origin: timestamp on the origin;
- Sender: identifier of the node that sent the message;
- Timestamp: timestamp of when the message was sent. It is the message identifier;
- Lighting Radius.

The AD message is used to inform the nodes that are in the anticipation distance. Each message has the following fields:

- Origin: identifier of the node that detected movement;
- Timestamp of origin: timestamp on the origin;
- Sender: identifier of the node that sent the message;
- Timestamp: timestamp of when the message was sent. It is the message identifier;
- Anticipation Distance.

We have to take into account the number of messages sent when implementing the system. It was necessary to find a method of sending messages to reduce the number of messages sent without compromising the functioning of the system. On the first motion detection, the node that detected the movement sends an LR message to all its neighbors. The nodes that receive this message, send a new message to the node that is in the direction in which the message was received and the nodes in the adjacent directions. From the second movement detection, the node that detected motion sends an AD message in the predicted direction and sends a LR to the other neighbors. When a node receives an AD message, if the anticipation distance has not been traveled, a AD message in the predicted direction. For the other neighbors, except the sender, it is sent a LR message. If the anticipation distance has been traveled, the node that received the AD message sends message to all neighbors except the sender, a LR message.

Each node internally operates as a state machine. There are in total five states: Minimum, Low, Medium, and Anticipated. Each node starts in the Minimum state. The output of each state corresponds to one of the four intensities: minimum, low, medium and high. There are five inputs to change the state of a node:

- Motion Detection (D);
- AD message (AD);
- LR message with lighting radius greater than 1 (LR>1);
- LR message with lighting radius equal to 1 (RI=1);
- Timer reached zero (T).

In figure 2 we have the state machine of each node.

Each node has only the information required for the operation of the algorithm. When a node is created, it has the following information:

- The coordinates that serve as identifier;
- The initial state, which is the minimum state;
- The intensity, initialized with the minimum intensity;
- The timeout, which is the time that the node is in a state before moving to the next state;
- The lighting radius;
- The anticipation istance;
- The list of neighbors;

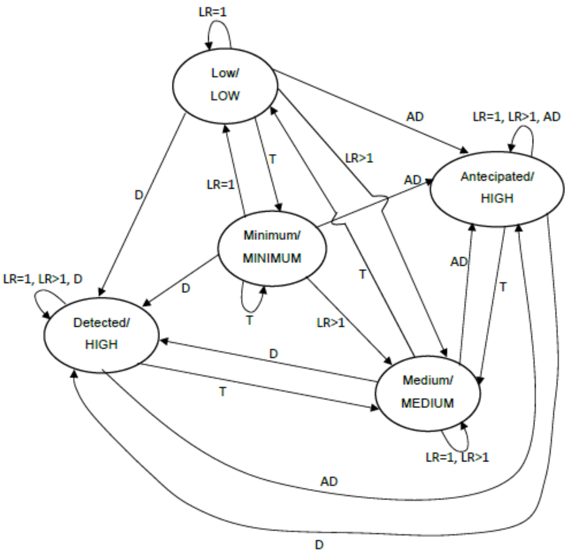

Figure 2 – State Machine

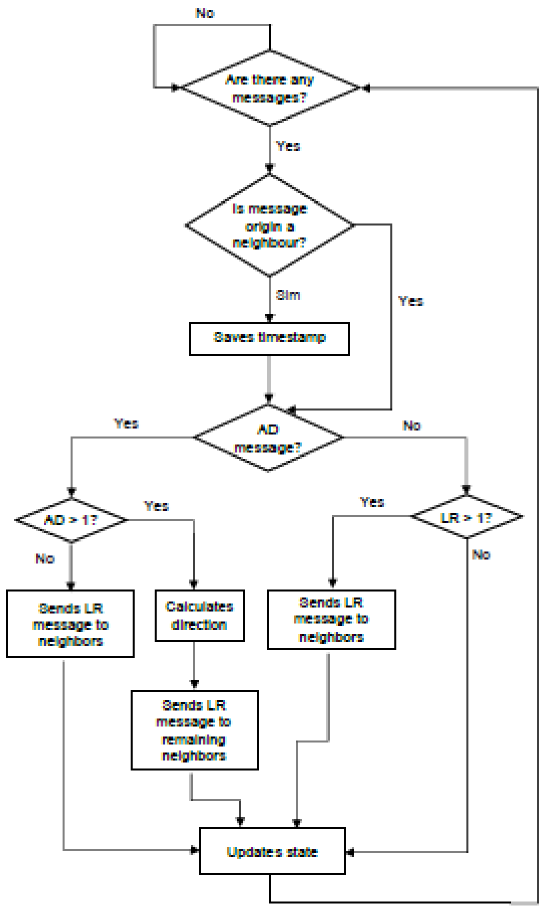

Figure 3 – Flowchart of the main loop of the algorithm.

