Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Estimación de pose de manos bajo consideración de relaciones topológicas

Documento de tesis sometido a consideración para optar por el grado académico de Maestría en Electrónica con Énfasis en Procesamiento Digital de Señales

Randall J. Esquivel Alvarado

Cartago, 25 de enero, 2018

Declaro que el presente documento de tesis ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos y resultados experimentales propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de tesis realizado y por el contenido del presente documento.

Randall J. Esquivel Alvarado

Cartago, 25 de enero de 2018
Céd: 5-0357-0111

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Tesis de Maestría

Tribunal evaluador

Tesis de maestría defendida ante el presente Tribunal Evaluador como requisito para optar por el grado académico de maestría, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

M.Sc. Melissa Montero Bonilla
Profesora lectora

Dr.-Ing. Juan Luis Crespo Mariño
Profesor lector

Dr.-Ing. Pablo Alvarado Moya
Director de Tesis

Los miembros de este Tribunal dan fe de que la presente tesis ha sido aprobada y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 25 de enero de 2018

Resumen

En este trabajo se realiza un estudio basado en el método de predicción de la posición de las articulaciones de la mano, tomando en cuenta la información de la jerarquía real de las articulaciones principales en un proceso de entrenamiento, con la finalidad de reducir la cantidad de datos a procesar y evitar hacer evaluaciones aleatorias que aportan poca información.

El método utilizado se basa en entrenar árboles binarios con información de la jerarquía de las articulaciones de la mano, el cual es utilizado para guiar el proceso de entrenamiento de un conjunto de árboles binarios de predicción y con esto hacerlo más eficiente.

Se hace un estudio del efecto de cada una de las variables que afectan el proceso de entrenamiento considerado en el entrenamiento y predicción de las articulaciones.

Palabras clave: Predicción, árbol binario, árbol de regresión, modelo de árbol latente, bosque de árboles aleatorios

Abstract

In this work a study based on the method of prediction of hand joint position is made, taking into account the information of the real hierarchy of the hand joints in a training process, in order to reduce the amount of data to process and avoid making random evaluations that provide less relevant information.

The method used consists on the training of binary trees using information for the hand joint hierarchy, which is used to guide the training process of a set of binary trees and thereby make it more efficient.

A study is made of the effect of each one of the variables that affect the training process in the training and the prediction of the joints.

Keywords: Prediction, binary tree, regression tree, latent tree model, random regression forest

a mis queridos padres

Agradecimientos

Le agradezco especialmente a mi famila por el apoyo en durante todo el proceso de estudio e investigación en el cual se ha desarrollado esta maestría, por sus palabas de aliento y motivación durante el tiempo que ha sido necesario.

Gracias a la Escuela de Ingeniería Electrónica por permitirme realizar este proyecto de investigación y específicamente al profesor Pablo Alvarado por la asesoría durante el desarrollo de este trabajo.

Randall J. Esquivel Alvarado

Cartago, 25 de enero de 2018

Índice general

Índice de figuras

Índice de tablas

Lista de símbolos y abreviaciones

Abreviaciones

CLNJ	Chow-Liu Neighbor Joining
LRF	Latent Regression Forest
LRT	Latent Regression Tree
LTM	Latent Tree Model
RDF	Random Decision Forest
RDT	Random Decision Tree
SIP	Segmentation Index Points

Notación general

$\mathbf{A}$	Matriz. $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$
Φ	Conjunto de características a evaluar.
ϕ_i	Característica i a evaluar.
ρ_i^I	Posición 3D del nodo i en la imagen I
τ_i	Umbral para la característica i
$tr(\cdot)$	Función traza
$\underline{\mathbf{x}}$	Vector. $\underline{\mathbf{x}} = [x_1 \; x_2 \; \ldots \; x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

Capítulo 1

Introducción

El reconocimiento de pose de mano es necesario cuando se desea implementar una interfaz humano-máquina que permita la interacción del usuario con un sistema computacional de forma natural utilizando modalidades de comunicación no verbales. Estas interfaces tienen el objetivo de evitarle al usuario la complejidad técnica del uso de otro tipo de dispositivos de entrada al sistema.

Existen dispositivos físicos como guantes para detectar la pose de la mano [15], cuya naturaleza intrusiva no es conveniente en escenarios donde el usuario interactúa esporádicamente o require libertad total de movimiento. Este trabajo se concentra en el estudio de métodos no intrusivos basados en la interpretación del contenido de imágenes digitales.

Los algoritmos utilizados en el reconocimiento de pose de mano se basan algunos en la estimación de la posición de cada una de las articulaciones directamente de la información obtenida de la imagen de la mano, mientras que otros se basan en el ajuste de un modelo tridimensional a la imagen para obtener la posición de las articulaciones.

Los sistemas de visión por computador para reconocimiento de pose de mano y en particular para el reconocimiento de lenguaje de señas ha atraído la atención de investigadores por aproximadanmente 20 años [15]. Sin embargo, la disponibilidad de dispositivos con la capacidad de capturar imágenes con información de profundidad y sistemas computacionales de bajo costo que puedan procesar los algoritmos necesarios para la detección de pose no han estado disponibles hasta recientes años.

1.1 Imágenes RGBD

Los primeros métodos para estimación de mano utilizaban imágenes en escala de grises o en color. La estimación de la pose de la mano a partir de la información en dichas imágenes se ve afectada por variaciones en el proceso de formación de las imágenes como cambios en la iluminación, variabilidad en la vestimenta de los usuarios y la dificultad

para segmentar la mano de otras partes del cuerpo o de la escena capturada. Algunas implementaciones requerían el uso de sistemas con múltiples cámaras para poder detectar la profundidad, lo que conlleva mayor demanda de ancho de banda para transferir las imágenes al computador y mayor cantidad de datos que procesar, en comparación al caso de una única cámara.

Más recientemente, con el desarrollo de los sensores RGBD es posible obtener información de la profundidad de cada punto en el espacio correspondiente a un píxel de la imagen bidimensional [23] a partir de un mismo dispositivo. Estos dispositivos se clasifican en sensores de luz estructurada y sensores de tiempo de vuelo (ToF por las siglas en inglés para Time of Flight) de un haz de luz.

En los sensores que operan con luz estructurada la información de profundidad es obtenida mediante la proyección de un patrón de luz infrarroja, el cual es capturado luego por una cámara infrarroja integrada, con lo cual es posible reconstruir un mapa de profundidad para cada punto de la imagen a partir de la deformación del patrón conocido proyectado y la correspondiente imagen capturada para dicho patrón.

Los sensores de ToF se basan en el principio de que la profundidad de cada punto de la escena puede ser obtenida a partir de un conjunto de haces de luz emitido midiendo el cambio de fase experimentado por la luz reflejada al ser capturado de nuevo por el sensor [13]. Estos sensores representan una mejora con respecto a los de luz estructurada puesto que permiten obtener información de profundidad de objetos con menor área, debido a que los de luz estructurada necesitan un grupo de puntos proyectados en un mismo objeto para poder detectarlo.

En ambos tipos de sensores la salida generada proporciona un mapa de profundidad y una imagen RGB como la generada con una cámara convencional, tal como se muestra en la figura 1.1.

Figura 1.1: Ejemplo de imágenes proporcionadas por un sersor RGB-D. Se puede observar una escena y el mapa de profundidad respectivo. Tomado de [1]

Tomando en cuenta la imagen generada por este dispositivo se puede determinar la distancia a la cual se encuentran distintos puntos en la escena y si forman parte de un mismo objeto, considerando dicha información de profundidad. En el caso del reconocimiento de pose, estos datos son utilizados para detectar partes del cuerpo considerando los patrones en los cambios de profundidad entre puntos que indican los límites del mismo. Esta información de límites obtenidos de la información de profundidad se utiliza tanto en el

caso de detección de pose del cuerpo como en la detección de la pose de los dedos de la mano.

1.2 Sistemas de estimación de pose

El modelado de la mano en sistemas de interfaz humano máquina es esencial cuando se requiere capturar la compleja interacción de las distintas articulaciones que la conforman para poder ser utilizado en un sistema computarizado [25]. Existen sistemas de captura de información de la mano utilizando hardware de detección en guantes [8] y otros que utilizan de puntos de identificación directamente sobre la mano [33] para ajustar un modelo a una determinada imagen de entrada. A pesar de la elevada precisión y baja latencia, la intrusividad de estos sistemas interfiere con la correcta movilidad de la mano.

Puesto que la mano está conformada por varias articulaciones y cada una de ellas tiene de uno a tres grados de libertad de movimiento [30] el modelo que mejor se ajusta para su representación es el conformado por 24 puntos [5] pues así cada una de las articulaciones y las terminaciones de los dedos tienen su representación. En investigaciones recientes [32], [28] se ha utilizado un modelo simplificado de la mano con el objetivo de reducir la complejidad computacional haciendo uso solamente de 16 puntos, en el cual se obvian las articulaciones de la palma de la mano, considerando esta como un cuerpo rígido [9] y las puntas de los dedos, dando como resultado un diagrama como en mostrado en la figura 1.2.

En el reconocimiento de pose de la mano mediante métodos no intrusivos para ajustar un modelo de la mano a la información de una imagen de entrada, se han utilizado dos tipos de enfoque [29]:

- Generativo: en este enfoque se genera un modelo tridimensional de la mano, el cual se ajusta a una imagen de entrada para estimar la posición de las articulaciones en dicha imagen.
- Discriminativo: Estos modelos se basan en el análisis de los datos proporcionados por una imagen, de tal forma que la pose pueda ser clasificada como una de las poses previamente definidas.

1.2.1 Estimación de pose usando un enfoque generativo

En el enfoque generativo se toma una imagen de entrada y se ajusta un modelo tridimensional de la mano previamente diseñado a la pose actual y a partir de la posición obtenida con del modelo se determina la localización de cada una de las articulaciones. En el trabajo presentado en [22] representa la mano por medio de un modelo tridimensional a partir de imágenes tomadas con cámaras RGB-D, basado en un modelo de la

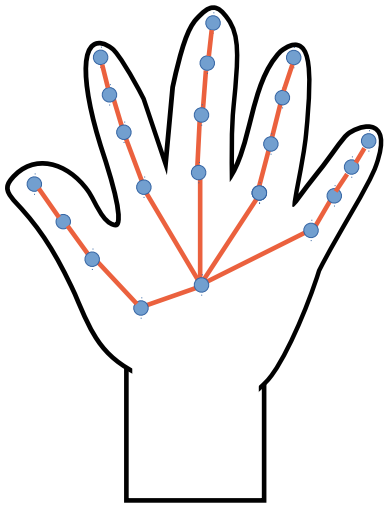


Figura 1.2: Modelo de mano simplificado

mano compuesto por 21 articulaciones. Este método depende de un modelo de la mano previamente construido, el cual es adaptado a la forma de mano detectada por la cámara, y además depende de la información de cuadros anteriores para generar el modelo. El algoritmo de ajuste utiliza algoritmos genéticos y necesita una implementación en GPU para lograr un modelado en tiempo real. El trabajo realizado en [18] propone una mejora en el algoritmo de ajuste de un modelo tridimensional a las articulaciones de la mano con la finalidad de reducir el error de aproximación considerando características anatómicas.

1.2.2 Enfoque discriminativo

Los métodos que utilizan un enfoque discriminativo toman una imagen de entrada y esta se analiza directamente para asignar a un determinado punto de la imagen una articulación correspondiente de acuerdo con un algoritmo de clasificación. El algoritmo presentado en [17] hace uso de algoritmos de segmentación para determinar la posición y pose de la mano, partiendo de imágenes de un brazo, haciendo uso de árboles aleatorios de decisión para la estimación de pose. En [10] se hace segmentación de la posición de la mano utilizando en una escena a partir de imágenes de la vista superior; en este caso se utilizan árboles de decisión aleatoria como clasificador para detectar la ubicación de la mano.

En la estimación de pose una técnica de clasificación utilizada son los bosques de decisión aleatorios o RDF por las siglas en inglés para Random Decision Forest, introducidos en [2]. Se crea durante el entrenamiento un conjunto de árboles de decisión en los cuales para cada nodo se genera un conjunto de características Φ que son desplazamientos aleatorios alrededor de un punto evaluado en la imagen, así como un conjunto aleatorio de umbrales de decisión τ. Esta técnica ya ha sido utilizada para predecir la posición de las articulaciones de la mano [15], [24]. En cada uno de los nodos se considera un punto a evaluar en el que se deben elegir la característica y el umbral que produzcan la mayor ganancia de información de los puntos al evaluar una función de separación f que divide

las características en dos grupos, de forma tal que al evaluar $f(\phi_i) < \tau_j$ un punto es clasificado como perteneciente al grupo izquierdo y $f(\phi_i) \geq \tau_j$ para el grupo derecho.

