J. Vis. Commun. Image R. 22 (2011) 153–163

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Temporal accumulation of oriented visual features

Nicolas Pugeault [a,*], Norbert Krüger [b]

[a] Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, United Kingdom
[b] Mærsk Mc-Kinney Møller Institute, Syddansk Universitet, DK-5230 Odense, Denmark

ARTICLE INFO

Article history:
Received 1 December 2009
Accepted 7 December 2010
Available online 14 December 2010

Keywords:
Object model building
Visual representation
Feature tracking
Temporal filtering
Unscented Kalman filtering
Edge features
Multiple hypotheses tracking
Structure from motion

ABSTRACT

In this paper we present a framework for accumulating on-line a model of a moving object (e.g., when manipulated by a robot). The proposed scheme is based on Bayesian filtering of local features, filtering jointly position, orientation and appearance information. The work presented here is novel in two aspects: first, we use an estimation mechanism that updates iteratively not only geometrical information, but also appearance information. Second, we propose a probabilistic version of the classical *n*-scan criterion that allows us to select which features are preserved and which are discarded, while making use of the available uncertainty model.

The accumulated representations have been used in three different contexts: pose estimation, robotic grasping, and driver assistance scenario.

1. Introduction

This article presents a framework for on-line generation of an internal representation of unknown objects or scenes, that are observed by the system while subjected to motion. The proposed method is generic and can be applied to any feature. Also, it allows the correction over time of not only feature location, but also appearance information. In contrast, the state-of-the-art focuses on the accumulation of feature position only, while assuming the invariance of the feature's appearance; this invariance does not hold when objects are fully rotated. Moreover, this framework provides a complete representation of objects' edges structure, that makes it useful for a variety of visual as well as robotic tasks—as illustrated in Section 4.

In a first step, local contour descriptors are extracted from the image and reconstructed in 3D using stereopsis.[1] The model itself encodes the object's contours directly in 3D, and associate to them appearance information such as colour. The scene's contours are encoded in this representation as strings of local features called 3D-primitives, that provide a first representation of the 3D shapes in the scene, enriched with appearance information. The appearance information has the quality of being robust under viewpoint changes, and therefore is used to improve matching reliability. At this stage, the representation is merely a collection of 3D-primitives, objects and background are not segmented in any way. By using the motion knowledge provided either by a robot or a separate motion estimation,[2] we segment the object from the scene (by selecting primitives that move according to the robot arm motion) and accumulate the representation. Having control over the object provides a very accurate knowledge of its motion that can be used to track individual 3D-primitives. At each frame, new observations are used to correct the 3D-primitives' full pose and to enrich the representation with new aspects of the object (e.g., parts that were previously occluded).

The mechanism presented herein improves the 3D object model obtained from stereo reconstruction in three respects:

1. Accuracy: The representation is corrected over time using new observations.
2. Reliability: Tracking primitives over time, it is possible to re-evaluate their reliability over time, and to discard erroneous ones. Since the tracking is done in 3D space, the likelihood for erroneous primitives to be tracked successfully is vanishingly small.
3. Completeness: Through manipulation of the object, the system witnesses it under a wide range of viewpoints, and accumulates $2\frac{1}{2}$D representations into a full 3D representation.

* Corresponding author.
E-mail addresses: n.pugeault@surrey.ac.uk (N. Pugeault), norbert@mmmi.sdu.dk (N. Krüger).

[1] Alternatively, shape-from-motion could be used to obtain 3D-primitives. One additional complexity with this alternative is that the reconstruction uncertainty and the motion uncertainty are then related. In this work we focus on stereopsis as it allows for a simpler formulation.

[2] In this work we will mainly show results using motion extracted from the robot arm, for simplicity, but it could also be applied to visually computed motion (as in Fig. 10C and [31]), as long as an estimate of the motion error can be computed. The rationale behind using known motion is that it simplifies the problem and allows a better interpretation of the accumulation error irrespectively of motion estimation accuracy.

1047-3203/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.12.001

This framework requires the capacity to track features over time, and to correct their position using several frames. This is an essential problem in computer vision, and solutions belongs to two groups.

The first group consists of the geometric analytic solutions, including multi-focal tensors [10] and bundle adjustment [40]. These approaches provide optimal solutions to the problem and are prominent for solving the batch structure from motion (SFM) scenario. They can be designed to be robust to erroneous data association (see [40] for a discussion). One major problem of these solutions stems from the fact that they are fundamentally *batch* processes: all views of the object need to be simultaneously available. This can make the problem intractable for large sequences, and implies a large delay for any active system. It has been proposed to split the problem into groups of, e.g., 3 frames, reducing both delay and computational cost [23]. Nonetheless, these approaches face the dead-reckoning problem: small motion errors accumulate over time to lead to large localisation errors. Therefore, they generally need an additional global integration stage. Nistér [24] proposed a live SFM approach based on preemptive RANSAC. Although the method is real-time it enforces strong constraints on feature disparity, and is limited to the estimation of feature position.

