Towards a Stable Robotic Object Manipulation Through 2D-3D Features Tracking

Regular Paper

Sorin M. Grigorescu1,* and Claudiu Pozna1,2

1 Department of Automation, Transylvania University of Braşov, Braşov, Romania
2 Informatics Department, Széchenyi István University, Gyor, Hungary
* Corresponding author E-mail: s.grigorescu@unitbv.ro

Received 16 Apr 2012; Accepted 24 Jan 2013

DOI: 10.5772/55952

© 2013 Grigorescu and Pozna; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract In this paper, a new object tracking system is proposed to improve the object manipulation capabilities of service robots. The goal is to continuously track the state of the visualized environment in order to send visual information in real time to the path planning and decision modules of the robot; that is, to adapt the movement of the robotic system according to the state variations appearing in the imaged scene. The tracking approach is based on a probabilistic collaborative tracking framework developed around a 2D patch-based tracking system and a 2D-3D point features tracker. The real-time visual information is composed of RGB-D data streams acquired from state-of-the-art structured light sensors. For performance evaluation, the accuracy of the developed tracker is compared to a traditional marker-based tracking system which delivers 3D information with respect to the position of the marker.

Keywords Object Tracking, Robot Vision, Mobile Manipulation, Service Robotics

1. Introduction

One of the main tasks of service robotics systems operating in human environments, such as the FRIEND (Functional Robot with dexterous arm and user-frIENdly interface for Disabled people) [1] and PR2 (Personal Robot 2) [2] platforms displayed in Figure 1, is to reliably handle common household objects, such as plates, bottles or boxes, usually placed in heavy cluttered scenes. The common approach in many mobile manipulation scenarios is to recognize and calculate the pose (position and orientation) of the objects of interest [3], followed by further manipulative actions [4]. Throughout the manipulation procedure, there is no visual information available with respect to the imaged scene, that is, of the pose of the objects of interest and of the obstacles. Such a high uncertainty increases the risk of robotic handling failures if the state of the environment changes, namely if the poses of the objects, as well as of different obstacles present in the scene, varies.
Figure 1. Mobile manipulation platforms: (a) FRIEND, (b) PR2.

The main goal of tracking rigid bodies in robotics is to determine their pose with respect to a common reference coordinate system $W$, such that, if the state of the environment changes, the manipulation path of the robotic system is changed accordingly. This challenge is illustrated in Figure 2, where different tracking snapshots of an object of interest are illustrated. The object is tracked both at the 2D image level (see Figure 2[a-d]), as well as in the 3D point cloud data (Figure 2[e-h]) delivered by an RGB-D (Red, Green, Blue, Depth) MS Kinect® camera. The main advantage of such sensors is the calculated depth map, or point cloud, structure, which, along with the RGB image, describes the colour and real-world distances between the sensor and the visualized surfaces. Recently, they have become an intensively used perception sensor in a large number of indoor robotic applications.

In this work, we tackle the challenge of a single rigid object of interest pose $O$ tracking with respect to $W$. The approach is based on a collaborative tracking framework which determines the pose of the object in the 2D image, while simultaneously tracking 3D point cloud features on the object of interest.

The contributions of the paper may be summarized as follows:

- development of a collaborative robotic tracking framework for fusing information from more trackers;
- implementation of a 2D-patch-based particle filtering algorithm for tracking a reference object of interest;
- occlusion detector based on ray casting using an RGB-D camera system;
- usage of object tracking for improving mobile manipulation scenarios.

As will be shown in the performance evaluation section, it is relatively difficult to compare the results of the proposed tracker, which by its nature processes 3D information, to existing 2D systems. Although there are a large number of image-based trackers which accurately deliver visual information in the 2D image domain, the tracker presented in this paper is intended to be used in service robotics application where 3D information regarding the objects of interest is imperative.

Figure 2. Tracking during object grasping and manipulation. Snapshots of 2D features (a-d) and 3D point cloud (e-f) tracking.
1.1 Related Work

Although object tracking is a well-developed research area in the computer vision community, its application both in mobile manipulation and robotics in general [5] is quite restricted. Recently, Krainin et al. [6] applied the concept of object tracking during manipulation for building online 3D models of objects of interest. As they remark, the online learning and tracking of new objects is an imperative task for successful robotic manipulation scenarios. Also, tracking approaches have been proposed by Wang et al. [7] and Krainin et al. [6] for the purpose of hand tracking and modelling.

Teichman and Thrun [8] proposed a semi-supervised, boosting classification approach to the problem of track classification in dense 3D range data. The method uses a series of 2D-3D features, such as Spin images, Histogram of Oriented Gradients (HOG) and the object’s oriented bounding box size.