El proceso de predicción evalúa una imagen de entrada utilizando los árboles generados durante el entrenamiento. En cada uno de los nodos de los árboles aprendidos solamente el umbral elegido como óptimo durante el entrenamiento es utilizado para decidir si un punto debe ser clasificado en uno de los dos grupos posibles, que le permita continuar su evaluación por alguno de los nodos hijos, izquierdo o derecho, del nodo evaluado. El entrenamiento produce que cada uno de los árboles sea distinto y evalúe diferentes características en cada nodo. Por lo tanto, el resultado de la predicción es una votación estadística en la cual se toma en cuenta la decisión final tomada en cada árbol para cada uno de los puntos seleccionados en la imagen de entrada, para obtener el resultado que ha obtenido más votos al considerar todos los árboles.

La principal ventaja de los RDF es que, a pesar de tener alto costo computacional para su entrenamiento, el proceso de decisión utiliza únicamente una secuencia de comparaciones que es relativamente de baja demanda computacional en comparación a otros clasificadores tradicionales como por ejemplo redes neuronales o máquinas de soporte vectorial, y por su naturaleza permite entrenar varias instancias de árbol en paralelo [6]. Esta sigue siendo la ventaja fundamental de los algoritmos con árboles de regresión, debido a que hasta ahora los algoritmos de aprendizaje profundo, utilizando redes neuronales, no han podido ser utilizados en tiempo real sin la paralelización lograda por medio de aceleración por hardware con el uso de GPU [31].

1.2.3 Enfoque discriminativo usando información topológica de la mano y árboles de regresión

Recientemente se ha utilizado una nueva forma de clasificación de datos a partir de la combinación de información generada en un árbol con nodos latentes (LTM por las siglas en inglés para Latent Tree Model) con un árbol de regresión [28]. La propuesta presentada en dicha investigación consiste en la utilización de un modelo de mano que permite establecer la conexión entre las diferentes articulaciones que conforman la mano y con ello hacer más eficiente el proceso de entrenamiento y de estimación de la posición de cada articulación. La figura 1.3 muestra el esquema en el cual se utilizan los conceptos de LTM para el modelo de la mano y como clasificador un nuevo enfoque denominado Bosques Latentes de Regresión, los cuales son presentados en las secciones siguientes.

El uso de árboles latentes como parte del proceso de decisión interno permite incluir nodos que tienen la finalidad de abstraer información presente en las variables observadas. Las variables observadas son datos medibles o generados en un proceso de entrenamiento y las variables latentes son variables intermedias que abstraen información común a un grupo de nodos particular la cual no es directamente medible en las variables observadas [19].

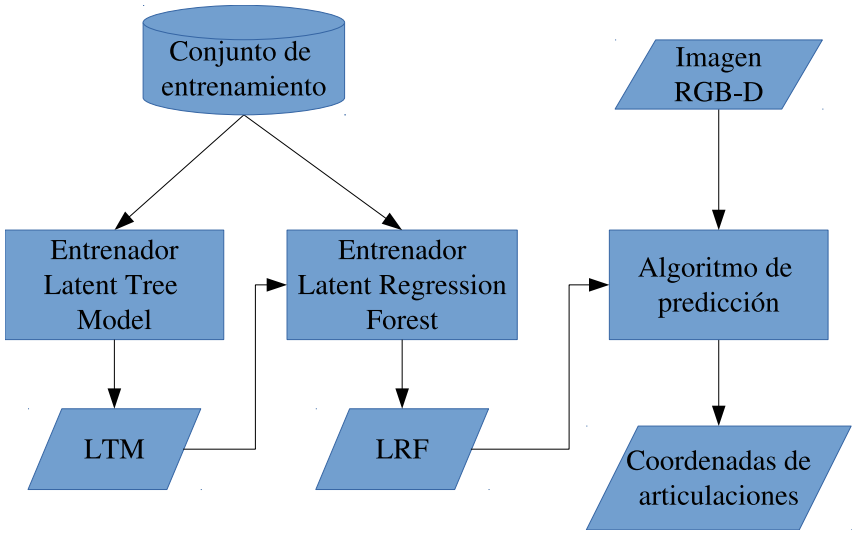


Figura 1.3: Diagrama general del sistema

Entrenamiento

El proceso de entrenamiento genera primero un modelo de la mano utilizando nodos latentes a partir de la información de la pose de la mano tomada de un conjunto de imágenes utilizadas solamente para el entrenamiento. Luego, con la información sintetizada en el LTM, se guía el aprendizaje de un conjunto de árboles de regresión. La guía reduce el tiempo requerido para entrenar en comparación con un sistema discriminativo que clasifica cada píxel con un RFD. Algunos trabajos desarrollados en [26], [22] hacen uso de árboles de regresión para la segmentación y estimación de pose de manos; sin embargo, el trabajo presentado que hace un entrenamiento previo de LTM para guiar el proceso de entrenamiento de los árboles de regresión es presentado en [28]. La información generada durante el entrenamiento correspondiente a los árboles de regresión es almacenada para ser utilizada como entrada del algoritmo de predicción, de forma que deba ser generada solamente una vez.

Predicción

Durante el proceso de predicción inicialmente se cargan los datos generados durante el entrenamiento con lo que se reconstruyen los árboles binarios generados para mantenerlos en memoria RAM. Una vez cargados los árboles, el algoritmo de predicción recibe como entrada la imagen de una pose de mano a identificar, cuya información es utilizada en cada uno de los árboles generados en el entrenamiento y el resultado del conjunto de árboles es utilizado para determinar las posiciones del conjunto de articulaciones que conforman el modelo utilizado para la mano. El resultado obtenido en cada uno de los árboles es considerado para cada articulación y la posición final se calcula utilizando la media de la posición obtenida para una determinada articulación en todos los árboles.

El presente trabajo toma la propuesta de Tang, et al. [28] que sigue la estrategia descrita

en esta sección como punto de partida y con base a la evaluación de los resultados del método propone varios cambios en los procesos.

1.3 Datos de entrenamiento y retos en la estimación

Los datos de entrenamiento utilizados en los sistemas de estimación de pose de manos se pueden dividir en las siguientes categorías:

- **Imágenes reales con anotación manual** En esta modalidad las imágenes son capturadas usando cámaras RGBD y la posición de cada una de las articulaciones es asignada manualmente por personas al conjunto de imágenes. Este método es propenso a la introducción de errores humanos en la asignación de las posiciones y dado el tamaño utilizado generalmente para las bases de datos, la generación de las imágenes anotadas necesarias es impráctico.
- **Imágenes reales con anotación automática** La anotación automática de imágenes reales en este caso depende del uso de guantes con sensores para ubicar cada punto de interés, los cuales deben ser cuidadosamente calibrados, pero tienen la desventaja de que interfieren con el movimiento natural de la mano y pueden alterar la imagen misma.
- **Datos seudo sintéticos** Es un conjunto de imágenes generado parcialmente con imágenes reales y anotado manualmente pero que es extendido utilizando variaciones computacionales de las imágenes existentes, como rotaciones y si se tiene acceso a múltiples cámaras se pueden tener distintos ángulos para la misma captura.
- **Datos completamente sintéticos** En este caso las imágenes son generadas usando un modelo tridimensional computarizado que genera directamente las imágenes de profundidad con las correspondientes anotaciones evitando los problemas relacionados con este paso.

Los resultados obtenidos con los distintos algoritmos de estimación disponibles dependen de exactitud con la que ha sido asignada la posición de los datos en las imágenes utilizadas para el entrenamiento, así como de otras variables intrínsecas a la escena capturada. La detección tiene un mayor porcentaje de acierto si las imágenes muestran la mano completa y con alta resolución.

Algunos factores inherentes al movimiento de la mano como rotación, el ángulo de la mano y el traslape entre dedos son retos importantes en la asignación de la posición de las articulaciones. También factores como ruido de fondo hacen el reconocimiento más complejo debido a la información adicional que debe ser filtrada de los datos de entrada.

1.4 Problema y esbozo de la solución

El método del que se parte en este trabajo [28] tiene como desventaja la dependencia del punto de partida para la estimación de la mano, pues en el proceso se propaga el error de estimación.

La investigación realizada en este trabajo evalúa alternativas de una mejora a los algoritmos de estimación de pose de mano, los cuales se encuentran en pleno desarrollo actualmente.

Es necesario evaluar los conceptos propuestos en recientes investigaciones, para corroborar el funcionamiento y validez de estos algoritmos alternativos para la predicción de pose de manos.

1.5 Objetivos y estructura del documento

El objetivo de este trabajo es realizar una investigación tomando como punto de partida algoritmos recientes de cálculo de posición de las articulaciones de la mano utilizando sus características topológicas para guiar el entrenamiento y lograr con esto una mayor eficiencia en comparación con los algoritmos discriminativos.

Este documento introduce en el capítulo 2 el estado del arte de los algoritmos de reconocimiento de manos. El capítulo 3 presenta en detalle los métodos utilizandos en este trabajo y la solución desarrollada. En el capítulo 4 se muestran los resultados obtenidos con los métodos que se toman como base para este trabajo y la comparación utilizando las mejoras implementadas en la solución propuesta. Finalmente el capítulo 5 contiene las conclusiones obtenidas a partir de los resultados del capítulo 4.

Capítulo 2

Marco teórico

Este capítulo describe los métodos utilizados en la estimación de pose de la mano. En particular se detallan los métodos basados en árboles de regresión e información topológica de la mano como la guía para el entrenamiento de los árboles.

Los árboles de decisión utilizados en tareas de clasificación y regresión son árboles binarios en los cuales las decisiones evaluadas en cada uno de los nodos fueron seleccionadas de un conjunto de opciones generadas en forma aleatoria, como es detallado en este capítulo.

Los conceptos requeridos para el funcionamiento de los bosques de decisión aleatorios, la generación de un modelo de mano con inferencia de información usando nodos latentes y los métodos alternativos utilizados como base para el entrenamiento y predicción en la solución propuesta son presentados.

2.1 Árboles de decisión aleatorios

La estructura de datos conocida como árbol de decisión consiste en un conjunto de nodos conectados de forma jerárquica en la cual no se presentan ciclos, tal como se muestra en la figura 2.1. En cada uno de los nodos internos de este árbol se evalúan funciones que determinan lógicamente la secuencia de nodos que debe recorrer un determinado dato al atravesar el árbol.

Los árboles de decisión utilizados en algoritmos de aprendizaje automatizado se dividen en árboles de clasificación y árboles de regresión [6]:

- **Árboles de clasificación**: tienen la finalidad de evaluar un conjunto de datos y asignarlos a una de varias categorías previamente establecidas, luego de evaluar las condiciones internas dentro de los nodos del árbol.
- **Árboles de regresión**: a diferencia de los árboles de clasificación, los árboles de regresión calculan un valor numérico en un rango continuo, que se calcula a partir de datos almacenados en los nodos del árbol.

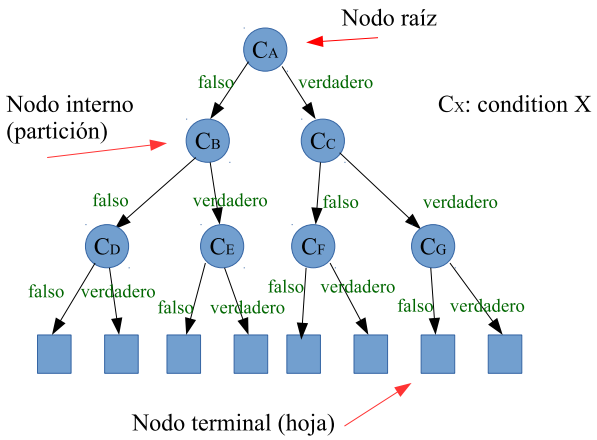


Figura 2.1: Esquema de un árbol de decisión genérico

Un árbol de decisión aleatorio es creado con la finalidad de asignar a un dato de entrada (un píxel, en el caso de procesamiento de imágenes), a un resultado particular basado en decisiones en cada uno de los nodos que lo componen.

El entrenamiento de un árbol de este tipo asigna a cada pixel x evaluado en un conjunto de imágenes S, la probabilidad de pertenencia a una clase c_i. Para un árbol específico t, esta probabilidad se denota con $P_t(c_i|S, x)$.

