Automatic Creation of Virtual Trips in Websites Using a Mobile Robot.
Jesús Savage, Humberto Espinosa, Edna Marquez, Emmanuel Hernández, Gabriel Vázquez
Laboratory of Intelligent Interfaces, University of Mexico, UNAM.

Abstract - In this paper we describe how to use a mobile robot to take pictures automatically to be used in image rendering applications. A potential fields algorithm is used to control the movements of a mobile robot with a video camera that it takes photographs of the environment, and with them panoramic pictures are created. Then, a navigable website is created by making a movie with the superposition of the panoramic pictures that follows the users' movements of the mouse.

Keywords: Mobile robots, Panoramic Pictures, Image rendering, Potentials fields

1 Introduction

In the last years, there has been an increase of the use of panoramic pictures to create virtual environments. This technique has the advantage of displaying real images instead of rendered ones. These pictures are created by applying and off-line process that stitch, blends, manipulates photographs and display the resulting image. The pictures are taken with expensive cameras and a special tripod. This method requires several hours of human work, and it makes the use of panoramic pictures expensive. We develop a method of doing this automatically. With this approach more complex environments can be created using several pictures in order that an user may travel in the environment in any direction instead of just rotating in one fixed place. This technique is part of a new computer graphic approach called image-rendering [4]. We use a mobile robot to take the photographs, allowing us to have a strict control over the position where they are taken.

2 Panoramic Pictures

A panoramic picture is an unusually wide picture that shows as much width-ways as the eye is capable of seeing, if not a greater left-to-right field of view than we can ever see (e.g. it shows 'behind you' as well as 'in front'). Cinema has exploited wide images, which fill more of the viewer's peripheral vision in order to enhance the immersive effect of films. Now panoramic images are being used for virtual reality and image-rendering application [6] such as the navigation on a website. Panoramic pictures are made by taken several pictures using a tripod, rotating it some angle between pictures and then joining them together.

The process of creating the panoramic is a stitching process. In this process an algorithm attempts to distort the pictures aligning them. To do this it is necessary make some optical analysis of the pictures. Another process that it is necessary to do is warping, this process depends of the lens used and the field of view that the image represents. The main problem that may cause that some pictures cannot stick together and completely ruin the panorma is the parallax. It can also cause ghosting or blurring. Parallax is a distortion in the picture which is the result form the moving of the nodal points of the camera while it is rotating to take the panoramic picture, this problem can be avoided mounting the camera in a way that it rotates over its nodal point.

Because we have control over the angle where the photograph was taken we can easily to automate the stitching process to generate all the panoramic pictures needed.

3 Mobile Robot Operation

There are several architectures to control mobile robots; we proposed the following that is based on the Aura architecture developed by Ronald Arkin [5]. In our approach the mobile robot's operation it is formed by several layers, see Figure 1, each one having a specific function that in whole control the behavior of the robot.

This architecture have two advantages, it can control a robot independently of its hardware design and all the task are self-governing of the others who give a great flexibility for the design, and implementation of a solution. Each of the layers will be described briefly in the following sections, given more details on the planner module.

3.1 Perception

One of the main objectives of the perception module is to obtain a symbolic representation of the data coming from the robot's internal and external sensors, from the Human/Robot interface, from the robot's tasks and if chosen from the simulator. The symbolic representation is generated after applying digital signal processing
3.2. Internal Sensors

Each of the sensors is represented by a data structure that has the following elements: sensor's name, sensor's type, sensor's position in the robot and a set of the sensor's values. We use a B14 robot, developed by Real World Interfaces, that has two wheel encoders and a battery level detector.

3.3. External Sensors

The robot has an array of 8 infrared sensors, 8 sonar sensors, 20 tactile sensors and a video camera.

3.4. Simulator

In the ViRBot system it is possible to simulate each of sensors' signals according to a physical model of the sensors. Thus, when the system is tested using the virtual robot, the simulator provides each of the sensors values. An user may include his own simulation algorithms [2].

3.5. Human Robot Interfaces

The communication between the robot and an user is done by several means: speech recognition, text provided by keyboard and control pads.

3.6. Robot's Tasks

The robot is programmed to perform certain tasks, like picking objects and delivering them to another place, during the day at a certain time.

3.7. World Model

The world model module uses the belief generated by the perception module and together with the information provided by the cartographer and the knowledge representation it generates a situation that needs to be solved.

3.8. Cartographer

For every room of the working environment there is a representation of how each of the rooms are interconnected between them, as well as each of the known obstacles included in them. The links between rooms are used later to form a tree that has in the root the origin room and one or several leafs have the destine room.

