Pattern Recognition Letters 30 (2009) 968–976

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Visibility-based modelling and control for network-based robotics

Sang Il Lee [a], Byoung In Cho [a], Jae-Kyu Lee [b,*], Seongjin Ahn [b], Jin Wook Chung [b]

[a] Agency for Defense Development, South Korea
[b] Sungkyunkwan University, Suwon 440 746, South Korea

ARTICLE INFO

Article history:
Received 14 May 2008
Received in revised form 1 April 2009
Available online 22 April 2009

Communicated by B. Kamgar-Parsi

Keywords:
Line-based recognition
Plane sweep
Real plane
Visibility test

ABSTRACT

We present an algorithm to model 3D workspace and to understand test scene for navigation or human computer interaction in network-based mobile robot. This was done by line-based modelling and recognition algorithm. Line-based recognition using 3D lines has been tried by many researchers however its reliability still needs improvement due to ambiguity of 3D line feature information from original images. To improve the outcome, we approach firstly to find real planes using the given 3D lines and then to implement recognition process. The methods we use are principle component analysis (PCA), plane sweep, visibility test, and iterative closest point (ICP). During the implementation, we also use 3D map information for localization. We apply this algorithm to real test scene images and to find out our result can be useful to identify doors or walls in indoor environment with better efficiency.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

1.1. Overview the problem

Recognizing 3D object and estimating its pose by mobile robot in real 3D environment is a challenging task since the scene comes from real environment that has many noises and uncertainties, therefore, its modelling and recognition become difficult. Also the feature information comes from real environment which is so ambiguous and uncertain that any single group of information cannot give significant contribution for recognition. One of the attempts to resolve this problem is evidence fusion based on probabilistic approach by collecting variety of evidences such as colors, textures, lines, and planes. Among the efforts to detect real planes using 3D line segments it can be of unique advantage for line-based recognition since 3D lines which belong to the same object or the same environment are located in the same image plane in spite of changes of viewing angles. However, these 3D lines are scattered around 3D space, therefore, to discover their correspondence to the models directly is a very confusing procedure. Our goal is, first, to classify 3D lines along with their real plane correspondences, second, to reduce ambiguity and uncertainty of model matching process, and finally, to generate better hypotheses to assist Bayesian network and evidence fusion in line-based recognition.

1.2. Scope of this paper

The network-assisted or network-based robotics mean the of contents for robot's main functions are maintained and supported by remote server, and the *terminal robot* is controlled and supported via wireless internet. The advantage of network-based robot is, it can reduce the size and weight of robot platform since lots of main contents for robot are maintained by remotely located main server. There is also cost reduction factor on integration as well. We build a test bed for mobile robot supported by wireless network as shown in Fig. 1. Currently, we install 3D line extraction module on the robot platform for vision-based navigation and human computer interaction. To assist these efforts, we develop real plane search algorithm for line-based recognition. The main scope of this paper is, therefore, on real plane search algorithm for test scene understanding and model matching.

1.3. Our approach

To classify real planes and their coplanar 3D lines, we use plane sweep and visibility test algorithm. As preprocess, we collect a group of 3D line segments. These 3D lines can be obtained from 3D point clouds and from 2D segmentation result. Then, at runtime, we first build reference planes using every 3D line pair. To build reference plane, we use principal component analysis (PCA). Then using plane sweep, we find coplanar 3D lines that belong to each reference plane. Normal distance and line orientation are also measured during coplanarity test in the boolean way. Then we conduct visibility test to prune off ambiguous planes and to classify real planes after we define the explicit planes by applying

* Corresponding author. Tel.: +82 31 290 7212; fax: +82 31 290 7230.
E-mail addresses: happyjoy@add.re.kr (S.I. Lee), fifth@add.re.kr (B.I. Cho), jklee1965@songgang.skku.ac.kr (J.-K. Lee), sjahn@songgang.skku.ac.kr (S. Ahn), jwchung@songgang.skku.ac.kr (J.W. Chung).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.04.006

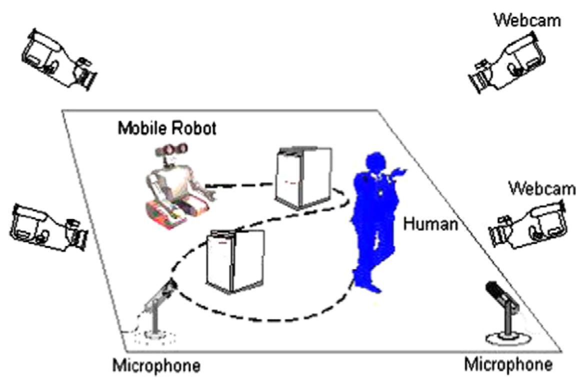

Fig. 1. Schematic diagram of network-based mobile robot. Courtesy of NCSU ADAC Lab.

the maximum and minimum lengths of 3D line pairs to the reference planes. After eliminating ambiguous planes from explicit planes, we patch up real planes by evaluating plane–plane coplanarity. During the whole process, we also use 3D map information of workspace for localization. This 3D map has local information on the region of interest (ROI) and on position of robot's camera as shown in Fig. 1. After having real planes and their corresponding 3D line information, we start on iterative closest points (ICPs) for model match.