Un bosque aleatorio está compuesto por múltiples árboles de decisión entrenados de forma independiente, cada uno de ellos con distintas evaluaciones en cada nodo y como resultado, la probabilidad de pertenencia a una determinada clase $P_t(c_i|S, x)$ para el píxel i es diferente en cada árbol.

La probabilidad conjunta del bosque para dicho píxel x puede ser calculada como una media de los resultados de los diferentes árboles generados

$$P(c_i|I, x) = \frac{1}{N} \sum_{n=1}^{N} P_n(c_i|I, x) \tag{2.1}$$

El sistema tiene un resultado estadísticamente válido que se obtiene al calcular las distribuciones probabilísticas para cada uno de los nodos con cada imagen en el subconjunto de entrenamiento que le corresponde, utilizando para ello un conjunto de características creadas aleatoriamente. Todas las características son evaluadas en el nodo hasta determinar la que hace una partición óptima del conjunto de datos de entrenamiento en el nodo con respecto al criterio de ganancia de información, en este caso maximizando dicha ganancia. El proceso descrito es repetido recursivamente en los nodos hijos hasta alcanzar los nodos hoja del árbol, lo cual hace este entrenamiento computacionalmente costoso.

En [12] se hace la comparacion del uso de árboles aleatorios de clasificacion y regresión para la detección de pose de cuerpo completo. Para el caso del árbol de clasificacion se entrena el árbol para asignar etiquetas a los diferentes puntos de una imagen de profundidad correspondientes a diferentes secciones del cuerpo. En el caso del árbol de regresión se

entrena el árbol con los centroides de cada una de las mismas secciones entrenadas para el árbol de clasificación y el resultado de este es la posición de cada uno de estos puntos para una imagen de entrada. En ambos casos la clasificación brinda resultados correctos con una precisión ligeramente mejor obtenida para los árboles de regresión.

El estudio de investigaciones realizado en [27] muestra propuestas recientes de algoritmos de reconocimiento de pose de manos y hace el estudio de una estrategia de estimación con redes neuronales. Como resultado de esta investigación se muestra que las redes neuronales alcanzan mayor precisión en la detección y estimación de pose de manos articuladas en comparación con los árboles de decisión, sin embargo, el tiempo necesario para procesar las imágenes lo hace aún inviable como una alternativa de tiempo real.

En [16] se introduce un nuevo método inspirado por el algoritmo de LTM para guiar el proceso de entrenamiento, pero en lugar de este se hace uso de un algoritmo de puntos indexados de segmentación en nodos internos para decidir cómo se distribuyen los datos en las distintas etapas del entrenamiento mediante el ajuste de una función de optimización. Sin embargo, puesto que no se tiene un algoritmo que indique la relación entre los nodos y la necesaria optimización realizada, el proceso de entrenamiento es computacionalmente más costoso.

Algoritmos basados en aprendizaje profundo, utilizando redes neuronales, han demostrado un desempeño comparable a los árboles de decisón y en algunos casos un mejor desempeño a estos, pero los resultados dependen enteramente del conjunto de datos utilizados para entrenar [27]. Recientemente se ha estado utilizando este tipo de entrenamiento en la mayoría de los trabajos de reconocimiento de pose de manos con propuestas utilizando redes neuronales convolucionales [11], [20], [7], [21]. En todos los casos citados el algoritmo es capaz de ejecutarse en tiempo real utilizando el paralelismo de las múltiples unidades de procesamiento de al menos un GPU con el cual obtiene rendimiento similar en velocidad de predicción a los métodos con árboles de regresión que utilizan únicamente CPU.

La menor complejidad computacional requerida durante la predicción de imágenes es la razón por la cual el presente trabajo continúa con la investigación de los métodos basados en árboles de regresión.

2.2 Modelo de árbol latente

El modelo de árbol latente (Latent Tree Model o LTM por sus siglas en inglés) es una estructura de datos de tipo árbol en la cual se agregan nodos internos (latentes) con la finalidad de hacer inferencias en los datos de entrada para ayudar a encontrar similitudes entre las muestras consideradas.

En el caso de la mano, se trabaja con un conjunto de datos estructurado con la posición en tres dimensiones de cada una de las articulaciones. La característica de interés en el agrupamiento de los nodos de la mano es la distancia entre articulaciones, con la cual se aprende la jerarquía de estas empezando por el punto de entrada central de la mano que

corresponde al centroide de la imagen de profundidad y terminando en cada una de las articulaciones como nodos hoja.

Así, el LTM es es un modelo de árbol gráfico estructurado que contiene un conjunto de nodos observados O que corresponden a las articulaciones reales y un conjunto de nodos latentes L con información inferida a partir de los datos de entrada para construir la jerarquía.

2.3 Puntos indexados de segmentación

Una implementación alternativa en la cual se ha evitado hacer uso del entrenamiento guiado de un árbol de regresión se presenta en [16] en el cual se utiliza el concepto de puntos indexados de segmentación (SIP por las siglas en inglés para Segmentation Index Points) para la creación de los árboles de regresión.

Este método no utiliza el esquema entrenado de LTM para guiar el proceso de entrenamiento, sino que cálculos realizados durante el proceso de generación de los árboles latentes de regresión son los que permiten mejorar el proceso de entrenamiento. La decisión de cómo se deben partir los subconjuntos de datos en cada uno de los nodos internos del árbol se realiza por medio del uso de un algoritmo de optimización de dos etapas con el cual se decide las mejores características a usar para la partición de datos.

Capítulo 3

Sistema de estimación de pose de manos

El sistema de estimación de pose de manos implementado en este trabajo sigue el modelo presentado previamente en la figura 1.3. El proceso de estimación incluye las fases de entrenamiento y predicción. En el entrenamiento se deben aprender características de un conjunto de datos proporcionado utilizando LTM para aprender la información de las relaciones del conjunto de articulaciones que conforman la mano y árboles de decisión para entrenar las diferentes variaciones que puede haber en el conjunto de datos. En el proceso de predicción se toman los datos generados durante el entrenamiento y se utiliza esta información para estimar la pose de una imagen de entrada.

En este capítulo se incluye en detalle la información de los algoritmos utilizados para los procesos de entrenamiento y predicción descritos.

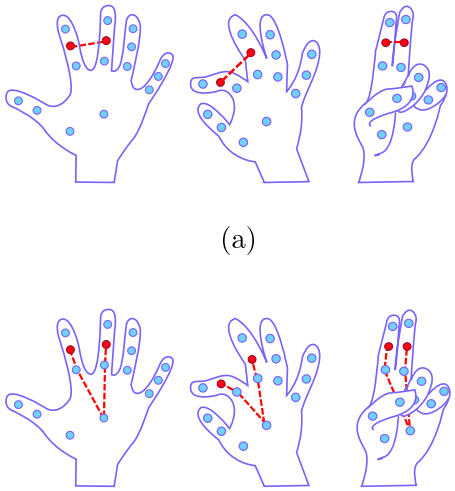
3.1 Modelo de árbol latente

El modelo utilizado para guiar el proceso de entrenamiento en la solución implementada utiliza los árboles latentes de regresión. La característica utilizada para agrupar los datos en el árbol con nodos latentes está dada por

$$D_{xy} = \frac{\sum_{I \in S} \delta(I, x, y)}{|S|} \tag{3.1}$$

donde δ representa la distancia geodésica entre las articulaciones x e y para una imagen I. El modelo de la mano utilizado considera cada uno de los nodos correspondientes a las articulaciones como las variables observadas. La distancia entre estas articulaciones puede variar drásticamente si se usa la distancia euclidiana al cambiar entre diferentes poses de la mano, pero esto no sucede en el caso de la distancia geodésica, pues esta considera la anatomía de la mano [28]. La distancia geodésica entre dos articulaciones se

calcula utilizando trayectorias que pasan solamente por puntos válidos de la mano en la imagen de profundidad. Con esto es posible calcular la distancia más cercana entre dos articulaciones considerando una trayectoria que debe pasar por el nodo raíz, en lugar de considerar la distancia absoluta entre nodos, tal como es ilustrado en la figura 3.1, donde se resaltan las distancias que deben tenerse en cuenta en el caso presentado.



(a)

(b)

Figura 3.1: Comparación de distancias en articulaciones a) Distancia euclidiana entre articulaciones b) Distancia geodésica. Tomado de [28]

El modelo latente de la mano es calculado con la finalidad de obtener un modelo de la dependencia espacial entre articulaciones que sea más representativo de la relación real entre estos. El primer paso en la formación del árbol latente consiste en construir un grafo que enlaza los nodos observados entre sí de acuerdo con las distancias entre cada uno [3], con lo cual se genera una matriz de $N \times N$, donde N representa la cantidad de articulaciones consideradas en el modelo simplificado de mano utilizado.

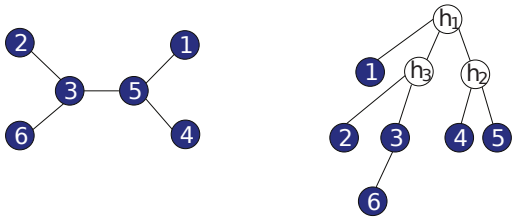
La formación del árbol latente ubica puntos observables en un grafo no dirigido utilizando el algoritmo de Chow-Liu [4] para establecer las conexiones entre diferentes nodos. Inicialmente se crea un grafo completamente conectado y utilizando la información de las distancias geodésicas entre los distintos nodos del grafo, se decide cuáles deben ser descartadas debido a discontinuidad para conservar entonces las conexiones de interés. La matriz de distancia está dada por

$$\mathbf{D} = \begin{pmatrix} 0 & d_{12} & \dots & d_{1N} \\ d_{21} & 0 & \dots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \dots & 0 \end{pmatrix} \tag{3.2}$$

donde la diagonal es siempre cero porque representa la distancia de cada punto a sí mismo.

A partir de este grafo se aplica el algoritmo de agrupación de nodos vecinos CLNJ (Chow-Liu Neighbor Joining por sus siglas en inglés) para agregar nodos internos [3]. Este

algoritmo es un proceso recursivo de agrupamiento de nodos por características similares. En el caso de las manos se considera la distancia entre ellos, hasta generar una jerarquía de nodos donde las variables observadas se ubican como nodos terminales y los nodos internos infieren información de los nodos hijos correspondientes, tal como lo ilustra la figura 3.2.



a) b)

Figura 3.2: Comparación gráfica de algoritmos de Chow-Liu y CLNJ a) Grafo generado usando el algoritmo de Chow-Liu b) Árbol generado luego de aplicar CLNJ. Tomado de [3].

La estructura del árbol generado constituye un árbol binario, es decir, cada uno de los nodos internos puede estar relacionado con no más de tres nodos para mantener la estructura del modelo de la mano y las variables observadas son asignadas estrictamente a los nodos hoja en la estructura. La figura 3.3 muestra cómo cada uno de los nodos observados es representado como una hoja y los nodos internos son nodos latentes que agrupan subconjuntos de articulaciones de la mano utilizando distancia geodésica.

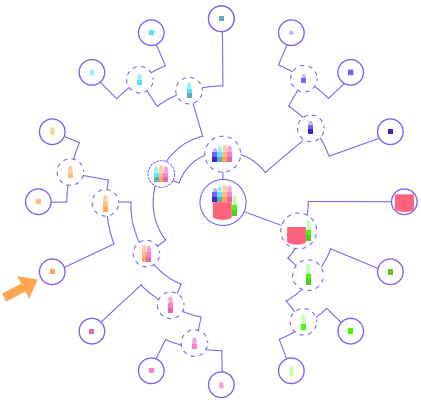


Figura 3.3: Esquema teórico de nodos de la mano generado con nodos latentes. Tomado de [28]

3.2 Estimación de pose con árboles de regresión y LTM

El entrenamiento de árboles de regresión basados en la estructura jerárquica de la mano es un método con el cual se busca hacer más eficiente el proceso de entrenamiento al evitar

la evaluación de todos los puntos de la imagen en cada nodo de árbol como en el caso de los bosques de árboles aleatorios (Random Forests). Con esto se reduce la complejidad del proceso de división del conjunto de datos conforme se avanza hacia abajo en el árbol [28].