The know obstacles are defined as polygons that consists of a clockwise ordered list of its vertices. Representing the obstacles as polygons makes easier the search for a solution to navigate from one point to another. They are represented as a fact in an expert system as follows:

\[
\text{(polygon type room obstacle-name x0 y0...xn yn)}
\]

where type represents the type of obstacle, like a wall, a desk, etc. The xi and yi represents the coordinates of the obstacle. From this obstacle representation forbidden areas are created, which are areas not allowed for the robot to enter, they are built by growing the polygons by a distance greater than the radius of the robot, to consider it as a point and not as a dimensioned object. Figure 3 shows the forbidden areas of the experiment’s room. It is possible to create the configuration space in this way because our robot has cylindrical shape.

3.9. Knowledge Representation

An expert system is used to represent the knowledge that the robot has. In an expert system, the knowledge is represented by rules, each one contains the encoded knowledge of an expert, that is, the actions that the robot
would do if certain conditions are met.

The environment was defined as facts in an expert system, we used the expert system shell called CLIPS, developed by NASA [1], in which we add a TK graphics environment and sockets communication that allows to send data to it and to the robot through Internet.

3.10. Goal Activations

Given a situation recognition, a set of goals is activated in order to solve them.

3.11. Options

Set of hardwired procedures that solve, partially, specific problems.

3.12. Planner

The basic problem of planning the robot’s movements is reduced to: Given the initial position and heading of the robot $A$ in space $W$, a path $\tau$ specifying a continuous sequence of positions and headings of the robot $A$ must be found in order to avoid collisions with the obstacles $B_i$’s, beginning in the initial position and heading and finishing in the goal position and heading. If there is not such a path, the impossibility to solve the problem is reported.

Planning is defined as a procedure or guide for accomplishing an objective or task. This requires searching a space of configurations to find a path that corresponds to a set of the operations that will solve the problem. For example, in our system we may want the robot to move from one place to another and to take a picture. This problem will be solved by finding a sequence of operations that leads from an initial state to a goal state. Then the objective of planning is to find a sequence of physical operations to achieve the desired goal.

The robot’s planning organization consists of several hierarchical layers. The top layer is the planner, which takes as an input an initial and a final state and an environment description, and produces an overall plan that will take the robot from the initial to the final state. In the case that the command is to move the robot from one room to another the planner finds the best sequence of movements between rooms until it reaches the final destination. Inside each room it finds also the best movement path taking into account the known obstacles, that represent some of the objects in the room. Each of the objects in the environment has a database representation that includes a polygonal description shape of the object. Thus the planner uses this information to find the best path avoiding the polygons that interfere with the goal.

3.12.1. Local Plans Using Potential Fields

Under this idea the robot is considered as a particle that is under the influence of an artificial potential field $U$ whose local variations reflect the free space structure and it depends on the obstacles and the goal point that the robot needs to reach [3]. This approach has been widely used in partial known environments [7].

The potential field function is defined as the sum of an attraction field that pushes the robot to the goal and a repulsive field that takes it away from the obstacles. The movement planning is done by iterations, in which an artificial force is induced by

$$ F(q) = -\nabla U(q) $$

that forces the robot to move to the direction that the potential field decrees, where $\nabla$ is the gradient in $q$ and $q=(x,y)$ represent the coordinates of the robot position.

The potential field is generated by adding the attraction field $U_{atr}$ and the repulsive field $U_{rep}$

$$ U(q) = U_{atr}(q) + U_{rep}(q) $$

thus

$$ F(q) = F_{atr}(q) + F_{rep}(q) $$

$$ F_{atr}(q) = -\nabla U_{atr}(q) $$

$$ F_{rep}(q) = -\nabla U_{rep}(q) $$

3.12.2. Repulsive potential field

The goal of the repulsive potential field is to create a force that takes away the robot from the obstacles this is obtained using a potential value that tends to infinite in the surface of the obstacle and decreases as the robot goes away from it. The following equation shows a field with the previous characteristics.
\[ U_{\text{rep}} (q) = \frac{1}{2} \eta \frac{1}{\|q - q_{\text{obstacle}}\|} \]

where \( q_{\text{obstacle}} \) represents the coordinates of the obstacle. For several obstacles, the total field potential is the superposition of the individual potential field of each obstacle,

\[ U_{\text{rep}} (q) = \sum_{i=0}^{k} U_{i}^{k} (q) \]

We used these areas to find the repulsion forces that act on the robot. There are several approaches to find the repulsion force that a forbidden area acts on the robot, one of them is to find the centroid of the forbidden area and consider that all the mass is concentrated in that point.