A novel paradigm for training a binary classifier in the context of tracking has been proposed in [9, 10, 11] for the problem of pure 2D image Region of Interest (ROI) tracking. The learning process is guided by positive (P) and negative (N) constraints which restrict the labelling of the unlabelled set. P-N learning evaluates the classifier on the unlabelled data, identifies examples that have been classified in contradiction with structural constraints and augments the training set with the corrected samples in an iterative process. Another online boosting tracking technique has been proposed by Grabner and Bischof [12] and Stalder et al. [13]. These authors propose an on-line AdaBoost feature selection method. Contrasting them with traditional features such as Haar wavelets [14], Wang et al. [17] proposed the usage of so-called superpixels for the purpose of object tracking.

Boosting is a machine learning technique used in a variety of computer vision applications such as image segmentation, text and object recognition, natural language processing, medical diagnostics, etc. In this paper, a boosting approach similar to the one described by Grabner and Bischof [12] has been used to track the object of interest in the 2D image domain.

One of the first applications of boosting in the area of computer vision was conducted by Viola and Jones [14] for the purpose of fast object detection. In this research, boosting, in its AdaBoost variant [19], was used for the purpose of feature selection. Traditionally, the classifier training approach was performed off-line using a large amount of training samples [14]. In this work, the focus is on object tracking without using off-line training. Hence, the training examples are composed of the first streams of images from the image acquisition system. The tracked features are composed of Haar-like patches.

In a number of recent papers, such as that by Choi et al. [15], the output results from more trackers are fused together using a weighting scheme to improve the performance of the overall tracking procedure. The same concept has been successfully applied by Yang et al. [16] in relation to the problem of tracking the motion of the human heart. In Yang et al. [16] a collaborative tracking framework is suggested.

Tracking has also been heavy investigated in relation to camera pose and the dense 3D reconstruction of human environments. The DTAM (Dense Tracking and Mapping) system proposed by Newcombe et al. [17] relies not on feature extraction, but on dense every-pixel processing. As a single hand-held RGB-D camera passes over a static scene, detailed textured depth maps at selected keyframes are estimated in order to produce a surface patchwork with millions of vertices.

2. A collaborative tracking framework

The collaborative nature of the tracking framework is defined within the classical Bayesian tracking approach. Let $x_t$ be the state, that is, the true 6-DoF (Degrees of Freedom) pose of the tracked rigid body, or object, at time instance $t$:

$$x_t = [x_t \ y_t \ z_t \ \varphi_t \ \psi_t \ \theta_t],$$  \hspace{1cm} (1)

where $x_t$, $y_t$ and $z_t$ give the 3D position along the three Cartesian axes, while $\varphi_t$, $\psi_t$ and $\theta_t$ represent the orientation.

The tracking challenge can be defined as the estimation of the posterior probability $p(x_t \mid z_{1:t})$, where $z_{1:t}$ are the past t poses, or state measurements:

$$z_{1:t} = \{ z_{1:t}, z_{2:t}, ..., z_{t} \}. \hspace{1cm} (2)$$

Sequential Bayesian tracking follows a Markov modelling approach, where the prediction step is defined as:

$$p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}) \cdot p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \hspace{1cm} (3)$$

The update state is defined as:

$$p(x_t \mid z_{1:t}) = p(z_t \mid x_t) \cdot p(x_t \mid z_{1:t-1}) \hspace{1cm} (4)$$

In order to improve the tracking capability, we propose a set of collaborative trackers that can take advantage of each other’s data. The tracking framework is built around a set of two trackers:

- a 2D patch-base tracker $p(r_1)$ which tracks a reference patch in the RGB-D input stream;
- a 2D point-base tracker $p(r_2)$ that aims to track 2D point features in the 2D image domain, as well as determining direct 2D-to-3D correspondences for estimating the 3D pose of the tracked object.
Both trackers contribute to the final state estimate as:

$$p(x_t | x_{t-1}) = p(x_t | z_t, p(r_1)) + p(x_t | z_t, p(r_2))$$

(5)

As suggested by Yang et al. [16], the information from both trackers may be fused as:

$$x_t = \arg \max_{x_t} p(x_t | z_{1:t})$$

(6)

### 3. 2D-3D Object Tracking

The block diagram of the proposed tracking loop for improving the mobile manipulation capabilities of service robots is shown in Figure 3. The first step in the diagram is to acquire a stream of RGB-D data, that is, data of images with corresponding depth information provided by structured light sensors (e.g., MS Kinect®, Asus®) or stereo cameras (e.g., Point Grey’s Bumblebee®).

