
Detector de quedas baseado em sinal de acelerômetro (simulação)

23 de maio 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.

DESENVOLVIMENTO

1. ATUAL

1.1. MONITOR

Atualmente, a função `main()` do monitor executa as ações de inicializar um descritor de socket com protocolo INET e atuando por stream de dados, configurar um endereço de socket para uma determinada porta, fazer bind do descritor com esse endereço e esperar por novas conexões. Após cada accept, as informações relevantes à conexão detector-monitor são inseridas numa lista encadeada e uma thread é criada.

1.2. DETECTOR

Há três threads periódicas no detector: `read_gps()`, `read_accelerometer()` e `estimate_fall()`. Dentre elas, apenas a `read_gps()` está de fato implementada, carregando dados de um arquivo .csv de forma circular e atualizando a posição atual.

2. FUTURO

2.1. MONITOR

As threads que gerenciam as conexões com cada dispositivo detector devem gerenciar envio de requisições e recebimento de dados. As seguintes requisições serão possíveis: receber localização GPS, ativar e desativar exibição de dados do acelerômetro em tempo real.

Quanto aos dados recebidos, haverá o alerta de queda, dado esporádico que só chega quando o dispositivo detecta uma queda, a localização GPS, que depende de uma requisição do monitor e os dados do acelerômetro, que são continuamente enviados após ativação.

2.2. DETECTOR

Futuramente, a thread `read_accelerometer()` lerá dados de um arquivo .csv para um buffer circular, de onde `estimate_fall()` deve ler uma janela de valores, para então processar esses dados e através de uma função de decisão (previamente

encontrada por um algoritmo de classificação) determinar se houve queda. Caso haja queda, a mesma deve habilitar que a thread `write_socket()` envie um alerta para o monitor. De acordo com as requisições recebidas pela thread `read_socket()`, a thread de escrita enviará também a posição atual do dispositivo, assim como os dados lidos do acelerômetro.