**An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data**

**Abstract:**

Position-estimationsystemsforindoorlocalizationplayanimportantroleineveryday life. The global positioning system (GPS) is a popular positioning system which is mainly eﬃcient for outdoor environments. In indoor scenarios, GPS signal reception is weak. Therefore, achieving good position estimation accuracy is a challenge. To overcome this challenge, it is necessary to utilize other position-estimation systems for indoor localization. However, other existing indoor localization systems especially based on inertial measurement unit (IMU) sensor data, still face challenges such as accumulated errors from sensors and external magnetic ﬁeld eﬀects. This paper proposes a position-estimation algorithm that uses the combined features of accelerometer, magnetometer and gyroscope data from an IMU sensor for position estimation. In this study, we ﬁrst estimate the pitch and roll values based on a fusion of accelerometer and gyroscope sensor values. The estimated pitch values are used for step detection. The step lengths are estimated by using pitch amplitude. The heading of the pedestrian is estimated by fusion of magnetometer and gyroscope sensor values. Finally, the position is estimated based on the step length and heading information. The proposed pitch-based step detection algorithm achieves 2.5 % error as comparedtoacceleration-basedstepdetectionapproaches.Theheadingestimationproposedinthis study achieves a mean heading error of 4.72 degrees as compared to azimuth and magnetometer based approaches. The experimental results show that the proposed position-estimation algorithm achieves high position accuracy that signiﬁcantly outperforms that of conventional estimation methods used for validation in this paper.

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1 GB

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/UBUNTU.
* Implementation : NS2
* NS2 Version : 2.28
* Front End : OTCL (Object Oriented Tool Command  Language)
* Tool : Cygwin (To simulate in Windows OS)