The reference features are calculated from an object recognition module. In this paper, a clustering object detection system is used, which segments objects of interest on flat surfaces such as tables [18]. The determined 3D object cluster is further projected on the 2D image, and an initial tracking patch is calculated in the form of an ROI. As will be explained, the initial patch is tracked using an on-line boosting method. Parallel with the patch-based tracker, an optical flow system determines the 2D-3D correspondences between consecutive frames. Finally, an occlusion detector evaluates possible object occlusions.

#### 3.1 2D Patch-Based Tracking

A patch is considered an ROI in the 2D image that is being tracked by a specific tracker. This patch provides a search area for the second tracker, which establishes the 2D-3D point correspondences and estimates the transformation between consecutive poses of the object of interest. For tracking the initial patch, an on-line boosting learning technique has been implemented, as will be further explained below.

The boosting approach is a general framework used to improve the accuracy of a certain machine learning algorithm. This is performed by combining a weighted voting scheme using N hypotheses which have been generated by a repeated training built around different subsets of training data. A boosting classifier is composed of so-called weak and strong learners, or classifiers:

- **Weak learner**: has to perform only slightly better than random guessing, that is, its overall classification error has to be slightly smaller than 50%. The hypothesis \( h_{\text{weak}} \) is obtained from a learning algorithm.
- **Strong learner**: from a set of \( N \) weak learners, a strong learner, or classifier, is obtained as a linear combination of weak learners:

$$h_{\text{strong}}(x) = \text{sign}(\text{conf}(x)),$$

(7)

where \( x \) is a data sample.

The \( \text{conf}(\cdot) \) operator is a confidence measure that provides the classification precision of the boosting learner:

$$\text{conf}(x) = \sum_{n=1}^{N} \alpha_n \cdot h_{\text{weak}}^n(x)$$

(8)

where \( \alpha_n \) is the assigned weight:

$$\alpha_n = \frac{1}{2} \ln \left( \frac{1 - e_n}{e_n} \right).$$

(9)

The basic boosting classification algorithm works as follows. Given a set of training samples:

$$\chi = \{(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L) | x_i \in \mathbb{R}^m, y_i \in \{-1, +1\}) \},$$

(10)

where \( L \) is the number of training samples, with positive +1 and negative −1 labels, and an initially uniform distribution over the examples:

$$p(x_i) = \frac{1}{L}.$$  

(11)

A weak learner \( h_{\text{weak}} \) is trained based on \( p(x) \). Based on the error \( e_n \) the weak classifier \( h_{\text{weak}} \) is assigned a weight \( \alpha_n \). \( p(x) \) is updated as the probability of the misclassified samples increases. If a sample is correctly classified, the
corresponding weight is decreased, such that the algorithm will focus on difficult samples. This procedure is repeated until the algorithm converges on a certain stopping criterion.

In the case dealt with in this paper, which concerns on-line boosting, the initial training set is composed only of an initial set of RGB-D frames. From these first frames, an initial training set composed of a series of Haar features is built. Further, the classifier is trained on-line by redetecting the tracked features.

At the 2D image level, the on-line boosting performances have been improved using a classical particle filtering framework [20]. In our experiments we have used a non-linear motion model governed by a set of 200 particles and a constant acceleration of the object.

The boosting tracking architecture provides a 2D ROI in the RGB image, which can be used to track the 3D pose of an object of interest. As will be further detailed below, this pose is obtained from a Lucas-Kanade (LK) optical flow estimation system [21].

3.2 Optical Flow for 3D Object Pose Estimation

The 3D pose of the reference point cloud is calculated based on the 2D ROI obtained from the boosting method and the available 3D point cloud data. 2D point features from the ROI are calculated based on the LK point tracking method [22], as shown in Figure 4. The 2D points used in LK tracking are extracted through the Harris corner detector, followed by correspondence matching using a traditional cross-correlation similarity measure. In order to account for larger motions of the object, a pyramid-based segmentation is used.

For each tracked 2D point in the image, a corresponding 3D point in the point cloud data is available, that is, there exists a direct 2D to 3D correspondence between the points obtained from the optical flow and the 3D point available in the point cloud delivered by the RGB-D system. Keeping this in mind, the consecutive 3D pose of the reference point features \( P_{\text{ref}} \) is determined using a Singular Value Decomposition (SVD) rigid body transformation method.