In addition to this information, each node also has a list of messages, where messages waiting for processing are placed, a detection list, where the timestamp of the last detection for each neighbor is stored, and a timer. The timer starts with the timeout value and restarts when it reaches zero or the node changes state.

In Figures 3 and 4 are two flowcharts showing the operation of the algorithm. The flowchart of Figure 3 illustrates the main loop of the algorithm, waiting and processing messages. Figure 4 shows what happens when there is a motion detection which interrupts the main loop.

To better realize the operation of the algorithm let's assume that we have a user moving in an area with intelligent luminaires. Lighting radius and anticipation distance are equal to 2. When the user moves, the node that detects the movement changes its state to Detected and sends a LR message for all its neighbors. The fields origin and sender are the coordinates of the node, the timestamp and the timestamp of the origin are equal and the lighting radius is equal to 2.

The neighbors, when they treat LR message, change their status to Medium. Since it was a neighbor that detected the movement, the nodes save the message timestamp associated with the node coordinates, on the detections list. Each timestamp has a validity, after which is then eliminated of the detections list. Nodes send a LR message using the method described for sending messages, where the fields origin and timestamp of origin are equal to the fields of the previous message, the sender field is the node's coordinates, and the lighting radius is decremented, leaving the value 1. The other nodes that received the second message RI change their status to Minimum, because the lighting radius is equal to 1. Since it has been reached lighting radius limit, there is no need to send more messages.

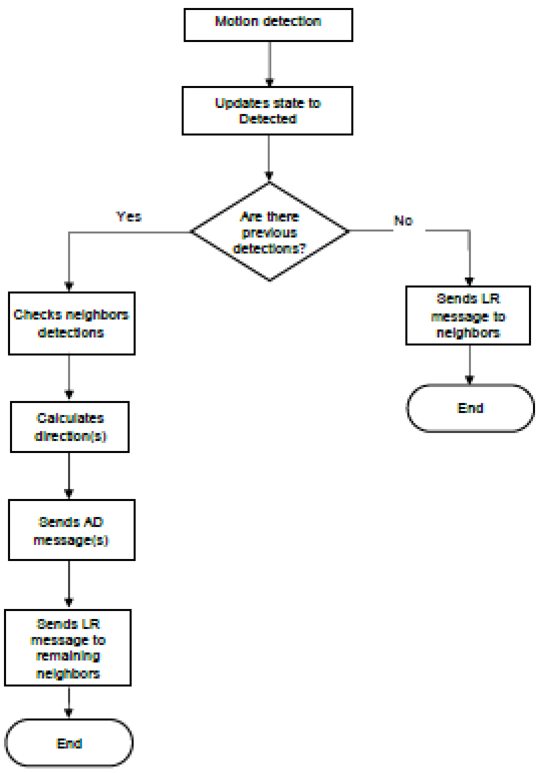

Figure 4 – Flowchart of when there is a detection.

In the second motion detection, the node that detected the movement calculates the direction of the user. After calculating the direction, sent one LR message and an AD message are sent. The AD message is sent to the neighbor node that is in the calculated direction. The origin and the sender of that message are the coordinates of the node that detected the movement, the timestamp and the timestamp of the origin are equal and the anticipation distance has the value 2. The LR message is completed similarly to the AD message and is sent to the remaining neighbors. The RI message is then propagated.

The node that receives the AD message changes its state to Anticipated and switches to high intensity. Since the anticipation distance is greater than 1, this node will also send a AD message and a LR message. In the AD message the fields of the origin and the timestamp of the origin are equal to the received message, the sender field are the coordinates of the node, the timestamp field is filled with the timestamp and the anticipation distance is decremented, leaving the value 1. The DA message maintains the direction, being sent to the corresponding node. The LR message is completed similarly to the message. It is then sent to the other neighbors except the sender of the message.

The node that received the AD message changes its state to Anticipated and switches to high intensity. As the value of the anticipation distance is equal to 1, this node only sends a LR message to all neighbors except the node that sent the message. The origin of the LR message and the origin timestamp will be the same as the AD message, the sender will be the coordinates of the node, the timestamp field is filled with the timestamp and the lighting radius will have the value 2. Since the second detection this process is repeated until the system no longer detects the user.

When there is more than one user, the system behaves as described above, while the users do not intersect. When users intersect each other the directions of those users are illuminated. This is due to the list of detections. When a node detects movement it will check the list of detections, on which neighbors detected movement recently and to those neighbors an AD message is sent, making it possible to illuminate all directions of users.