3.2.1 Entrenamiento

Este tipo de árbol LRT está conformado de tres tipos de nodos: nodos de partición, nodos de división y nodos hoja. Dado un conjunto de datos de entrenamiento S, los nodos de partición tienen la función de guiar el proceso de entrenamiento para un nodo particular del LTM mediante la evaluación de una función de prueba y separar los datos en dos nuevos subconjuntos: uno para el hijo de la derecha y otro para el hijo de la izquierda, y así sucesivamente hasta alcanzar un criterio de parada. En este punto se introduce un nodo de división que corresponde al siguiente nodo del LTM, se aumenta la cantidad de datos y se genera una nueva etapa de nodos de partición hasta llegar al siguiente nodo de división. El proceso continúa hasta que los nodos hoja son introducidos cuando el proceso de partición acaba en un nodo que solamente contiene información de una articulación.

Dada una imagen de profundidad I, su respectivo conjunto de articulaciones como datos etiquetados y el modelo de árbol latente (LTM) previamente entrenado según la topología de la mano representada por M, donde para cada nodo i en M, $i = 1, \ldots, |M|$, su padre está definido como $p(i)$ y sus nodos hijos como $l(i)$ y $r(i)$. En cada imagen I utilizada en el entrenamiento una posición tridimensional ρ_i^I está asociada con el nodo i del LTM, de forma que para los nodos observables $\rho_i^I, i \in O$ corresponde a la posición real de la articulación y para los nodos latentes $\rho_i^I, i \in L$ corresponde al centroide de las posiciones de los nodos observables de los que se compone.

El proceso de entrenamiento para un árbol latente se hace en etapas, considerando cada uno de los nodos del LTM para guiar el proceso de entrenamiento con cada una de las imágenes de entrada representado en el conjunto (I, ρ_i^I). Inicialmente se toma el nodo $i = 0$ en M para separar el conjunto de datos correspondientes a $l(i)$ y $r(i)$ mediante la evaluación de una función de prueba. En el entrenamiento se genera un conjunto de pares aleatorios $\Phi = \{(f_i, \tau_i)\}$ conformados por una característica f_i y un umbral τ_i. Estos permiten dividir el conjunto de entrenamiento S en dos subconjuntos $S_l = \{I | f_i(I) < \tau_i\}$ y $S_r = \{I | f_i(I) \geq \tau_i\}$. La función de decisión está dada por:

$$f_i(I) = d_I \left(\rho_i^I + \frac{\overrightarrow{u}}{d_I(\rho_0^I)} \right) - d_I \left(\rho_i^I + \frac{\overrightarrow{v}}{d_I(\rho_0^I)} \right) \tag{3.3}$$

donde u y v son vectores aleatorios de desplazamiento con respecto a la posición de ρ_i^I y d_I es el valor de la profundidad en la imagen I. Estos valores de desplazamiento son divididos para cada uno de los puntos evaluados por el valor de profundidad del centroide de la mano ρ_0^I con la finalidad hacer este valor invariante ante el valor de profundidad en distintas imágenes.

La evaluación de las características calculadas aleatoriamente sucede utilizando en cada nodo i una función de ganancia de información dada por

$$IG_i(S) = \sum_{m}^{l(i),r(i)} tr(\sum_{S}^{im}) - \sum_{k}^{l,r} \frac{S^k}{|S|} \left(\sum_{m}^{l(i),r(i)} tr(\sum_{im}^{S^k}) \right) \tag{3.4}$$

Esta ecuación se evalúa en el nodo i para cada una de las imáganes del conjunto de entrenamiento con cada uno de los pares calculados aleatoriamente con la finalidad de encontrar el Φ_i que maximiza el valor de la ganancia de información, esto es, el que produce una mayor separación espacial de los datos de entrenamiento en dos conjuntos.

Esta ganancia de información se calcula como la diferencia de la traza $tr(\cdot)$ de la matriz de covarianza del conjunto completo de vectores de desplazamiento $\{(\rho_m^I - \rho_i^I | I \in X)\}$ usados en el nodo i menos la suma de las trazas de las matrices de covarianza de cada uno de los conjuntos S^l y S^r.

En cada uno de los nodos i del LTM se repite este proceso agregando capas de nodos de partición que continúan separando el conjunto de datos hasta que la ganancia de información sea menor a un umbral determinado. En ese momento se crea un nodo de división en el cual se avanza en el LTM hacia siguiente nivel considerando los hijos $l(i)$ y $r(i)$ y se continúa nuevamente el proceso de división hasta alcanzar los nodos hoja.

En cada uno de los nodos i del LTM se repite este proceso agregando capas de nodos de partición que continúan separando el conjunto de datos hasta alcanzar una cantidad máxima de niveles de nodos de partición. En ese momento se crea un nodo de división en el cual se avanza en el LTM hacia el siguiente nivel considerando los hijos $l(i)$ y $r(i)$ y se continúa nuevamente el proceso de división hasta alcanzar los nodos hoja. En el caso en que los datos no sean suficientes para alcanzar el máximo de nodos de partición, un nodo hoja o un nodo de división es insertado para la siguiente etapa del algoritmo.

Para cada una de las imágenes se calculan los desplazamientos del nodo ρ_i^I usando la característica Φ_i que maximiza la ganancia de información. Los nodos de división guardan los vectores de desplazamiento $\theta_m = \rho_m^I - \rho_i^I$ hacia cada uno de los nodos hijos usando la posición calculada con información del LTM, donde $m \in \{l(i), r(i)\}$. En el caso de los nodos hoja, la información de interés que debe guardarse en cada nodo es el desplazamiento con respecto al nodo padre $\left(\rho_i^I - \rho_{p(i)}^I\right)$ para cada imagen.

La información obtenida al crear un LTM de la mano es utilizada para optimizar el proceso de entrenamiento de un conjunto de árboles de regresión, que son los que van a predecir la posición de las articulaciones para una imagen de mano dada.

El proceso de entrenamiento se describe con detalle en el algoritmo mostrado en la figura 3.4. Este consiste en la toma de puntos de manos de imágenes generadas en las cuales cada una de las articulaciones se encuentra debidamente etiquetada con su posición respectiva. En este trabajo se realizaron distintas particiones de datos para evaluar su efecto en los resultados finales. En una de las estrategias de partición, el conjunto de puntos S de cada

una de las imágenes se distribuye en forma aleatoria en M grupos correspondientes a la cantidad de nodos latentes en el LTM de forma tal que $S = S_0 \cup S_1 \cup ... \cup S_M$, con la finalidad de agregar información de forma gradual al proceso de entrenamiento. Se evalúa el efecto de hacer también una partición de datos en bloques de distinto tamaño utilizando bloques de tamaño creciente determinado por la profundidad del LTM. En este caso el tamaño de los bloques se duplica en cada etapa proporcionando una mayor cantidad de datos a las etapas más profundas, asegurando que dichos datos son consistentes con el proceso de predicción. Combinaciones de ambos métodos se analizan en este trabajo, activando o desactivando bloques crecientes o predicción durante el entrenamiento.

El entrenamiento produce un conjunto de árboles de decisión binarios en los que las variables de decisión en cada uno de los nodos es elegida en forma aleatoria. Estos árboles son entrenados en M etapas correspondientes a cada uno de los nodos internos del LTM. En este proceso inicialmente se entrena para el nodo raíz usando el conjunto de datos S_0, para una segunda etapa del proceso el nuevo conjunto de datos es $S_0 \cup S_1$ y así sucesivamente, conforme se necesita entrenar una nueva etapa cada vez más cercana a las hojas se agrega un nuevo conjunto de datos, de forma que los nodos con mayor rango de movilidad sean entrenados con mayor cantidad de información. Estos árboles son utilizados para determinar la posición de las articulaciones para una imagen de mano dada. Una vez entrenado el árbol, al analizar una imagen en cada nodo hoja se determina a cual de las articulaciones corresponde un determinado conjunto de puntos de forma similar a como se hace para los RDF, utilizando (2.1). Tomando la información estimada para dicha articulación en cada uno de los árboles se determina la posición global correspondiente para cada uno de las articulaciones en el espacio tridimensional.

3.2.2 Predicción

El proceso de predicción para una imagen I inicia con la información de la posición del nodo raíz ρ_0^I y avanza por las condiciones entrenadas para un determinado árbol LRT utilizando las características guardadas en cada uno de los nodos de partición hasta alcanzar un nodo de división. En el nodo de división se acumulan votos calculados con

$$d_k = \rho_i^I + \frac{\theta_j}{\rho_0^I} \mid \theta_j \in \theta_{k=l,r} \tag{3.5}$$

en dos espacios de Hough H^l y H^r, donde θ_j representa los desplazamientos guardados en el nodo de división. La moda calculada para cada uno de estos espacios determina las nuevas posiciones $\rho_{l(i)}^I$ y $\rho_{r(i)}^I$ de los nodos hijos a partir de los cuales iniciará de nuevo el proceso hasta que la predicción acaba en un nodo hoja distinto para cada articulación con la posición estimada con la información de dicho árbol. La predicción es realizada para cada uno de los árboles generados durante el entrenamiento y el resultado final es calculado estadísticamente utilizando (2.1) como en el caso de un bosque de árboles aleatorios.

Entrada: Un conjunto de entrenamiento S
Entrada: LTM preaprendido donde $M = (O \cup L, E)$ con profundidad D
Salida: Un árbol *LRT* T
LRT (S, M)
  Dividir el conjunto S en subconjuntos aleatorios $S_0, \ldots, S_D$
  $i \leftarrow 0, j \leftarrow 0$ ▶ Nodo i del LTM y del nodo j del LRT
  d $\leftarrow 0$ ▶ Primera etapa de entrenamiento

  LrtSplit $(i, r(j), S_0, d)$

LrtSplit (i, j, S, d)
  Generar un set de aleatorio de características Φ
  para todo ϕ *in* Φ **hacer**
    Particionar S en S^l y S^r usando ϕ obtenido con (3.3)
  fin
  Usar la entropía en (3.4) para obtener ϕ óptimo
  si $IG_i(S)$ *es suficiente / máximo nivel de nodos de partición* **entonces**
    Guardar j como nodo de partición en T
    LrtSplit $(i, l(j), S^l, d)$
    LrtSplit $(i, r(j), S^r, d)$

  si no, si $i \in L$ **entonces**
    Guardar j como nodo de división en T $S \leftarrow S \cup S_{d+1}$
    LrtSplit $(l(i), l(j), S, d+1)$
    LrtSplit $(r(i), r(j), S, d+1)$

  en otro caso
    Guardar j como nodo hoja

  fin
  devolver

Figura 3.4: Algoritmo para generación de un LRT

Capítulo 4

Resultados y análisis

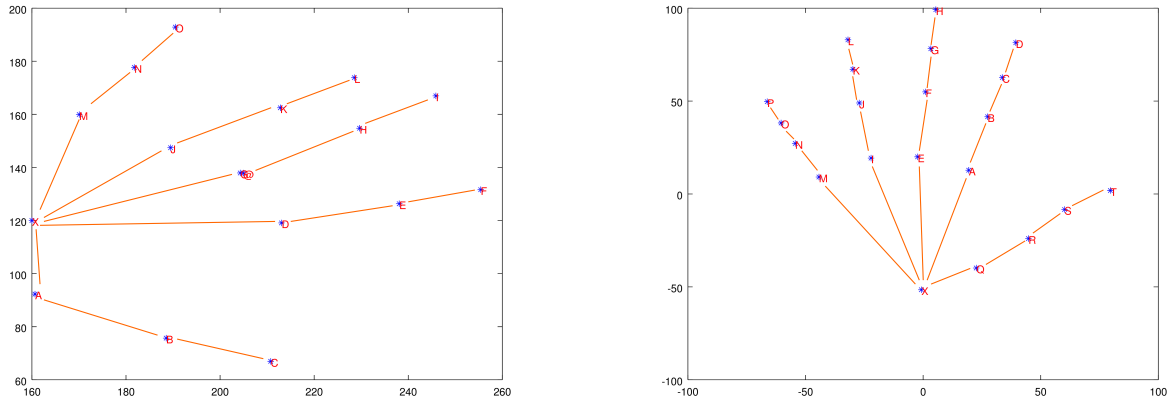
En este capítulo se presentan los resultados de distintas etapas de la implementación realizada del algoritmo descrito en este trabajo así como el análisis respectivo de los datos obtenidos. La primera etapa del algoritmo consiste en el entrenamiento del árbol de nodos latentes (LTM), el cual es utilizado para guiar el proceso de entrenamiento de los diferentes árboles de regresión.

