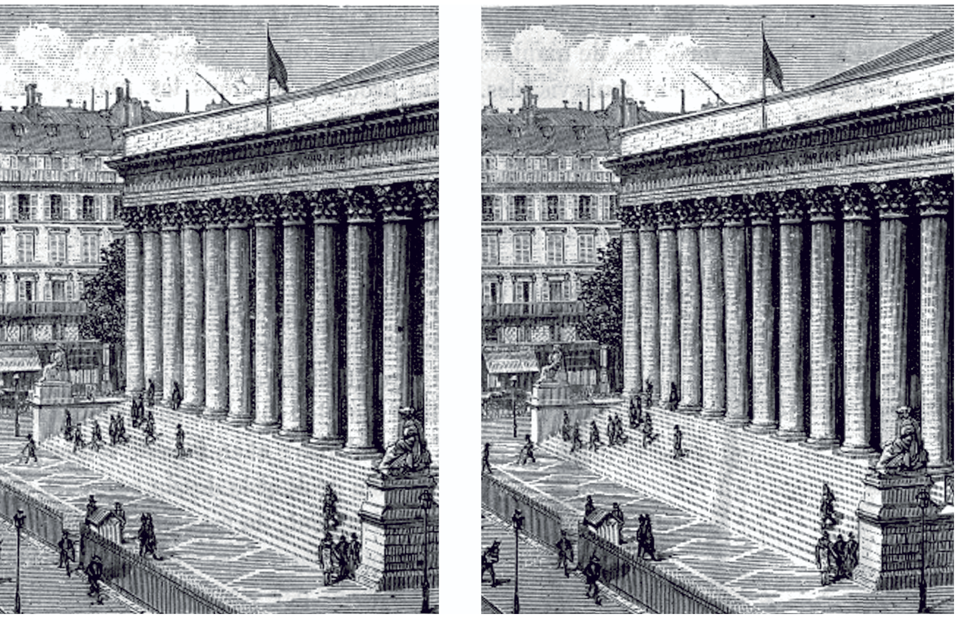
Part V Robotics, Vision and Control

Part

V Robotics, Vision and Control



It is common to talk about a robot moving to an object, but in reality the robot is only moving to a pose at which it *expects* the object to be. This is a subtle but deep distinction. A consequence of this is that the robot will fail to grasp the object if it is not at the expected pose. It will also fail if imperfections in the robot mechanism or controller result in the end-effector not actually achieving the end-effector pose that was specified. In order for this approach to work successfully we need to solve two quite difficult problems: determining the pose of the object and ensuring the robot achieves that pose.

The first problem, determining the pose of an object, is typically avoided in manufacturing applications by ensuring that the object is always precisely placed. This requires mechanical jigs and fixtures which are expensive, and have to be built and setup for every different part the robot needs to interact with – somewhat negating the flexibility of robotic automation.

The second problem, ensuring the robot can achieve a desired pose, is also far from straightforward. As we discussed in Chap. 7 a robot end-effector is moved to a pose by computing the required joint angles. This assumes that the kinematic model is accurate, which in turn necessitates high precision in the robot's manufacture: link lengths must be precise and axes must be exactly parallel or orthogonal. Further the links must be stiff so they do not to deform under dynamic loading or gravity. It also assumes that the robot has accurate joint sensors and high-performance joint controllers that eliminate steady state errors due to friction or gravity loading. The non-linear controllers we discussed in Sect. 9.4.3 are capable of this high performance but they require an accurate dynamic model that includes the mass, centre of gravity and inertia for every link, as well as the payload.

None of these problems are insurmountable but this approach has led us along a path toward high complexity. The result is a heavy and stiff robot that in turn needs powerful actuators to move it, as well as high quality sensors and a sophisticated controller – all this contributes to a high overall cost. However we should, whenever possible, avoid solving hard problems if we do not have to. Stepping back for a moment and looking at this problem it is clear that

the root cause of the problem is that the robot cannot see what it is doing.

Consider that the robot could see the object and its end-effector, and could use that information to guide the end-effector toward the object. This is what humans call *hand-eye coordination* and what we will call vision-based control or visual servo control – the use of information from one or more cameras to guide a robot in order to achieve a task.

The pose of the target does not need to be known apriori, the robot moves toward the observed target wherever it might be in the workspace. There are numerous advantages of this approach: part position tolerance can be relaxed, the ability to deal with parts that are moving comes almost for free, and any errors in the robot's open-loop accuracy will be compensated.

A vision-based control system involves continuous measurement of the target and the robot using vision to create a feedback signal and moves the robot arm until the visually observed error between the robot and the target is zero. Vision-based control is quite different to taking an image, determining where the target is and then reaching for it. The advantage of continuous measurement and feedback is that it provides great robustness with respect to any errors in the system. There are of course some practical complexities. If the camera is on the end of the robot it might interfere with the task, or when the robot is close to the target the camera might be unable to focus, or the target might be obscured by the gripper.

In this part of the book we bring together much that we have learnt previously: robot kinematics and dynamics for arms and mobile robots; geometric aspects of image formation; and feature extraction. The part comprises two chapters. Chapter 15 discusses the two classical approaches to visual servoing which are known as *position-based* and *image-based* visual servoing. The image coordinates of world features are used to move the robot toward a desired pose relative to the observed object. The first approach requires explicit estimation of object pose from image features, but because it is performed in a closed-loop fashion any errors in pose estimation are compensated for. The second approach involves no pose estimation and use image plane information directly. Both approaches are discussed in the context of a perspective camera which is free to move in $SE(3)$ and their respective advantages and disadvantages are described. The chapter also includes a discussion of the problem of determining object depth, and the use of line and ellipse image features.

Chapter 16 extends the discussion to hybrid visual-servo algorithms which overcome the limitations of the position- and image-based visual servoing by using the best features of both. The discussion is then extended to non-perspective cameras such as fisheye lenses and catadioptric optics and arm robots, holonomic and non-holonomic ground robots, and a flying robot.

This part of the book is pitched at a higher level than earlier parts. It assumes a good level of familiarity with the rest of the book, and the increasingly complex examples are *sketched out* rather than described in detail. The text introduces the essential mathematical and algorithmic principles of each technique, but the full details are to be found in the source code of the MATLAB® classes that implement the controllers, or in the details of the Simulink® diagrams. The results are also increasingly hard to depict in a book and are best understood by running the supporting MATLAB® or Simulink® code and plotting the results or watching the animations.