

HologramHand seguimiento de lesiones

Amengual Miguel Angel, Castagna Cristian Damian, Krause Ezequiel Otto, Scarpello Franco Dario

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
amengualma6@gmail.com, cristiancastagna_1993@hotmail.com, ezeqkr@gmail.com,
scarpello.franco@gmail.com

Resumen. En el presente documento se desarrolla la funcionalidad de detección de daño motriz a emplear en el proyecto HologramHand. Para esto se empleará un modelo de procesamiento basado en MPI en donde se genera una red conformada por todas las HologramHand que se encuentren en el mercado. A mayor cantidad de HologramHand disponibles, mayor será el poder de procesamiento conjunto, necesario para resolver la evaluación de daño en la mano del paciente dentro de una única HologramHand. Se pretende con el mecanismo descrito, ahorrar en los costos debidos a soluciones basadas en adquirir nuevo hardware, aprovechando los tiempos ociosos de los procesadores que ya están disponibles por el simple hecho de haber adquirido una unidad de HologramHand.

Palabras claves: HologramHand, HPC, MPI.

1. Introducción

HologramHand es un proyecto basado en la manipulación de hologramas a partir de gestos realizados con la mano. Para cumplir su objetivo principal, cuenta con sensores que cumplen la función de medir la curvatura de los dedos y un acelerómetro. En este documento ahondaremos en la capacidad de evaluar la recuperación de la mano de un usuario que allá sufrido una lesión en esta.

Generar esta evaluación será una característica que aporta un valor agregado. Sin embargo, es algo complejo que requiere de un poder de procesamiento mayor al que puede tener un único Arduino

R1. Para ofrecer esta funcionalidad comenzaremos explicando cómo se obtendrán las mediciones del movimiento en los dedos del paciente, luego explicaremos como sera procesada la información, y por ultimo como podrá esto ayudar al usuario.

2. Desarrollo

Obtención de la medición

Los movimientos que realizan los dedos de la mano de una persona que se encuentra utilizando el dispositivo (HologramHand), son capturados por los sensores flex, uno por cada dedo de la mano, los cuales convierten la curvatura de su flexión en valores de resistencia eléctrica.

El sistema embebido implementará el filtro de Kalman^[1], el cual nos servirá para poder obtener mediciones más limpias de los sensores flex. El filtro de Kalman es un algoritmo recursivo y utiliza un modelo predictivo y correctivo.

Procesamiento de los valores obtenidos

Para conseguir el poder necesario, se utilizará un sistema de comunicación punto a punto^[2], en donde cada nodo conectado a la red estará conformado por un dispositivo HologramHand. El nodo que realice la solicitud de procesamiento al resto de los nodos mediante mensajes MPI (de ahora en adelante “nodo maestro”) recibirá como respuesta la cantidad de sensores que puede procesar de cada nodo que responda la solicitud (desde ahora “nodos esclavos”).

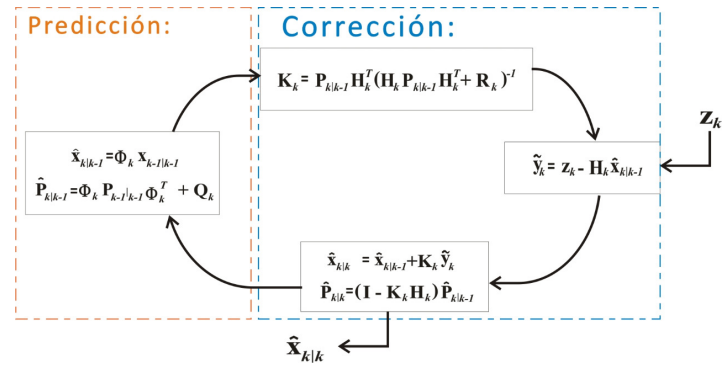
El nodo maestro informa a los nodos esclavos los registros de su memoria interna a la cual deben acceder para leer los valores obtenidos por los sensores flex. Por defecto solo se solicitará a los nodos esclavos que procesen la información de un único sensor flex.

Una vez que los valores son leídos por los nodos esclavos, estos generarán una porción del informe del estado de la mano.

Luego de que el nodo receptor haya arme el informe, el mismo es devuelto al nodo maestro.

Interpretación de los valores procesados

El nodo maestro recibe de los nodos esclavos las porciones de informe del estado de la mano, será su trabajo unificar los informes recibidos y presentarlos al usuario de HologramHand.



3. Explicación del algoritmo

Al recibir el anuncio desde la aplicación de crear una evaluación de seguimiento de la recuperación de la mano, este HologramHand pasara a ser el nodo Maestro, entonces se dispara la ejecución del siguiente algoritmo^[3].

NODO MAESTRO

realizarSolicitudEvaluacion();

solicitarProcesamientoExterno();
indicarRegistrosDeMemoria();

unirInputsDeInforme();
enviarInforme();

NODO ESCLAVO

MPI_INIT();
escucharSolicitudesProcesamiento();

responderPosibilidadDeProcesar();

leerRegistrosDeMemoria();
generarInputDeInforme();
responderInputDeInforme ();
MPI_Finalize();

4. Pruebas que pueden realizarse

Una de las pruebas que pueden realizarse es implementar el algoritmo con y sin el empleo de MPI, para así demostrar de forma certera la diferencia en los tiempos de ejecución entre una y otra.

Para poder llevar a cabo la propuesta debe haber en el mercado suficientes unidades de HologramHand como poder de procesamiento sea necesario, a mayor cantidad mayor poder de procesamiento. Cuando el usuario solicite el informe de la mano, se realizará la solicitud a todos los HologramHand que se encuentren disponibles para que puedan ayudar en el procesamiento de la información.

Finalizado el procesamiento, el usuario podrá visualizar el informe de su mano en un dispositivo móvil que tenga la aplicación de HologramHand instalada. La misma se comunicará vía WiFi con el sistema embebido, el cual brindará el informe para luego ser mostrado en el dispositivo móvil.

5. Conclusiones

El seguimiento de una lesión con los medios actuales se vuelve tedioso e inclusive por esta razón es un proceso que suele ser abandonado por los pacientes. Es por este motivo, que la solución planteada está basada en que los usuarios del HologramHand puedan aprovechar al máximo los recursos que ya disponen, hagan el seguimiento de una forma mucho mas cómoda, y ofrecer una mejora al producto, sin elevar el costo final significativamente.

Como el informe está basado en la información brindada solo por los 5 sensores de los dedos, es lógico que el resultado tenga ciertas limitaciones.

Para finalizar, agregamos, que con la misma lógica pero usando otros sensores se podrían generar distintos tipos de evaluaciones, por ejemplo con el giroscopio se podría evaluar la condición de la muñeca y así generar un informe final aun más completo.

6. Referencias

1. An Elementary Introduction to Kalman Filtering (2019), <https://arxiv.org/pdf/1710.04055.pdf>
2. Computación de Alto Rendimiento, MPI y PETSc (2016), <https://cimec.org.ar/~mstorti/curso-mpi-petsc/slides.pdf>
3. Performance study of multithreaded MPI and

OpenMP tasking in a large scientific code (2017),
[https://www.researchgate.net/publication/
318127376_Performance_Study_of_Multithreaded_MPI_and_OpenMP_Tasking_in
a_Large_Scientific_Code](https://www.researchgate.net/publication/318127376_Performance_Study_of_Multithreaded_MPI_and_OpenMP_Tasking_in_a_Large_Scientific_Code)