4.1 Latent Tree Model

El proceso de entrenamiento realizado se ha llevado a cabo utilizando un conjunto de imágenes con 21 puntos [23] que representan las articulaciones con mayor información. El diagrama de los puntos considerados para una mano extendida se muestra en la figura 4.1 b), donde cada nodo es identificado con un caracter como referencia para una articulación específica en las otras imágenes de diagramas de LTM en este capítulo. El conjunto de imágenes original [28] para este algoritmo utiliza un conjunto de datos similar pero con 16 articulaciones tal como se muestra en 4.1 a).

Inicialmente la información de todos los puntos debe relacionarse entre sí utilizando un grafo completamente conectado. A partir de este grafo deben eliminarse todas las conexiones entre nodos entre los cuales existe un cambio abrupto de profundidad, lo cual indica que no existe continuidad en la superficie de la mano considerada. El proceso de entrenamiento del LTM utiliza múltiples imágenes en distintas configuraciones y diferentes rotaciones con lo cual el grafo resultante elimina discontinuidades considerando todas las imágenes disponibles.

El algoritmo de Chow-Liu es utilizado posteriormente sobre el grafo generado con la finalidad de definir la jerarquía de la estructura de nodos conectados basado en la matriz de distancias entre los diferentes nodos. El resultado de este algoritmo es el gráfico de la figura 4.2a) en el cual se muestra la estructura sintética de nodos conectados generada a partir del esquema de la mano mostrado en la figura 1.2, considerando la posición de la mano extendida. Esta estructura sintética corresponde a conectar cada uno de los nodos



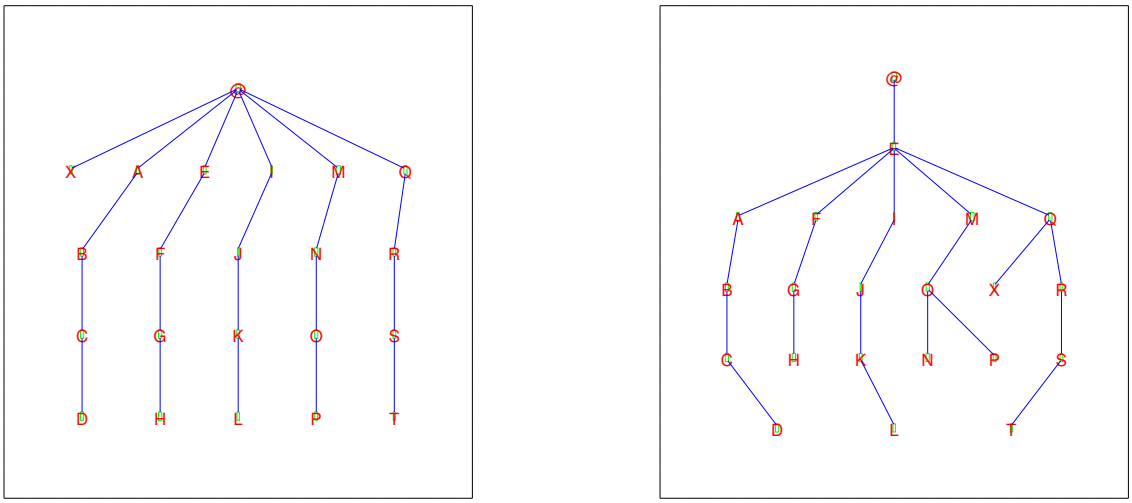
a) b)

Figura 4.1: Posición espacial de nodos de articulaciones de la mano. a) Modelo de mano con 16 articulaciones, b) modelo de 21 articulaciones

de la mano según el esquema, donde existe un punto central correspondiente a la palma de la mano y a este nodo se conecta el nodo más cercano de cada uno de los dedos. Al considerar un conjunto de imágenes con su profundidad respectiva, el resultado de generar un grafo con este algoritmo se muestra en la figura 4.2b), utilizando un conjunto de 8500 imágenes para determinar cuáles son los nodos más cercanos entre sí para el caso de 21 articulaciones. Este resultado muestra cómo el algoritmo puede aprender una estructura de la mano distinta a la estructura sintética esperada según la jerarquía de nodos de la mano, tal como se observa en la figura 4.2b), donde el nodo E es promovido a nodo raíz y otros nodos como en el caso de N y X son asignados como hijos de otros nodos, lo cual se debe a que las distancias entre dos nodos no directamente conectados en el grafo sintético pueden ser menores a la distancia entre nodos directamente conectados.

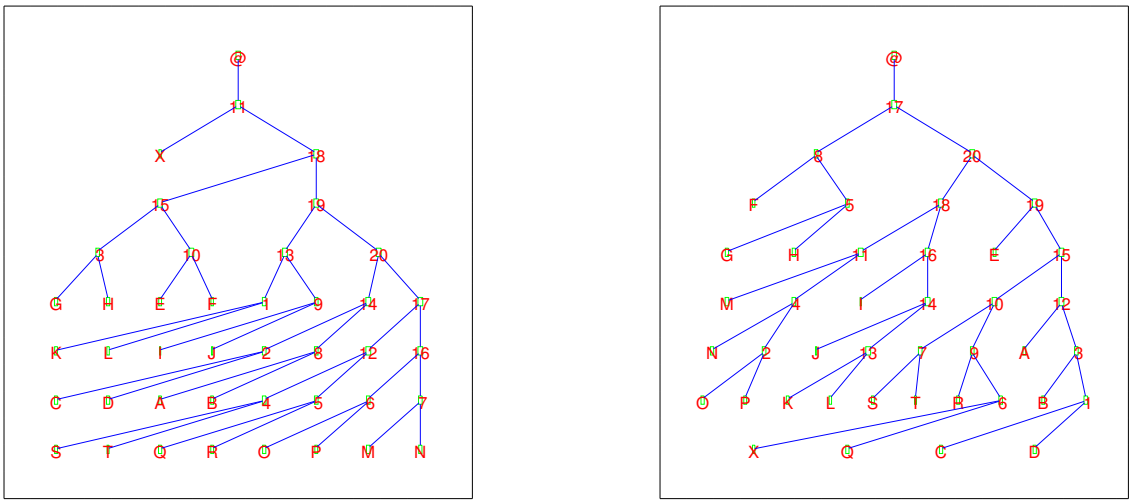
La generación de un árbol conectado con información inferida en nodos latentes se hace utilizando el algoritmo CLNJ (Chow-Liu Neighbor Joining) que hace uso del algoritmo de Chow-Liu para encontrar los nodos más cercanos entre sí y del algoritmo de NJ (Neighbor Joining), cuyo diagrama se muestra en la figura 4.3. En esta figura se muestra la relación entre nodos de la mano obtenidos con el algoritmo de NJ y los nodos latentes introducidos, identificados con caracteres numéricos. El resultado de calcular las relaciones utilizando el algoritmo de CLNJ se muestra en la figura 4.4, el cual es el grafo utilizado como punto de partida para el entranamiento de los árboles de regresión tal como se propone en [28].

En el caso de los árboles obtenidos con cada uno de los algoritmos, se nota que para el caso del algoritmo NJ los nodos que se encuentran fuera de lugar con respecto al diagrama de nodos sintético son los nodos X, E y I; en el caso del algoritmo CLNJ los nodos que discrepan con el esquema sintético X, E y Q, por lo tanto el comportamiento en ambos casos es similar y cualquiera de los dos árboles podría ser considerado como una representación válida de la estructura de la mano. Sin embargo, el algoritmo CLNJ se prefiere por tener, en general, mayor exactitud y eficiencia al representar árboles latentes



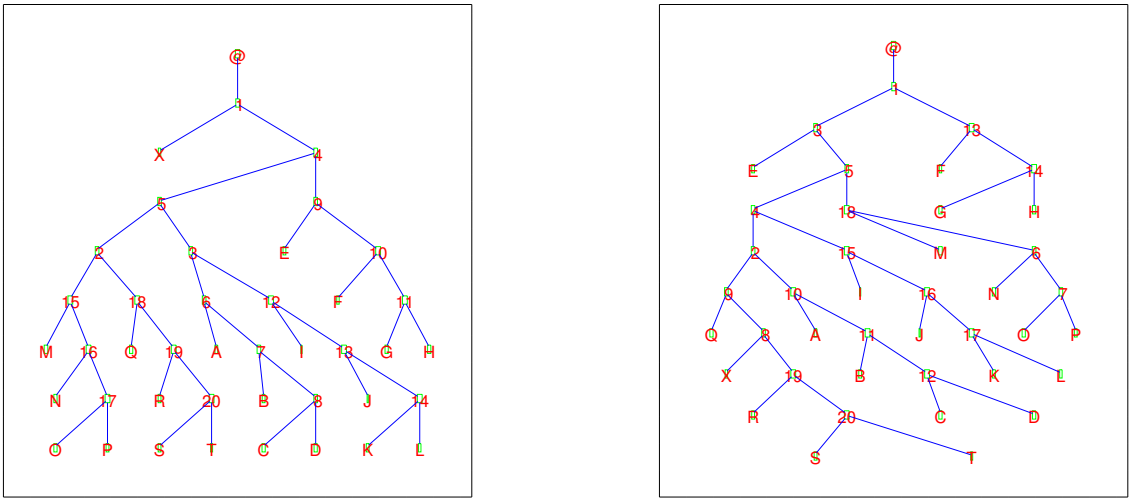
a) b)

Figura 4.2: Esquema nodos generado para la mano con algoritmo de Chow-Liu para LTM de 21 articulaciones. a) Esquema sintético, b) Esquema entrenado



a) b)

Figura 4.3: Esquema nodos generado para la mano con algoritmo NJ para LTM de 21 articulaciones. a) Esquema sintético, b) Esquema entrenado



a) b)

Figura 4.4: Esquema nodos generado para la mano con algoritmo CLNJ para LTM de 21 articulaciones. a) Esquema sintético, b) Esquema entrenado

con varios niveles de nodos latentes [3] y por esto es el utilizado para guiar el entrenamiento de los árboles de regresión en este trabajo.

Este entrenamiento tiene la finalidad de obtener una relación entre nodos de la mano acorde con las imágenes del conjunto de entrenamiento utilizado que puede no ser idéntica al caso sintético, como se evidencia en este caso. Esta diferencia se obtiene al considerar las diferentes distribuciones topológicas encontradas en las imágenes del conjunto de entrenamiento que divergen de la posición de mano extendida considerada para el diagrama sintético.

4.2 Bosque latente de regresión

La evaluación de las características aleatorias en cada uno de los nodos de los árboles de regresión debe producir la mejor partición de los desplazamientos espaciales para un determinado punto por medio del uso de la ganancia de información (3.4). El objetivo de evaluar múltiples conjuntos de datos consiste en escoger el que produce que los desplazamientos que representan la posición del hijo derecho e izquierdo del nodo actual se encuentren en conjuntos con la mayor separación posible espacialmente, al maximizar la ganancia de información.

La figura 4.5 muestra un ejemplo del resultado de la evaluación de un conjunto de desplazamientos correspondiente al conjunto de características con la mayor ganancia de información en el nodo de un árbol de regresión correspondiente al nodo de división para el nodo latente identificado con la etiqueta “2” en el esquema de la figura 4.4b), donde los

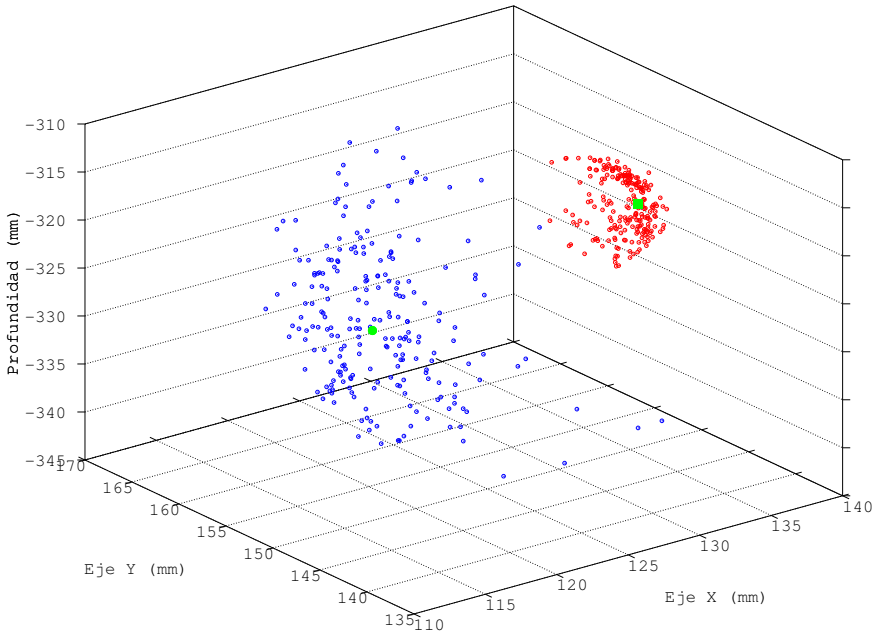


Figura 4.5: Separación espacial de datos basado en ganancia de información

puntos representados en rojo y azul corresponden a los conjuntos de offsets para el hijo izquierdo y derecho, respectivamente. En el caso del nodo considerado se observa que el algoritmo logra efectivamente la separación de dos grupos de datos que son luego propagados en los dos hijos del nodo. La posición de referencia de cada uno de estos conjuntos de datos se representa utilizando la media de las posiciones de los puntos que conforman cada grupo, este centroide se encuentra identificado en la imagen con los puntos verdes.

