Robotics and Autonomous Systems 62 (2014) 330–339

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Bootstrapping a robot's kinematic model

Alan Broun [a,*], Chris Beck [b], Tony Pipe [a], Majid Mirmehdi [b], Chris Melhuish [a]

[a] Bristol Robotics Laboratory, Bristol, UK

[b] Visual Information Laboratory, University of Bristol, Bristol, UK

HIGHLIGHTS

- We present a system that is able to autonomously build a 3D model of a robot's hand.
- A hand is located and moved to the centre of the robot's field of view using exploratory motions.
- The system and the built models are validated by a number of experiments.

ARTICLE INFO

Article history:
Available online 23 October 2013

Keywords:
Kinematic identification
Model building
Body schema

ABSTRACT

We present a system that is able to autonomously build a 3D model of a robot's hand, along with a kinematic model of the robot's arm, beginning with very little information. The system starts by using exploratory motions to locate and centre the robot's hand in the middle of its field of view, and then progressively builds the 3D and kinematic models. The system is flexible, and easy to integrate with different robots, because the model building process does not require any fiducial markers to be attached to the robot's hand. To validate the models built by the system we perform a number of experiments. The results of the experiments demonstrate that the hand model built by the system can be tracked with a precision in the order of 1 mm, and that the kinematic model is accurate enough to reliably position the hand of the robot in camera space.

1. Introduction

When building a robot, or larger robotic system, it is common for designers to explicitly give the robot all of the models it needs to control itself (in the form of kinematic and dynamics models) and all of the models it needs to interact with objects in the world around it (in the form of 3D CAD models). This is a practical approach, but it lacks flexibility in cases where properties of the real world may deviate from the model during the lifetime of the system. For example, an arm may degrade or be replaced with an arm of different dimensions, or novel objects may be encountered by the robot, meaning that existing models may be found wanting. In these cases the models must be extended or replaced. The task of taking the measurements for models may also be complicated by the robot being in hard to reach, or hazardous locations.

Our work focuses on building robots and robotic systems which can autonomously construct models of themselves and the external objects with which they interact. In particular, we explore the use of active 3D vision as a tool that a robot can use to explore itself, and its surroundings, in order to autonomously construct models.

The increasing availability of reasonably priced depth cameras such as the Mesa Imaging SwissRanger or the Microsoft Kinect has made it easier for robotic systems to perceive the world in 3D. These cameras provide depth values for pixels in the image, and so produce a 3D point cloud in camera space. The quality of the point clouds produced by these cameras reduces the need for researchers to set up technically challenging stereo camera systems, which often rely on the presence of highly textured areas in order to achieve reasonably similar results.

A practical problem for a robot with 3D vision is the task of relating the movement of its body to the Cartesian space of its camera system, so that it can interact with objects it sees. More fundamental than that, it may also be a problem for the robot to work out which parts of a 3D image belong to its body and to its hand.

We present a solution to both of these problems in the form of an extended version of the system we presented in [1]. The system allows a robot to reliably identify its hand in its field of view, and then to build a kinematic model of its arm in camera space. Building the kinematic model in camera space implicitly determines the transformation between camera space and the

* Corresponding author. Tel.: +44 7815773017.
E-mail addresses: abroun@alanbroun.net, alan.broun@brl.ac.uk (A. Broun), csxcb@compsci.bristol.ac.uk (C. Beck), tony.pipe@brl.ac.uk (T. Pipe), majid@compsci.bristol.ac.uk (M. Mirmehdi), chris.melhuish@brl.ac.uk (C. Melhuish).

0921-8890/$ – see front matter
http://dx.doi.org/10.1016/j.robot.2013.09.011

model. Once the kinematic model is built, inverse kinematics can be used to accurately move the manipulator of the robot in camera space. In effect, the robot is therefore able to 'bootstrap' itself, from a state of fairly limited knowledge, to having a kinematic model of its arm, which it can then use to further interact with, and to explore, the world.

The system works by first using a series of exploratory motions to roughly identify the location, and extents of the robot's hand in its field of view. It then uses a simple form of visual servoing to move the hand to the centre of the field of view, in order to maximise the quality of the subsequent processes. Once centred, the system builds a model of the robot's hand by turning the hand in front of a Kinect depth camera, whilst aligning and merging the point clouds obtained from the Kinect into a common reference frame. The model is then used to track future movements of the hand, by aligning the model against incoming point clouds. This system has the useful property of not just providing an estimate for the transformation of the hand in camera space, but also providing a 3D model of the hand which can be useful for other tasks such as planning grasps, or checking for collisions between the robot and the environment.