Main results: We have implemented our algorithm by Visual C.NET and applied it to real images. The computation time takes less than few hundreds milliseconds based on a dual core 3 GHz Pentium 4 machine. We also present visual outputs from real plane search and scene understanding process. Error analysis and algorithm complexity during plane sweep and visibility test are presented too. The rest of this paper consists as follows: In Section 2, we cover the related works to our algorithm. In Section 3, mathematical concepts and real plane search technique are presented and, in Section 4, we introduce our model matching process. In Section 5, we show our implementation results as well as performance analysis, and in Section 6 we make conclusions of this paper. The network-assisted implementation and hardware integration into platform are now being explored and will be shown in the next report.

2. Related works

Line-based recognition and real plane search (or facade reconstruction) have been extensively investigated by researchers in image processing, computer vision, and intelligent robotics wise guy. The study on plane sweep algorithm and its application also has a wealthy amount of literature. And network-based robotics becomes more popular in these days due to cheaper and faster internet service. In this section, we introduce line-based visualization and recognition, real planes search, and network-based robotics which directly inspired our algorithm.

2.1. Network-based robot

In this subsection, we describe the development of the context–aware framework for a network-based intelligent robot. Recently, while the existing industrial robot market is saturated, the research on the service robot is actively progressed. In order that the service robot comes into our daily lives like the electric home appliances or mobile devices, it has to provide intelligent services to a user. Generally, the robot has three functional components of sensing, processing and acting. The ability of these components depends on its own hardware performance. Network-based service robot (Kim et al., 2005) is the new concept of the service robot. In the concept of network-based service robot, a robot expands the ability of these components through the network. The robot can use not only its internal sensors but also external sensors which are embedded in the environment for sensing. A number of robots can share a high performance server to increase the processing power. As the interests on network-based service robot become more and more, attempts applying it to diverse kinds of research fields have been tried. However, the development of the the application of the network-based service robot has rarely been carried out. In network-based service robot, a system uses context to provide relevant information and/or services to the user, relevancy depends on the user's task (Dey, 2000). Such systems can be implemented in many ways. The approach depends on special requirements and conditions such as the location of sensors, the amount of possible users, the available resources of the used devices or the facility of a further extension of the system (Baldauf et al., 2007). A number of research groups have developed context–aware systems and applications from 1990s. For example, DANTE project (Fong et al., 1995) in NASA. They created a competent robot capable of continuous, reliable and self-reliant exploration. To do so, they created appropriate interface and used network-based participation methods that would result in synergistic human–machine interaction. There are two fundamental reasons to use this system for DANTE project: *ease of understanding* and *ease of use*. To achieve this, they realized the need of system state visualization and understanding as well as to the need to specify and use appropriate control mode by concept of network-based shared control or traded control (Sheridan, 1992). However, most early researches were remained as the experimental level on developing the prototype, because various context information and its achievement technologies were considered only for specific applications at specific platforms. Recent network-based service robot systems use a middle ware or context–aware server architecture for separating an acquisition and use of the context. These architectures have a layered structure that has expendability and reusability. The architecture of the context managing framework (Korpipaa et al., 2003) uses four main functional entities that comprise the context framework: the context manager, the resource servers, the context recognition services and the application. Whereas the resource servers and the context recognition services are distributed components, the context manager represents a centralized server managing a blackboard. It stores context data and provides this information to the client applications. Another framework based on a layered architecture was built in the hydrogen project (Hofer et al., 2002). Its context acquisition approach is specialized for mobile devices. While the availability of a centralized component is essential in the majority of existent distributed content–aware systems, the hydrogen system tries to avoid this dependency. It distinguishes between a remote and a local context. The remote context is information another device knows about, the local context is knowledge our own device is aware of. When the devices are in physical proximity they are able to exchange these contexts in a peer-to-peer manner via WLAN, Bluetooth, etc. This exchange of context information among client devices is called context sharing. Context toolkit (Dey et al., 2001) was developed as an intermediate dealing with context between a sensor and an application service. It collects context information from sensors, analyzes it, and delivers it to application services. The toolkits object-oriented API provides a super-class called base object that offers generic communication abilities to ease the creation of own components. CASS (Context–Awareness Sub-Structure) (Fahy and Clarke, 2004) proposed extensible centralized middleware approach designed for context–aware mobile applications. The middleware contains an Interpreter, ContextRetriever, Rule Engine and SensorListener. The SensorListener listens for updates from

Download English Version:

https://daneshyari.com/en/article/535123

Download Persian Version:

https://daneshyari.com/article/535123

Daneshyari.com