4.3 Predicción

La ejecución del algoritmo ha sido realizada iniciando por el entrenamiento de subárboles del LTM para evaluar resultados parciales con un conjunto reducido de datos y el esquema de nodos latentes. Las primeras pruebas incluyen el entrenamiento de árboles de regresión para el árbol latente de un único dedo y luego para 2 dedos. El entrenamiento se ha realizado de esta forma para validación parcial del algoritmo, y finalmente se incluyen resultados del entrenamiento con LTM completo.

4.3.1 Predicción utilizando múltiples árboles

El entrenamiento de los árboles de regresión en el algoritmo original [28] hace uso de múltiples árboles con los cuales se pueda calcular luego un promedio de las posiciones de las diferentres articulaciones tomando en cuenta todos los árboles entrenados. En las

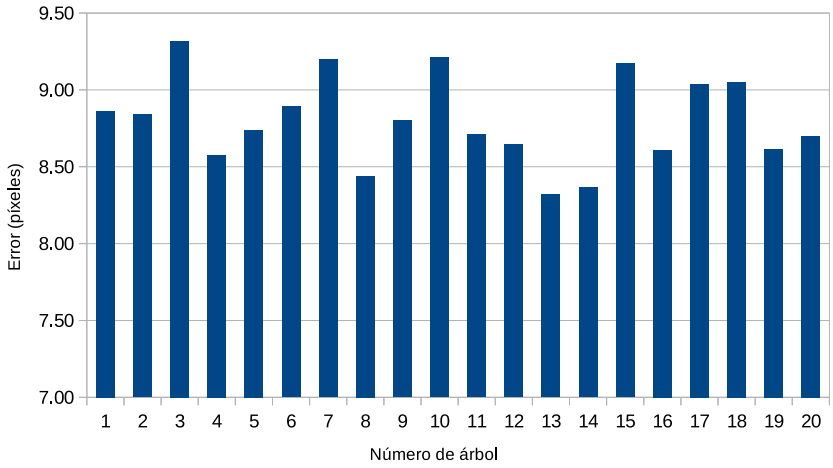


Figura 4.6: Error de predicción de articulaciones para 1 dedo utilizando 20 árboles y entrenamiento con profundidad de 1

pruebas para 1 y 2 dedos de esta sección se utiliza un máximo de 20 árboles, basado en que la cantidad propuesta en el algoritmo original es de 16 para la mayoría de las pruebas realizadas.

En esta prueba se hace una evaluación del error para cada uno de los árboles entrenados y los resultados se muestran en la figura 4.6, estos datos corresponden al error total de predicción para un dedo con 4 articulaciones evaluado en un conjunto de prueba de 50 imágenes, para cada árbol individual. El error mostrado en el gráfico corresponde al error total para el conjunto de imágenes,

Al tomar en cuenta los diferentes árboles se logra disminuir el error en la estimación de la posición final de cada una de las articulaciones. El resultado de promediar posiciones de diferentes árboles se muestra en la figura 4.7. Cada uno de los resultados en esta imagen corresponde a hacer el promedio en función de la cantidad de árboles y en el gráfico se evidencia que el error efectivamente disminuye con el número de árboles en comparación con el error de cada árbol individual. La prueba realizada se realizó utilizando un conjunto de entrenamiento de 2000 imágenes y un conjunto de prueba de 50 imágenes; 25 vectores de características y 25 umbrales distintos.

4.3.2 Predicción utilizando subárboles

El algoritmo es evaluado inicialmente utilizando solamente el subárbol del LTM que contiene entre sus hijos los nodos latentes y nodos hoja correspondientes a un único dedo, con la finalidad de evaluar el comportamiento en un escenario con reducida cantidad de información. El resultado gráfico de esta prueba se muestra en la figura 4.8, para la cual se ha calculado el error para un conjunto de 50 imágenes para 20 árboles con profundidad de 2 niveles de nodos de partición.

Con la misma finalidad de hacer una evaluación de un subárbol esta vez de mayor tamaño

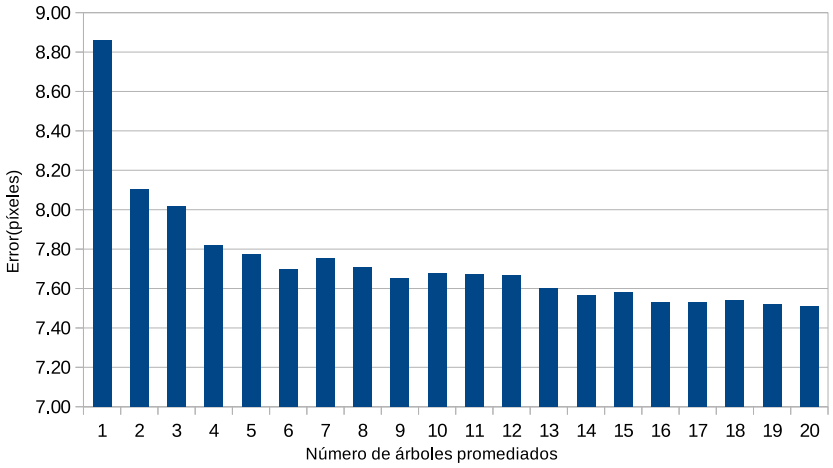


Figura 4.7: Reducción del error utilizando múltiples árboles para 1 dedo utilizando 20 árboles y entrenamiento con profundidad de 1. El error mostrado está en función de la cantidad n de árboles considerados y corresponde al error de la posición obtenida como promedio de las predicciones de n árboles

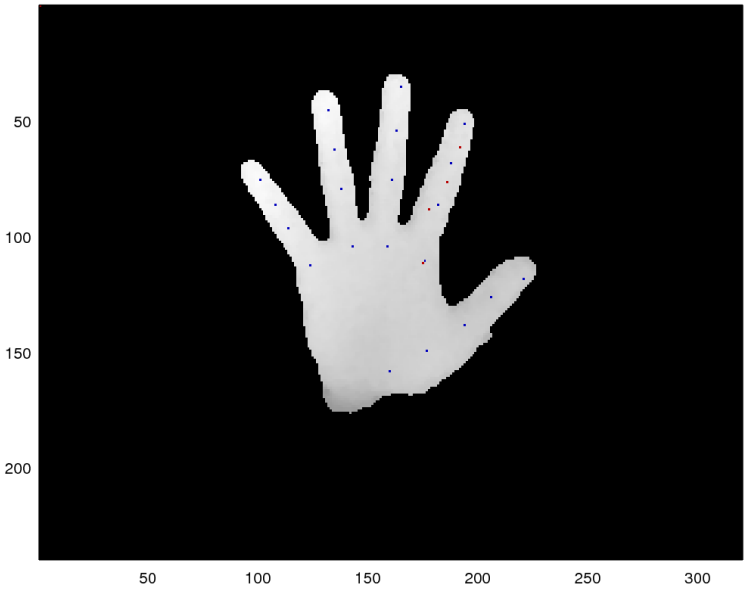


Figura 4.8: Predicción de articulaciones para 1 dedo utilizando 20 árboles y entrenamiento con profundidad de 2

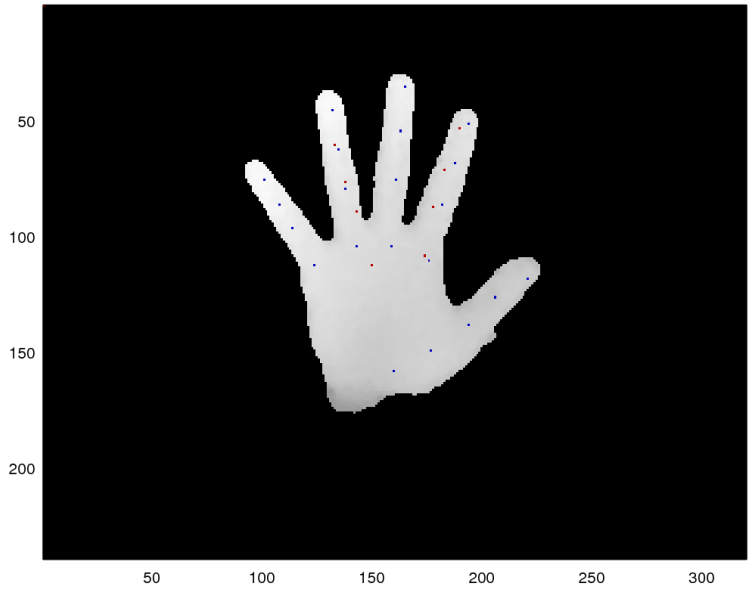


Figura 4.9: Predicción de articulaciones para 2 dedos utilizando 20 árboles y entrenamiento con profundidad de 1

se hace el entrenamiento a partir de un LTM con el esquema para 2 dedos. El resultado se muestra en la figura 4.8, para la cual se ha logrado obtener el error para un conjunto de 50 imágenes de prueba utilizando 20 árboles con profundidad de 2 niveles de nodos de partición. Esto equivale a un error promedio de 10.395 píxeles por articulación en esta configuración.

4.3.3 Imágenes necesarias

En esta sección se analiza la cantidad de imágenes necesaria para realizar el entrenamiento realizando la evaluación de las imágenes desde la raíz del árbol. Una mejora que se propone realizar en este trabajo consiste en realizar el entrenamiento de las diferentes etapas del árbol de regresión, y justo antes de insertar un nuevo nodo de división, no agregar un bloque completo de imágenes como punto de partida para la nueva etapa, sino propagar a esa etapa solamente imágenes que, al evaluarlas desde el nodo raíz, puedan haberse propagado hasta el nodo actual. Este enfoque tiene el objetivo de hacer que la cantidad de imágenes que se propaguen a cada nodo en el árbol hayan cumplido con las evaluaciones de características para cada nodo desde la raíz y evitar que se hagan decisiones con imágenes agregadas para evaluar una etapa sin tomar en cuenta que algunas no hubieran alcanzado los nodos de esa etapa al evaluar las características en todos los nodos superiores en el mismo árbol. El problema encontrado con este cambio está en la cantidad de imágenes necesarias para poder entrenar correctamente el árbol completo como será discutido para el caso de un dedo y para mano completa.

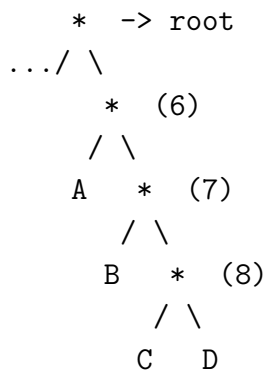


Figura 4.10: Diagrama de profundidad máxima para árbol completo

Profundidad n	2^{4+3n}	Imágenes necesarias
0	2^4	16
1	2^{4+3}	128
2	2^{4+6}	1024
3	2^{4+9}	8192
4	2^{4+12}	65536

Tabla 4.1: Mínima cantidad de imágenes distintas requeridas para entrenar un dedo según la cantidad de nodos de partición

Caso de un dedo

En caso de un dedo, la estructura del árbol utilizado para el entrenamiento tiene una profundidad máxima de 4 nodos de división tal como se muestra en la figura 4.11 para ningún nivel de nodos de partición. Conforme se aumenta la cantidad de nodos de partición entre dos nodos de división, la cantidad de imágenes necesarias para realizar el entrenamiento crece de acuerdo con la información mostrada en la tabla 4.1. En esta tabla se muestra que para 4 niveles de nodos de partición se hacen necesarias 65536 imágenes para el mejor de los casos, en el cual una imagen acabaría en cada uno de los nodos hoja del árbol final.