3.3 Occlusion Detection

One of the main components of the proposed 2D-3D tracker is the occlusion detection system. Its goal is to restart the tracker in case certain obstacles occlude the object of interest. In order to detect occlusions in real time, the ray-casting approach has been used. This means the reference point features \( P_{\text{ref}} \) are reprojected at every frame, based on the depth information available from the RGB-D sensor. The obtained 2D reprojections are used to search the current point cloud \( P_t \) for occlusions, that is, if a point in \( P_t \) has a depth smaller than its correspondence in \( P_{\text{ref}} \), that specific point in the tracked cloud is occluded. An example of occlusion detection can be seen in Figure 5. An occlusion is considered to have taken place if the number of detected point occlusions exceeds a specific threshold value.

4. Visual Pipeline and Experimental Results

The processing pipeline for the proposed tracking system is implemented within the Robotic Operating System (ROS) framework. The input to the tracking module represents a real-world metric-registered point cloud together with its
The evaluation of the overall visual tracking system is performed with respect to the real 3D poses of the objects of interest. Although a lot of 2D image-based object tracking methods exist, the literature is relatively scarce on 3D object trackers, which are a mandatory requirement for service robots operating in uncontrolled real-world conditions. One of the most reliable 3D object tracking systems, which relies only on 2D image information, is the ARToolKit library [23], against which the collaborative tracker presented in this paper has been evaluated.

The real 3D positions and orientations of the objects of interest were manually determined using the following setup. On the imagined scene, a visual marker, considered to be the ground truth information, was installed in such a way that the poses of the objects could be easily measured with respect to the marker. The 3D pose of the marker was detected using the ARToolKit library, which provides subpixel accuracy estimation of the marker’s location with an average error of approx. 5 mm [23]. By calculating the marker’s 3D pose, a ground truth reference value for camera position and orientation estimation could be obtained using the inverse of the marker’s pose matrix. Further, the positions of the object’s features were calculated using the proposed system. Both results, i.e., the 2D and 3D poses, were compared to the ground truth data provided by the ARToolKit marker. The 2D image marker position was calculated using its reprojection in the 2D image plane. As can be seen from Figure 7, the 2D and 3D position errors are within a tolerable range. The statistical analysis of the results is summarized in Table 1. It is important to emphasize here that the ARToolKit tracker was considered to be a reference value, or ground truth, against which the proposed approach was measured. In many situations, the classical marker-based approach fails to deliver proper pose estimation because of different lighting and surface reflection phenomena.

![Figure 6](image_url)  
**Figure 6.** Object tracking loop implementation within the ROS operating system.

<table>
<thead>
<tr>
<th>Max. err.</th>
<th>14</th>
<th>10</th>
<th>0.0967</th>
<th>0.0988</th>
<th>0.0958</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>6.6818</td>
<td>6.4091</td>
<td>0.0281</td>
<td>0.0350</td>
<td>0.0459</td>
</tr>
<tr>
<td>Std. dev.</td>
<td>3.3436</td>
<td>2.5196</td>
<td>0.0253</td>
<td>0.0339</td>
<td>0.0358</td>
</tr>
</tbody>
</table>

**Table 1.** Statistical results of errors between the proposed and the ARToolKit marker based 3D tracking system.

We have considered a processing cycle to begin with RGB-D data acquisition and end with occlusion detection. In the experimental setup, a calibrated MS Kinect® structured light sensor is used to acquire a sequence of 300 indoor images. The average computational time needed by a processing cycle is just over 400 ms, which is low enough to consider the proposed system as a real-time one for the considered case of object manipulation in domestic environments. The implemented tracking architecture has been tested on a typical portable computer running a 64-bit UNIX operating system on an Intel® i3 dual core CPU, each processor having a 2.40GHz clock speed.

Throughout the object handling routine, the required tracking accuracy is strictly dependent on the manipulative task and the configuration of the robot. In particular, the wider the opening angle of the gripper and the higher the number of degrees of freedom of the manipulator arm, the lower the tracking accuracy. In the presented experiments, a tracking error lower than 0.01m was required.
5. Conclusions

In this paper, a 2D-3D object features tracking system has been proposed. It has been proven to stabilize mobile manipulation and provide accuracy in tracking rigid household objects during robotic handling. The proposed approach has been built around a collaborative tracking framework which fuses information from more available trackers. As future work, the author will consider the extension of the collaborative framework with new state-of-the-art trackers in order to improve the accuracy of the proposed system. Also, the speed enhancement of the proposed system using state-of-the-art parallel processing equipment, such as FPGAs and GPUs, would significantly decrease the processing time.

6. Acknowledgment

Sorin Grigorescu was supported by the Sectoral Operational Programme, Human Resources Development (SOP HRD), financed by the European Social Fund and the Romanian Government under project no. POSDRU/15.6/S/15313. Claudiu Pozna was financed by the Széchenyi István University. The authors would like to thank Prof. Michael Beetz and Dejan Pangerlic for their support and constructive ideas.

7. References