Once a model of the robot's hand has been built and is being tracked, we then use it to automatically build a kinematic model of the robot's arm by tracking the movement of the hand as each revolute joint in the arm is rotated in turn. This allows us to build an accurate model of the arm, starting with very little information. This is an advantage, as a robot that can deduce information for itself, is potentially more robust, and requires less work to commission.

The rest of the paper proceeds as follows. Section 2 describes related work and reviews the techniques which we use to build our system. Section 3 describes the robotic platform we use for our experiments. Section 4 provides a description of how the robotic hand is modelled and tracked along with details of automatically building a kinematic skeleton for the robotic system. Section 5 evaluates the accuracy of the system, and Section 6 presents conclusions along with ideas for future work.

2. Background and related work

2.1. Exploratory motion and active vision

The work of Ballard [2] was amongst the first to look in depth at camera systems which were not simply passive. Ballard observed that more information may be obtained from a visual scene, or obtained at a lower computational cost, through the process of moving the camera system and observing the scene from a number of different viewpoints. Such systems are often termed *active* vision systems to distinguish them from passive vision systems.

An alternative to moving the camera in a system is to move the object or scene being observed. The idea of using exploratory motions to both identify a robot's end effector, and also to segment objects of interest from the background was explored in work by Marjanovic et al. [3] and later Fitzpatrick and Metta [4] at MIT as part of the work on the Cog robot. The technique was explored in detail by Broun and Studley [5], who showed that a waving exploratory motion could be reliably detected, even in the presence of a large amount of distracting motion. Work has also been done by Katz and Brock [6] on using exploratory motions to autonomously identify the structure of articulated objects.

2.2. Object modelling and tracking

When building models from range data, such as that obtained from a laser scanner or depth camera, the Iterative Closest Point (ICP) algorithm presented by Chen and Medioni [7] and Besl and McKay [8] is a widely used algorithm for aligning one depth camera frame with either another frame, or with a reference model.

The ICP algorithm has been the subject of much research since its initial presentation. Rusinkiewicz and Levoy [9] identified the key stages that make up the ICP algorithm, outlining a number of techniques for making the algorithm more efficient and speeding up convergence. The ICP algorithm was used as a key part of an object modelling and tracking system built by Weise et al. [10], and a very similar system was used to build models of objects held in a robot's hand by Krainin et al. [11]. In both of these systems, models were constructed by first aligning point clouds from a depth camera into a common coordinate frame, using the ICP algorithm. Subsequently, corresponding points from the point clouds were averaged together to form surfel (surface element) models. Surfels as described in [12] are orientated 3D points, which can be used to describe complex geometric objects without explicit connectivity information. The advantage of averaging point clouds together to form a surfel model is that it smooths out a lot of noise that would otherwise accumulate as a result of estimating lots of small transformations [13].

Tracking a robot's hand in camera space is a special case of tracking an arbitrary 3D object in camera space, and this is an active area of research, with Lepetit and Fua [14] providing a comprehensive survey of the main techniques. Fiducial tracking [15] involves tracking markers attached to the object of interest. Model-based tracking involves posing a 3D model of the object of interest to best match the information coming from the camera. This method has been used extensively in human hand tracking applications, such as [16].

2.3. Kinematic identification

There are a number of methods available for identifying a robot's kinematic model. Early work from the 1980s includes Circle Point Analysis (CPA) used by Stone et al. [17], and described in detail by Mooring et al. [18]. CPA involves fitting circles to observed endpoint locations in order to identify the axis of revolution for a revolute joint. Another method for identifying the joint axes of a robot is the Jacobian Matrix Method of Bennett and Hollerbach [19]. This method requires either joint torque sensors or a method of estimating the linear and angular velocity of the robot's end link [19].

More recently, the field of developmental robotics has taken an interest in kinematic identification. Here, it has been explored as part of more general efforts to enable robots to build and maintain a *body schema* for themselves. In the context of robotics, Hoffmann et al. [20] describe a body schema as a group of body representations, which allow an embodied agent to control its actions, and to integrate sensory information such as vision or touch into common frames of reference. These representations may include kinematic and dynamic models, and the emphasis is on building the models autonomously. The aim is to give a robot the ability to adapt to changes in its body due to damage, or to dynamically extend its body schema to allow the use of tools. Hersch et al. [21] built a kinematic model of a robot by observing end effectors using an iterative gradient descent approach. Sturm et al. [15] presented a system that uses a Bayesian network to learn arbitrary kinematic chains, which can also cope with changes in the kinematic chains as the system runs. This system however, requires observations of all joint positions to build the kinematic chain, whereas our system only needs to observe the movement of the end of the chain.

Finally, recent work by Hart and Scassellati [22] takes a similar approach to the one presented here. The difference lies in the fact that the method of Hart and Scassellati requires an Augmented Reality (AR) marker to track the hand, whereas our method builds and tracks a complete model of the robot's hand, and so can operate without AR markers.