Caso de mano completa

En el caso de entrenar un LTM para mano entera, la profundidad del árbol utilizado es típicamente de 8 nodos, como se muestra en la figura 4.11. La cantidad de imágenes requeridas para distintos niveles de nodos de partición se muestran en la tabla 4.2. Así por ejemplo, para una profundidad de 2 nodos de partición, la cantidad de imágenes necesarias es de 4194304, valor alto si se considera que las bases de datos de las que se parte en este trabajo constan de alrededor de 64000 [23] y alrededor de 336000 [28] para la base de datos del trabajo original del presente algoritmo.

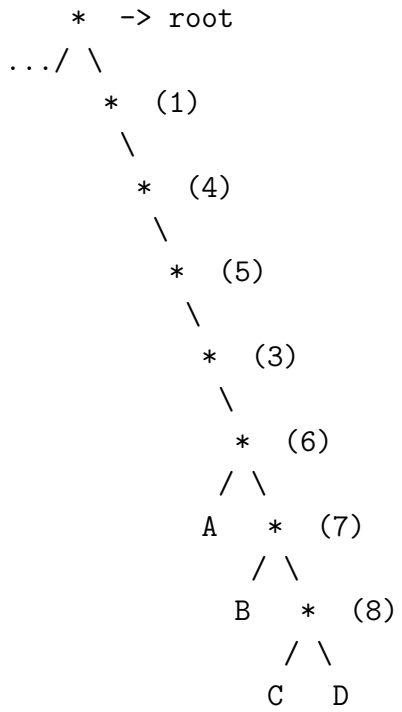


Figura 4.11: Diagrama de profundidad máxima para árbol completo

Profundidad n	2^{8+7n}	Imágenes necesarias
0	2^{8}	256
1	2^{8+7}	32768
2	2^{8+14}	4194304

Tabla 4.2: Mínima cantidad de imágenes distintas requeridas para entrenar un árbol de regresión completo

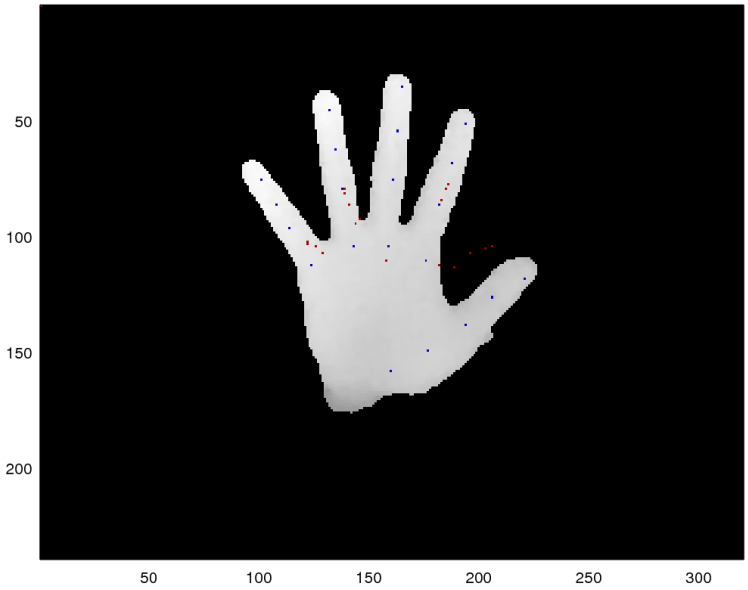


Figura 4.12: Predicción de articulaciones para mano completa usando diagrama de nodos latente sintético

4.3.4 Entrenamiento de mano completa

El resultado de evaluar el algoritmo con mano completa se muestra como referencia para el algoritmo actual. El tiempo de entrenamiento necesario para evaluar el algoritmo hace prohibitiva la evaluación con árboles de regresión con profundidad suficiente para aprender suficiente información para predecir correctamente los nodos de las manos. La figura 4.12 muestra el resultado de la predicción de la posición utilizando para el entrenamiento un conjunto de 500 imágenes de mano completa basado en el diagrama sintético de nodos latente de la figura 4.4a). La figura 4.13 muestra el resultado de evaluar el algoritmo usando el diagrama de nodos latentes entrenado en 4.4b).

En ambos casos la profundidad utilizada es de 0 niveles de nodos de partición. La profundidad utilizada en este caso no es suficiente para permitir al algoritmo predecir las posiciones con mayor precisión.

4.3.5 Tiempo de entrenamiento

El tiempo necesario para realizar los entrenamientos se muestran a continuación. La máquina utilizada durante el proceso de entrenamiento corresponde a un sistema con CPU Intel(R) Xeon(R) CPU E5-4667 v3 @ 2.00GHz con 16 núcleos físicos, 32 procesadores lógicos. El sistema cuenta además con 128 GB de memoria RAM.

Se realizan pruebas de entrenamiento tomando como base el diagrama sintético de uno y dos dedos para tener control de los nodos padre de cada uno de los casos y además se

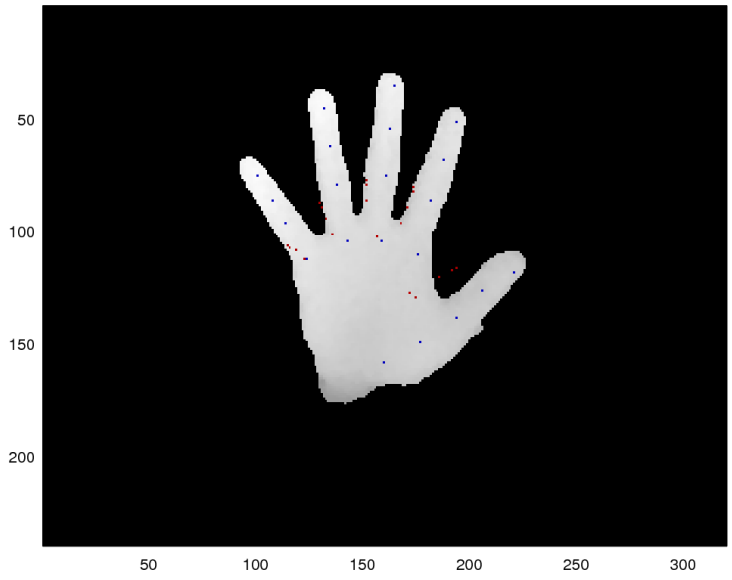


Figura 4.13: Predicción de articulaciones para mano completa usando diagrama de nodos latente entrenado

evalúa el algoritmo utilizando mano completa con el LTM entrenado para comparación con el caso del LTM sintético.

Efecto de la cantidad de características evaluadas en cada nodo

El proceso de entrenamiento en cada uno de los nodos de partición consiste en tomar las características generadas aleatoriamente y evaluarlas contra los umbrales, también generados aleatoriamente para lograr la mejor partición de los datos, maximizando la ganancia de información.

Características	Umbrales	Tiempo (s)	Error (píxeles)
5	5	186.18	14.06
10	10	658.27	14.01
20	20	2455.92	13.94
30	30	4856.61	13.87
40	40	7252.13	13.89
50	50	10797.86	13.69

Tabla 4.3: Tiempo de entrenamiento para distinta cantidad de características a evaluar en los nodos de partición

Con la finalidad de evaluar el efecto de utilizar diferente cantidad de características, se

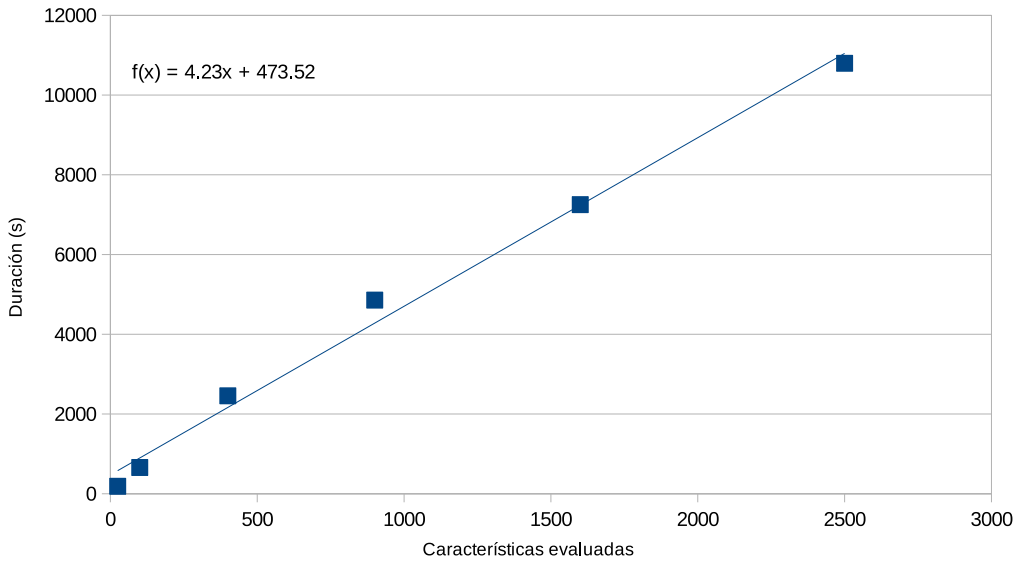


Figura 4.14: Tiempo de entrenamiento con respecto a cantidad de características evaluadas para un dedo.

muestra el resultado de evaluar el algoritmo con cantidad variable de estos. Los resultados de esta prueba se encuentran en la tabla 4.3. En esta prueba se utiliza un subárbol para un dedo, profundidad de nodos de partición de 1, un conjunto de entrenamiento de 500 imágenes y un grupo de 50 imágenes de prueba, solamente con la finalidad de evaluar el tiempo de entrenamiento. Debido a la reducida cantidad de imágenes utilizadas para el entrenamiento, el error en esta prueba es superior al resultado de predicción obtenido mas adelante al evaluar la predicción para un dedo.

La figura 4.14 muestra la representación del tiempo necesario para el entrenamiento en función de la cantidad de características evaluadas, en el cual se muestra que el tiempo necesario tiene un comportamiento lineal al aumentar la cantidad de características a evaluar.

Las pruebas realizadas para este propósito no muestran diferencia significativa en el error de predicción al variar la cantidad de características evaluadas en cada nodo; sin embargo, en el resto de pruebas de esta sección el entrenamiento se realiza utilizando 25 características y 25 umbrales que se combinan para tener un total de 625 combinaciones a evaluar.

Este número fue elegido por el tiempo requerido para entrenamiento aunque el algoritmo original sugiere una combinacion de 2000 características para cada nodo evaluado. Se muestra un ligero decremento en el error por articulación al aumentar la cantidad de características evaluadas, diferencia que debe hacerse más evidente utilizando mayor cantidad de imágenes, lo cual además tendría el efecto de disminuir el error de predicción.