The nodes when processing messages have a policy of always remain at the highest possible intensity. We can consider the following order for the intensity: minimum> low> medium> high. This means that if a node receives a message that puts it on a lower intensity, the intensity will not change. If it receives a message that keeps it in the same intensity, the node maintains the intensity, but restarts the timer. If it receives a message that puts it in higher intensity, the node will change its intensity.

The presented algorithm has some limitations, particularly when multiple users are nearby. In case of multiple detections by different nodes in a short space of time, it may illuminate directions that the users are not following.

To the base algorithm described above, some improvements were made, having regard to the type of space for which this system was designed. The first improvement is the definition of a base intensity. Each node has a base intensity that is the lowest intensity that the node will have. The node can increase its intensity but never go below the base intensity. The goal is to allow areas that for various reasons, must always be illuminated. The second improvement refers to the special directions. Special directions allow a node when it detects motion in a given direction send the message to any other directions. This feature is used to create small paths, where it appears that users generally follow a different direction from that expected by the system.

C. Configuration Application

The configuration application allows to manage and monitor the system. However, as the system was developed using a simulator, the application goal became the creation and editing map scripts.

In Figure 5 we can see the home screen of the setup application. Of the various buttons present, the most important are the New Setup and Edit Setup. The New Setup lets create a new map script and the Edit Setup allows to choose a map script previously created and edit it. When the New Setup button is pressed, a new window opens where the map script can be created. If we click on the Edit Setup button a new window appears that lets you select a text file. Editing the map script is done with the rest of the home screen buttons.

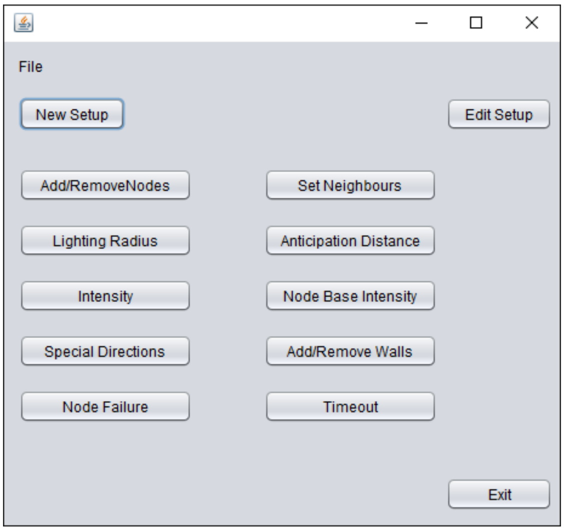

Figura 5 – Home screen of the configuration application.

The configuration application has the following features:

- Add and remove nodes;
- Set the neighbors of each node;
- Change the lighting radius;
- Change the anticipation distance;
- Set the power, in percentage, for each of the intensities;
- Add and remove walls;
- Set the base intensity for each node;
- Set special direction for each node;
- Set the timeout value;
- Simulate the failure of a node.

Each of these features opens a new window. These features are common to either create or edit a map script.

4. Evaluation

4.1 Evaluation Method

The algorithm will be evaluated through various tests. In each test the algorithm behavior is evaluated using a set of similar situations the system would face in the real world. The algorithm will be evaluated using two criteria:

- Energy savings;
- Average number of messages sent per node per time unit;

The main objective of this system is to save energy. It will be considered that the algorithm meets the purpose if the energy savings achieved during the tests are at least 30%. The calculation is made using a weighted average of all intensities. With the criterion of the average number of sent messages per node per unit time it is intended to evaluate the processing load subjected to each luminaire. This criterion is a determining factor in the choice of hardware components. The calculation is made using the average of messages sent divided by the running time of the simulation. This criterion only takes into account the nodes that sent at least one message.

Each test will consist of a map and multiple users. The map has an area of 20 by 20 lamps. In the center of the map there is an area with no light fixtures and walls around them, which is the input / output. Two of the lamps has the base intensity with high intensity and represent emergency exits. For each node the minimum intensity will be 25% of lamp power, low intensity will be 50%, the average intensity is 75% and the maximum intensity will be 100%. The timeout value is 2 units of time.

To evaluate the algorithm a total of six tests will be made. In each test the number of users and their movement pattern will be different. Users will have two movement patterns, the first, where users will go through similar paths and the second, where each user path will be different. For each of these movement patterns the number of users will vary, in order to simulate the period of lower use, normal use and increased use.

4.2 Testes

In total, six tests were made. The first test was done with two users who traveled similar paths. In the second test were four users to on similar paths and the third test were used seven users who traveled similar paths. In the fourth, fifth and sixth test were used two, four and seven users, respectively traveling different paths.

