
Detector de quedas baseado em sinal de acelerômetro (simulação)

Julho de 2019

VISÃO GERAL

Quedas são a maior causa de perda de autonomia, morte e ferimentos entre idosos. De acordo com a World Health Organization (Yoshida S., 2007), cerca de 30% da população mundial com idade acima de 64 anos sofrem pelo menos uma queda por ano. Como os danos provocados por quedas dependem também da velocidade de tratamento, é justificável desenvolver sistemas de detecção de quedas.

Atualmente, diversos estudos propõem e analisam sistemas para detecção de quedas. Dentre os estudos, o sinal mais utilizado para extração da resposta de queda é o sinal de acelerômetro, por seu baixo custo e alta discriminação em situações de queda. Após a aquisição do sinal, o mesmo é processado e a informação de queda é extraída através de mecanismos personalizados e/ou métodos de inteligência computacional. O detector então é embarcado num sistema fixo ao corpo do usuário, que deverá se comunicar relatando o estado atual. O presente projeto se propõe a simular o sistema embarcado em questão, e também o terminal monitor com o qual ele se comunicará.

OBJETIVOS

1. Implementar a simulação do sistema embarcado, que recebe sinal do acelerômetro e extrai informação de queda do mesmo.
2. Implementar monitor, que recebe informação de queda do sistema embarcado e toma uma ação de controle.

ESPECIFICAÇÕES

Para que se possa implementar tal sistema, deverá ser escolhido um ou mais conjunto de dados com sinal de acelerômetro e situação de queda, assim como as etapas de processamento e o algoritmo de classificação. Uma vez que o sistema embarcado detecta situação de queda, a informação deve ser enviada a um monitor via sockets, para que esse tome a devida ação de controle, como enviar SMS, email, etc, notificando um ou mais responsáveis da queda. Se possível, poderia haver integração direta com serviços de saúde.

IMPLEMENTAÇÃO DO SISTEMA

Repositório: https://github.com/vctrop/embedded_fall_detection_simulator

MONITOR

O monitor foi implementado com objetivo de fazer as seguintes requisições ao dispositivo:

- Obter localização;
- Habilitar envio dos dados adquiridos
- Desabilitar envio dos dados adquiridos

O mesmo faz uso de uma lista encadeada, armazenando as informações necessárias para comunicação com cada dispositivo em um nó diferente.

Ao receber um buffer, o monitor checa as primeiras posições do mesmo para decidir quais dados devem ser exibidos (Tabela 1 abaixo).

DISPOSITIVO

I. Threads periódicas

Foram implementadas três threads periódicas, sendo duas delas responsáveis por aquisição de sinais e uma pelo processamento dos mesmos.

- A. Thread `read_gps()`: lê informações de latitude e longitude de um banco de dados de localização GPS (CRUZ, M. O.; MACEDO, H.; GUIMARÃES, A. P.) a cada 5 segundos, atualizando a posição atual do dispositivo. Caso chegue no fim do arquivo, reinicia o descritor do mesmo.
- B. Thread `read_accelerometer()`: lê dados de aceleração de um banco de dados de sensor acelerômetro adquirido com um smartphone Samsung Galaxy (UniMiB SHAR dataset) a uma taxa de 50 Hz ($T = 0,02$ s), inserindo a componente vertical de cada amostra num **buffer circular duplo**.

-
- C. Thread estimate_fall(): a cada 3,02 segundos, a thread lê uma janela de 151 amostras da metade do buffer que estiver marcada como limpa, calcula o **valor médio**, centraliza e calcula o **número de vezes que o sinal cruza zero**, e usa esses dois números (média e número de cruzamentos) para determinar se houve queda ou não, baseado num modelo linear encontrado previamente segundo o modelo de **Support Vector Machines** linear (aprendizado de máquina).

II. Threads de comunicação

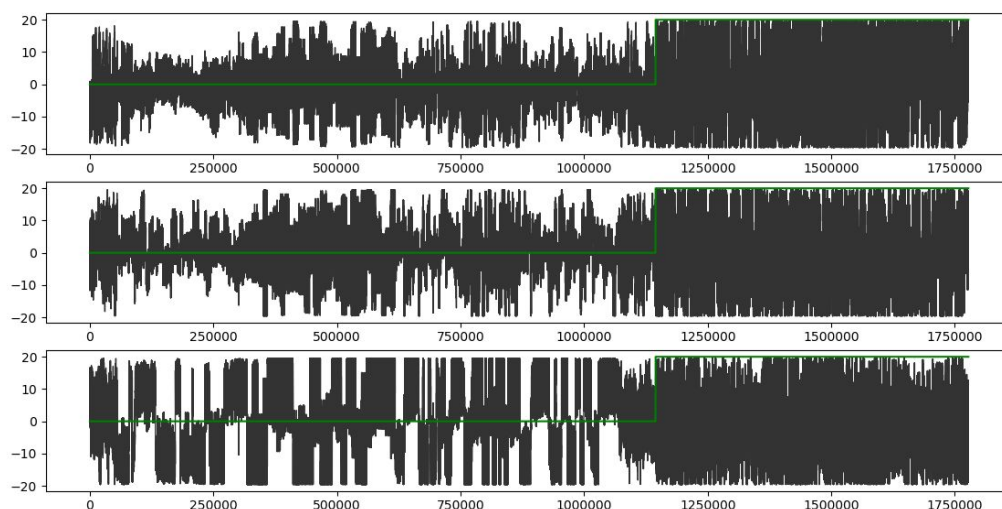
Tabela 1: organização do buffer enviado

Fall detected (1 byte)	flag_location (1 byte)	flag_display (1 byte)	location data (34 bytes)	sensor data (1661 bytes)
---------------------------	---------------------------	--------------------------	-----------------------------	-----------------------------

- A. Thread read_socket(): essa thread tem como objetivo receber e interpretar as requisições do monitor, levantando ou baixando as flags **flag_location** e **flag_display**.
- B. Thread write_socket(): para decidir se deve haver envio de dados, a thread utiliza uma variável de condição sobre a flag **send_socket**, sendo a mesma levantada quando ocorre uma queda, há requisição de localização ou também quando o envio de dados está habilitado enquanto o índice do buffer circular duplo cruza um ponto crítico (meio ou fim do buffer). Em qualquer dessas ocasiões haverá envio de dados, e quais dados estarão presentes no buffer dependem das respectivas flags. Caso flag_location esteja alta, **latitude e longitude** serão armazenadas no buffer de envio. Caso flag_display esteja alta, a **parte limpa do buffer do sensor** é armazenada no buffer de envio.

Sinal de acelerômetro

Para realizar a detecção de queda, foi necessário processar o sinal de acelerômetro e alimentar um algoritmo de classificação com o resultado desse processamento. Os três eixos do sinal utilizado podem ser visualizados na figura abaixo (neste trabalho, apenas o eixo vertical, no meio, foi utilizado). Em verde, é possível visualizar a indicação da ocorrência de queda nas amostras finais.



TUTORIAL

Para utilizar o sistema, basta executar monitor e dispositivo com o mesmo argumento, a porta utilizada.

Uma vez que a conexão esteja pronta, é possível enviar através do monitor requisição de localização ao enviar o caractere “l”. Para habilitar o envio dos dados do sensor pelo dispositivo, envia-se “e”, e para desabilitar, “d”.