The second group uses various flavours of the Bayesian filtering theory. This provides an on-line solution by formalising the problem as a Markov process where the state vector combines both the current pose and the visual features' bearing. This can be formalised as the general Bayesian tracking problem—see [1] for a review. This theoretical formulation allows for an optimal solution, i.e., a Kalman filter [13], if the state vector has a multivariate normal distribution and if the prediction and observation processes are linear. In the mobile robotics context, the object whose model is being incrementally built is the environment itself, described as a set of landmarks. The Kalman filter and its non-linear derivatives (e.g., extended Kalman filter) have been used extensively to solve the so-called simultaneous localisation and map-building (SLAM) problem (see, e.g., [5,42,8,39,21]). Davison [3] proposed a real-time monocular SLAM approach based on EKF. Also Monte Carlo Markov Chains have been used for tracking of multiple targets [15,16,44]. Tao et al. proposed a Bayesian approach for 2D motion segmentation in videos [38].

Because of the on-line constraint, the approach presented in this paper belongs to the second category. One essential difference to typical SLAM systems, is the large number of local features that the system needs to be able to track, to describe the object's shape completely, and the relatively low distinctiveness of these features, whereas SLAM applications generally rely on few sparse yet very distinctive features (e.g., SIFT [22]). Because of this large number of features, we will track each feature individually instead of maintaining a large joint covariance matrix. Moreover, we will track the full pose of the features as well as their appearance properties to make use of temporal information to improve both the accuracy of these appearance cues, but also to generate an estimate over time of their reliability—i.e., how invariant they are when the object is manipulated. The knowledge of this invariance is critical for object recognition and pose estimation. For example, for pose estimation, invariant cues are important for matching, whereas pose-dependent ones are important for estimating the pose.

The novel aspects of this work are:

- Full feature vector tracking: we make use of unscented Kalman filtering (UKF) [12] to track the distribution in the whole feature space, instead of only considering the feature's position. This includes the feature's orientation in space and the observed colour on both sides of the edge. This allows us to keep track of the relative reliability of different components of the feature vector by their filtered variance. It also allows for a straightforward extension to other feature types such as, e.g., junctions (see, [37]) or surface patches.
- Probabilistic matching of features based on both geometric and appearance information.
- Temporal re-evaluation of a feature's confidence according to the tracking success, and probabilistic argument for deletion or preservation of features during occlusions.

The framework is described in Section 2, then evaluated on different scenarios in Section 3. Applications making use of these representations are described in Section 4 before we conclude in Section 5.

2. Methods

In this section, we present the vision framework used to accumulate objects models. First in Section 2.1, we will describe the local features that we use in this work. Note that the framework is generic, and could be applied to any local feature that defines a full pose and some appearance information. Then Section 2.2.5 defines the state space based on such features. Section 2.3 discusses the feature tracking and filtering scheme, based on unscented Kalman filtering (UKF). Finally, Section 2.4 discusses the confidence re-evaluation and the probabilistic n-scan criterion.

2.1. 3D line features extraction

In this work we use sparse image descriptors called visual *primitives*, that exist both in 2D and 3D space, and were discussed in [20,27,33]. In the 2D space, those primitives provide a condensed representation of image information sparsely sampled along image contours. In a first stage, linear and non-linear filtering operations are applied to the image (see, e.g., [11]). These filtering operations provide local information such as the likelihood that a pixel is on an edge, the orientation of this edge, the phase (that contains the type of contrast transition, see [17]) and the colour. Primitives are first extracted at contours and form a feature vector containing the edge position with sub-pixel accuracy, the local orientation, phase (contrast transition), colour on both sides of the edge and optic flow. Positions are detected sparsely with sub-pixel accuracy at places likely to contain edges (see, e.g., [11] for a description). In the following, we refer to such features as *2D-primitives*.

Such 2D-primitives are extracted on stereo pairs of images and are matched using the epipolar line and similarity constraints (see Fig. 1B, and [30] for an assessment). Pairs of matched 2D-primitives provide enough information to reconstruct the 3-dimensional equivalent of a 2D-primitive, denoted *3D-primitive* in the following (see Fig. 1C). We direct the reader to [6,10] for a description on classical stereo reconstruction and [27,33] for the special case of primitives.

A 3D-primitive encodes a scene contour's local position and orientation along with the local contrast and colour on each side

$$s = (\mathbf{p}, \omega, \mathbf{c}) \tag{1}$$

where $\mathbf{p}$ is the full 6D pose in space; ω is the local phase; $\mathbf{c}$ is a 6-dimensional vector encoding the RGB colour values on both sides of the contour. As a consequence, a 3D-primitive is encoded as a 13-dimensional feature vector.

A 3D-primitive's covariance is encoded as the 13×13 block-diagonal matrix Σ

$$\Sigma_i = \begin{pmatrix} \Sigma_{G,i} & \\ & \Sigma_{A,i} \end{pmatrix} \tag{2}$$