Chapter 8: Conclusions

The actual result of this final term project is a robot control program, which enables the robot to meet the expectations set up at the beginning of the project. These functionalities are summarised and discussed on the basis of the user interface of this control program as presented on the next figure:

![Figure 63: Snapshot of the robot control program](image)

On the top left side of the interface, the user can choose for one of the three operation modes for the robot:

- **Automatic Operation**: The robot walks towards the target guided by the camera, but with this camera target tracking process integrated in the serial SMPA structure, so no parallel processing is performed.
- **Continuous Operation**: Generally the same as Automatic Operation, with this exception that the camera never stops tracking the target when this target has been reached after a run. When the target object is moved later, the camera will detect this and order the robot to start walking towards this new target position again.
• Blackboard Operation: In this working mode, the camera target tracking procedure runs as a parallel process, implementing as such a blackboard control architecture.

In the middle part of the interface, a clear distinction has been made between the camera control to the left and the robot control to the right. A number of different control buttons enable the user to control the robot and camera directly, to set up parameters or to access basic functions. In order to enhance the user-friendliness of the program and to enable a possible remote surveillance, the camera video image and the potential field shadow map as calculated by the robot, are shown to the user. In the lower right corner, the sixteen measurements of the four abstract sensors are made visible and more to the right, one can analyse how these readings have been fused to come to unambiguous data about target and obstacle. Even more towards the lower right corner, the program shows which behaviour and control action were deduced from these sensor readings. Finally, one can observe the robot position and orientation in the lower right corner.

Using this control program, the robot is able to navigate itself towards a certain target in a complex and a priori unknown environment, as was asked. The robot is furthermore capable of dealing with moving targets as long as the camera is able to follow this target object, which is limited by the speed of the camera target tracking algorithm and the controlling computer, and the maximum camera pan angle. On the other hand, the robot control program is not designed to cope with non-static environments, or more specifically with moving obstacles. The reason is that to deal with the problem of the extremely limited view of the robot sensors, the high potential of any detected obstacle is artificially kept into this high state for the rest of the operation time, unlike what is done for the target. This implicates that obstacles are never removed from the map and are therefore considered immobile. This approach is inevitable, however, because otherwise the robot would in general never have the notion of only one obstacle at a time, which would be fatal for a decent path planning.

Though “logic behaviour” has been a key aspect pursued throughout the implementation of the different robot control program modules, it must be admitted that this robot behaviour is sometimes not that logic, due to the combination of the potential field navigation method with the always very incomplete knowledge of the environment. However, with the given sensory equipment, this cannot be averted, unless the camera should also be used for retrieving other environmental information than the target object location. The potential field technique shows furthermore its robustness as tests showed that the target object could be put out of view for a considerable part of the operation time, without keeping the robot from reaching this target in the end.

Another imperfection one could point out is the slow movement speed of the robot, as it needs about ten seconds of “thinking time” between two consecutive movements. Moreover, this figure of ten seconds is only valid for the Automatic or Continuous operation modes, but not for the Blackboard
Chapter 8: Conclusions

approach. The execution speed in Blackboard mode depends greatly on which other processes are running on the computer, but exceeds these ten seconds in every case. However, as opposed to the logic behaviour, speed was not a key consideration throughout the implementation of the robot control program, as the pneumatic robot was already slow by itself and it was never the idea to make it a racing robot. Nevertheless, a lot of efforts have been made during the implementation process to accelerate all the different routines used, the most dramatic result being the reduction of the calculation time for the potential field. This computing time is still considerable however, but as already discussed in the map-building chapter, this delay would be hardly noticeable on a more modern computer.

In spite of all its limitations, the robot succeeds in performing the task that was set up at the beginning of the project: to walk towards a (moving) target in a complete unknown and complex environment with obstacles. This was not a simple demand in view of the limited sensory equipment of the robot and the fact that this pneumatic robot was not actually built to perform such a task; note that “normal” autonomous mobile robots generally have 24 or more ultrasonic sensors at their disposal and can be manoeuvred very precisely. Moreover, reusability of all the different programming components has been ensured, because it happens too often that interesting projects are disassembled and their results never used again due to a lack of portability.