### 3.12.3. Attractive potential field

Attractive field potential creates an attraction force through the goal configuration of the robot. It can be considered a parabolic field of the following form

\[ U_{\text{attr}} (q) = \frac{1}{2} \varepsilon_{i} \|q - q_{\text{destine}}\|^2 \]

where \( q_{\text{destine}} \) represents the coordinates of the destine.

There are several methods for planning using potential field, one of them is to use the gradient vector to guide the robot from the initial position to the goal. Defining the unitary vector pointing to the gradient direction

\[ \hat{f}(q) = \frac{\mathbf{F}(q)}{\|\mathbf{F}(q)\|} \]

in this way the movement in discrete times is defined by

\[ q_{i+1} = q_{i} + \delta_{i} \hat{f}(q) \]

As we can see in Figure 2 the growing structure represents an obstacle that generates a repulsive field, and the goal generates an attractive field in parabolic form close to it and linear far away that it is represented by an inclination in all the generated surface.

![Figure 2. Potential Field created by an obstacle and the destine.](image)

### 3.12.4. Local minimum avoidance

The major cause of inefficiency with a potential field planner system is the local minimum avoidance. A local minimum appears when the repulsive force cancels the attractive force creating a zero-force configuration of the total potential function. The solution used in the system described here consist in creating temporarily attractive forces near the corners of the obstacle that is creating the local minimum, this forces help the planner to leave the local minimum.

### 3.13. Navigator

The navigator takes the overall plan and it finds a set of trajectories that basically consists of distances and angles that the robot needs to execute to reach its destination. Given a set \((x_1, y_1, x_2, y_2...x_n, y_n)\) of points obtained by the planner and a time \(t\) to visit them the planner finds the angle of rotation \(\theta\) a distance \(d_i\) and a speed \(v_i\) to reach each of them.

### 3.14. Pilot

The pilot takes the trajectories generated by the navigator and executes them. Basically on each step \(i\) it has to move the robot a distance \(D_i\) and turn \(\theta_i\) degrees. It checks for unknown obstacles by the planner and tries to avoid them using behavior control.

### 3.15. Controller

The controller controls the Robot's motors and reads data coming for the motors and sensors.

We use a mobile robot model B14 fabricated by Real World Interface, Inc. It is a cylindrical mobile robot equipped with a wheeled base that allows the robot to
move in two axes: translation (movement parallel to the robot head alignment) and rotation (movement perpendicular to the robot head alignment). The robot have motion controllers that control the movements of the robot, it also have odometers to keep track of the position of the robot.

3.16. Learning

The system can learn to solve new problems by using genetic algorithms, probabilistic methods, Markov chains, etc.

4 Automation for the creation of panoramic pictures

One of the main problems for the automation of the creation of panoramic pictures is the control of all the parameters involved in the stitching and warping process. Using the internal robot encoders to control all this parameter can easily create an automatic process to make the panoramic pictures. This gives the opportunity to create image-rendering virtual environments in wide areas with a minimum of human work.

A set of points where the panoramic pictures will be took are selected accordingly by the user to create the virtual trip, as is shown in Figure 3.

Then the robot needs to visit these points and take the pictures. The planner finds the robot's movements by using the potential fields algorithm to control the motion of the robot over the whole area using the symbolic representation of the working area. The robot moves from one point to another, as is shown in Figure 4, using the potential field algorithm, stopping and taking pictures by rotating its base until it completes a 360-degree turn.

After the pictures are taken commercial software is used to create all the panoramic pictures needed. These panoramic pictures are used to create the virtual visit.

Figure 5 show an image created by the image-rendering engine, Figure 6 shows two pictures used to form this image, the pictures were took with the robot 30 degrees apart.

Figure 7 shows a panoramic view created using this technique in one of the defined points. Using all the panoramic views virtual trips in websites can be made.
5 Conclusions

Using a mobile robot is easy to acquire photographs for its posterior use in image-rendering techniques. Here is described a method to make virtual trips in Websites using panoramic pictures. Using a mobile robot is an easy and fast way to take the image to create and model a real-world environments. The robot’s encoders helps to have more control of where each picture is took. This data is very important for applying a suitable image-rendering method. Also, construct a robot architecture made of modules helps the programming of new tasks because only small changes must be done in order to have completely new behaviors.

Figure 6. Pictures tool by the robot.

6 References


Figure 7. Panoramic view of one of the defined points in the room created by the mobile robot