4.3 Result evaluation

In Table 2 we can see the energy saving statistics of all tests. The Total Energy field refers to the energy that would be spent if all the luminaires were at full intensity. The field Energy Spent is the energy spent by intelligent luminaires during the test. The Savings field is the energy savings obtained by intelligent luminaires. Observing the energy savings obtained in the tests we can conclude:

- When the space has little use, the movement pattern of users has little impact on the achieved savings (tests 1 and 4);
- The energy savings are higher when users run along similar paths. This difference is most obvious by comparing the results of test 3 and 6;
- As you increase the number of users present energy savings decreases;
- The system performed better than expected, with an average savings of around 60%, well above the expected 30%.

Test	Total Energy	Energy Spent	Saving
1	10875	3580	67,08%
2	10875	3880	64,33%
3	10875	4488	58,73%
4	10125	3390	66,52%
5	10500	4216	59,85%
6	10500	5047	51,93%
Média			61,42%

Table 2 – Energy saving statistiscs on the performed tests.

Table 3 presents data on sending messages. It can be observed the average of messages received by each node that received at least one message, the test duration in time units and the average message received by each node that received at least one message per time unit. With the data collected it can be seen that when users have similar paths the message average is higher due to a higher concentration of users in a smaller area. Each node processing about 2 messages per unit time is quite low. This means that the hardware for the luminaires can be simpler and therefore cheaper.

Test	Average	Duration	Average per unit of time
1	29,97	29	1,03
2	42,45	29	1,46
3	59,45	29	2,05
4	23	27	0,86
5	34,8	28	1,24
6	46,37	28	1,66

Table 2 – Statistics on sending messges on the performed tests.

5. Conclusion

With this project it was expected to develop an intelligent lighting system that works in a distributed manner. Due to some limitations, the system had to be tested using a simulator, developed specifically for this project. The main component of this system is the algorithm that is present in the fixtures. The approach meets largely the objectives of illuminating the area around the user and to determine its direction. However this approach has some limitations, in cases where there are multiple users close to each other.

In addition to developing the algorithm it was also provided the creation of an application that would allow to setup each luminaire during the installation and manage the system. Since the system was tested using a simulator, this application makes the initial configuration of the testing environment.

With the tests performed it was found that this system allows considerable energy savings. Moreover it was also found that the number of messages it sends each luminaire is relatively low, which can make the hardware of each luminaire cheaper.

For the future, the first thing to do would be to develop a prototype of the intelligent luminaires. After having some prototypes it will be possible to test in real conditions and analyze the system behavior. With the use of prototypes it would also be possible to create a web application that allows to manage the fixtures in real time. This application would have a set of features in addition to the features present in the application presented in this paper.

Something that can be introduced is the auto-configuration of fixtures. This means that it would not be necessary to set the initial parameters when installing the fixtures, simply connecting the lamp and the configuration would be done automatically.

6. References

[1] EIA projects world energy consumption will increase 56% by 2040, U.S. Energy Information Administration, July 2013 http://www.eia.gov/todayinenergy/detail.cfm?id=12251#

[2] EU energy in figures – Statistical Pocket Book 2015, page 11, European Union, 2015 http://ec.europa.eu/energy/sites/ener/files/documents/PocketBook_ENERGY_2015%20PDF%20final.pdf

[3] World Energy Outlook 2014, page 4, Internacional Energy Agency, 2014 http://www.iea.org/publications/freepublications/publication/WEO_2014_ES_English_WEB.pdf

[4] N. Yoshiura, Y. Fujii, N. Ohta, Smart street light system looking like usual street lights based on sensor networks, *13th International Symposium on Communications and Information Technologies (ISCIT), 2013*

[5] Tvilight, A Wireless Network and Sensor Technologies Company, http://www.tvilight.com

[6] Tvilight CityManager, Smart City Sensors, Network and Light Management Software, http://www.tvilight.com/management-software/#citymanager

[7] Tvilight CitySense, A Plug-and-Play Wireless Sensor, http://www.tvilight.com/citysense/

[8] Tvilight Case Study – Outdoor Parking http://www.tvilight.com/wp-content/uploads/2014/03/CSE_Hoogkerk_English_Online1.pdf

[9] R. Müllner, A. Riener, An energy efficient pedestrian aware Smart Street Lighting system, *International Journal of Pervasive Computing and Communications Vol. 7 No. 2, 2011*, pp. 147-161

[10] LiLong, ChuXiumin, WuYong, WuQing, The Development of Road Lighting Intelligent Control System Based on Wireless Network Control, *International Conference on Electronic Computer Technology, 2009*

[11] Z. Zhang, A. Mistry, W. Yin, P. L. Venetianer, Embedded Smart Sensor for Outdoor Parking Lot Lighting Control, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012*