Tiempo de entrenamiento para 1 dedo

Los resultados en la tabla 4.4 muestran el tiempo requerido para el entrenamiento de 20 árboles de regresión utilizando bloques de datos de tamaño creciente con la profundidad, y un entrenamiento con predicción al insertar un nuevo nodo de división en los árboles. Este entrenamiento toma cada imagen y evalúa desde la raíz del árbol para asegurarse que las imágenes llegan a un determinado nodo solamente si ha cumplido con las evaluaciones de los nodos superiores en el árbol. La profundidad corresponde al número de niveles de nodos de partición entre dos nodos de división. Inicialmente se divide la cantidad total de imágenes en bloques del mismo tamaño y se utiliza un bloque distinto en cada etapa del árbol. Los entrenamientos con un dedo utilizan un conjunto de 27000 imágenes en las cuales se presenta la mayor información para el dedo índice.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	9:00.4	8.17
1	22:48.1	7.99
2	1:48:19.1	7.94
3	9:01:10.9	8.11

Tabla 4.4: Tiempo de entrenamiento para 1 dedo con predicción y bloques del mismo tamaño

La misma prueba se realiza luego utilizando bloques de datos de tamaño creciente en cada etapa. Para esta prueba se utilizan los mismos conjuntos de datos de la prueba anterior y la misma cantidad de árboles. Los resultados del promedio de los 20 árboles se muestran en la tabla 4.5. El error es similar al caso anterior, sin embargo, se observa el tiempo de entrenamiento utilizando bloques de tamaño creciente es alrededor de 5 veces el necesario para entrenar utilizando bloques de imágenes del mismo tamaño.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	53:15.4	8.25
1	2:05:15.4	8.01
2	9:43:48.7	7.68
3	2:12:15:02.1	8.35

Tabla 4.5: Tiempo de entrenamiento para 1 dedo con predicción y bloques de tamaño creciente por etapa

En la tabla 4.6 se muestra el tiempo necesario para el entrenamiento de 20 árboles de regresión con bloques de imágenes del mismo tamaño en cada para cada etapa correspondiente a un nodo de división. En este caso se ha suprimido la predicción realizada durante

el entrenamiento antes de cada nuevo nodo de división dejando de lado la evaluación de cada una de las imágenes desde la raíz del árbol. Aunque se puede notar cómo el error disminuye en comparación con el caso anterior, el tiempo de entrenamiento necesario es alrededor de 4 veces mayor en las capas más profundas del árbol al utilizar tamaño fijo para los bloques de datos.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	1:03:07.4	7.99
1	5:19:56.2	7.84
2	1:07:40:21.7	7.51
3	9:15:26:12.7	7.43

Tabla 4.6: Tiempo de entrenamiento para 1 dedo sin predicción y bloques del mismo tamaño

La tabla 4.7 muestra el tiempo necesario para el entrenamiento de 20 árboles de regresión con bloques de imágenes de tamaño creciente pero sin ejecutar predicción durante el proceso de entrenamiento. En esta prueba se utilizan bloques de información que se duplican en tamaño conforme se avanza a una nueva etapa del algoritmo al insertar un nuevo nodo de división, sin embargo, debido a las limitantes de cantidad de imágenes posibles para este caso, solo es posible entrenar los primeros niveles con una cantidad de imágenes reducida.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	1:28:55.9	7.91
1	9:22:13.7	7.81
2	2:16:11:20.0	7.62

Tabla 4.7: Tiempo de entrenamiento para 1 dedo sin predicción y bloques crecientes

Tiempo de entrenamiento para 2 dedos

En la tabla 4.6 se muestra el tiempo requerido para el entrenamiento de dos dedos con 20 árboles de regresión y bloques de imágenes del mismo tamaño en cada para cada etapa correspondiente a un nodo de división. Los entrenamientos con árboles de regresión para 2 dedos se realizaron utilizando un conjunto de 18000 imágenes.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	1:22:24.2	11.60
1	13:31:08.4	11.22
2	6:20:25:17.9	10.40

Tabla 4.8: Tiempo de entrenamiento para 2 dedos sin predicción y bloques del mismo tamaño

Tiempo de entrenamiento para mano completa

La tabla 4.9 incluye el tiempo necesario para el entrenamiento de la mano completa con 5 árboles de regresión y bloques de imágenes del mismo tamaño para cada etapa correspondiente a un nodo de división. La profundidad del entrenamiento aumenta considerablemente el tiempo necesario para el entrenamiento. Por lo tanto, la cantidad de árboles entrenados en esta configuración es de solamente 5 y para profundidades de 0 y 1 niveles de nodos de partición. Además el conjunto de datos utilizado para evaluar la mano completa es un subconjunto de 4500 imágenes de mano completa.

Profundidad n	Tiempo entrenamiento DD:HH:MM:SS	Error articulación (píxeles)
0	29:59.3	17.10
1	01:21:15:13.9	16.28

Tabla 4.9: Tiempo de entrenamiento para mano completa sin predicción y bloques del mismo tamaño

Los resultados muestran que el error de predicción aumenta al entrenar árboles con mayor cantidad de nodos, siendo de 7.43 píxeles en el mejor de los casos para 1 dedo y llegando hasta 16.28 para el caso de la mano completa. Este número puede ser mejorado realizando un entrenamiento con mayor cantidad de nodos de partición en el árbol así como aumentando la cantidad de características aleatorias a evaluar en cada uno de los nodos. Debido al tiempo necesario para entrenar este árbol de regresión solamente se realizó en este caso el entrenamiento con profundidades de nodos de partición 0 y 1; evaluar una mayor profundidad requiere cambiar la implemenatción por una más paralelizable utilizando GPUs.

4.3.6 Tiempo necesario para la predicción

La evaluación del tiempo necesario para realizar la etapa de predicción ha sido realizada utilizando el mismo hardware descrito previamente con 16 árboles de regresión generados durante el proceso de entrenamiento para igualar la cantidad de núcleos físicos disponibles. Esto porque durante la predicción se evalúa cada árbol en paralelo.

El proceso de predicción medido se estima para este algoritmo en cuadros por segundos, que corresponde a la cantidad de predicciones que el algoritmo es capar de realizar en ese tiempo en un conjunto imágenes. Durante una predicción se calcula la posición de todas las articulaciones para una determinada imagen del conjunto de prueba.

Los resultados mostrados en la tabla 4.10 listan los resultados de los cuadros por segundo que el sistema es capaz de procesar para 16 árboles con 0 nodos de partición entre etapas de entrenamiento para los casos de 1 dedo, 2 dedos y mano completa respectivamente. Los valores mostrados en la tabla 4.11 corresponden a hacer esta misma evaluación utilizando árboles con un nivel de nodos de partición en cada etapa del árbol de regresión, aunque para mano completa se ha dejado pendiente debido al tiempo necesario para el entrenamiento de dicho árbol.

Configuración	Cuadros por segundo
Un dedo	110.95
Dos dedos	69.94
Mano completa	36.01

Tabla 4.10: Predicciones por segundo para el algoritmo actual. Caso básico con profundidad de 0 nodos de partición

Configuración	Cuadros por segundo
Un dedo	95.84
Dos dedos	58.79
Mano completa	–

Tabla 4.11: Predicciones por segundo para el algoritmo actual. Caso con profundidad de 1 nodos de partición

El método utilizado en este trabajo permite realizar la predicción de las articulaciones de la mano con mayor velocidad que otros algoritmos de la literatura basados en árboles de decisión, aún cuando la implementación obtenida tiene un tiempo de ejecución considerablemente mayor al mencionado en la implementación publicada, en la cual se reporta 62 cuadros por segundo en la predicción. Trabajos presentados anteriormente que utilizan este mismo tipo de árboles hace predicción con 12 cuadros por segundo [32], el algoritmo presentado en [14] tiene una velocidad también cercana a los 10 frames por segundo, tal

como se evalúa también en [27]. Algoritmos más recientes [16] tienen un rendimiento ligeramente menor (55 cuadros por segundo) al algoritmo en el cual se basa este trabajo (62 cuadros por segundo), sin embargo, en este se parte también de un algoritmo de mejora a los árboles de regresión tradicionales para la mejora de su rendimiento.

Otros algoritmos basados en redes neuronales han alcanzado mayores velocidades recientemente, 215 cuadros por segundo [11], 166 [20], 30 [7] y [21], pero todos estos hacen uso conjunto de CPU y GPU para entrenamiento y predicción. En el caso de algoritmos generativos, el rendimiento de estos es de 30 cuadros por segundo [22] también haciendo uso de CPU y GPU, aunque en general, los algoritmos generativos tienen un costo computacional mayor que los algoritmos discriminativos, tanto en entrenamiento como en el proceso de predicción [29].

Aunque es esperable que al aumentar la cantidad de nodos de partición aumente el tiempo requerido de predicción porque aumenta la profundidad del árbol, los resultados en esta sección muestran que el algoritmo es capaz de realizar las predicciones en tiempo real para los casos considerados utilizando solamente computación mediante CPU, y por esta razón el algoritmo es atractivo por su menor complejidad computacional

Capítulo 5

Conclusiones y recomendaciones

El algoritmo parcialmente evaluado ha permitido analizar los diferentes algoritmos necesarios para la solución y los comportamientos para cada uno de estos. La evaluación del árbol de nodos latentes utilizando múltiples imágenes efectivamente permite obtener un grafo conectado a partir de un conjunto de entrenamiento en el cual se eliminan las conexiones no deseadas en las que hay discontinuidades en la mano.

Durante el entrenamiento de los árboles de regresión se ha mostrado efectivamente que a pesar de la cantidad de datos que pudo ser usada y la cantidad de características evaluadas por el sistema no son suficientes para obtener la respuesta esperada del sistema. El error por articulación mostrado para cada uno de los dedos en el caso del entrenamiento con un dedo es muy similar a el error en el algoritmo original [28], de 8 píxeles; sin embargo cuando se utilza un LTM de mayor tamaño con mayor cantidad de nodos latentes, el error es mayor al error esperado.

Los problemas que se han presentado para la evaluación del algoritmo están relacionados al tiempo de ejecución, que no ha permitido hacer una completa evaluación tal como se propone originalmente, aún cuando se hace uso de procesamiento paralelo en CPU. La mejora siguiente para el algoritmo sería utilizar mayor paralelización con una implementación distribuida, que permita la ejecución del algoritmo en múltiples sistemas con CPU y GPU para reducir el tiempo requerido para entrenamiento.

El entrenamiento del algoritmo utilizando el método de propagación de imágenes desde la raíz debe reconsiderarse en caso de contar con suficientes imágenes para el proceso de entrenamiento, esto para no utilizar imágenes en nodos internos en los que esa imagen no hubiera llegado de otra forma, aunque mientras no sea posible utilizar dicha cantidad de imágenes, el enfoque de utilizar nuevos datos en cada etapa presenta resultados similares. Esto de comprobarse con una mayor capacidad computacional que el algoritmo obtiene mejores resultados con mayor profundidad de los árboles utilizados.

Bibliografía

[1] Achintya K. Bhowmik, Selim BenHimane, Gershom Kutliroff, David Molyneaux, Blake C. Lucas, Chaim Rand, and Hon Pong Ho. I1.3: Invited paper: Immersive applications based on depth-imaging and 3d-sensing technology. *SID Symposium Digest of Technical Papers*, 46(1):83–86, 2015. URL http://dx.doi.org/10.1002/sdtp.10280.

[2] L Breiman. Random forests. *Machine learning*, pages 5–32, 2001. URL http://link.springer.com/article/10.1023/A:1010933404324.

[3] Myung Jin Choi, Vincent Y. F. Tan, Animashree Anandkumar, and Alan S. Willsky. Learning latent tree graphical models. *J. Mach. Learn. Res.*, 12:1771–1812, July 2011.

[4] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, May 1968.

[5] Salvador Cobos, Manuel Ferre, and Rafael Aracil. Simplified Human Hand Models for Manipulation Tasks. *Edge Robotics 2010*, 2010.

[6] Antonio Criminisi. Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2-3):81–227, 2011.

[7] Xiaoming Deng, Shuo Yang, Yinda Zhang, Ping Tan, Liang Chang, and Hongan Wang. Hand3d: Hand pose estimation using 3d neural network. *CoRR*, abs/1704.02224, 2017. URL http://arxiv.org/abs/1704.02224.

[8] L. Dipietro, A.M. Sabatini, and P. Dario. A survey of glove-based systems and their applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(4):461–482, July 2008.

[9] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, and X. Twombly. A Review on Vision-Based Full DOF Hand Motion Estimation. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, 89557(1), 2005.

[10] Edison Fernández. Detección de manos en imágenes de profundidad mediante el uso de bosques de decisión aleatorios. Master's thesis, Tecnológico de Costa Rica, Cartago, Costa Rica, 2015.

[11] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Trans. Cybernetics*, 43(5), October 2013.

[13] M. Hansard, S. Lee, O. Choi, and R.P. Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. SpringerBriefs in Computer Science. Springer, 2012.

[14] Cem Keskin, Furkan K$#305;ra$#231;, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 852–863, Berlin, Heidelberg, 2012. Springer-Verlag. URL http://dx.doi.org/10.1007/978-3-642-33783-3_61.

[15] Cem Keskin, Furkan Kirac, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1228–1234, 2011.

[16] Peiyi Li and Xi Li. 3D Hand Pose Estimation Using Randomized Decision Forest with Segmentation Index Points. *Iccv*, pages 819–827, 2015.

[17] Marco Madrigal. Reconocimiento de pose estática de manos en imágenes de profundidad. Master's thesis, Tecnológico de Costa Rica, Cartago, Costa Rica, 2015.

[18] Melissa Montero. Estimación de pose de manos con un modelo antropomórfico en imágenes de profundidad. Master's thesis, Tecnológico de Costa Rica, Cartago, Costa Rica, 2015.

[19] Raphael Mourad, Christine Sinoquet, Nevin L. Zhang, Tengfei Liu, and Philippe Leray. A survey on latent tree models and applications. *Journal of Artificial Intelligence Research*, 47:157–203, 2013.

[20] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. *CoRR*, abs/1704.02201, 2017. URL http://arxiv.org/abs/1704.02201.

[21] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. *CoRR*, abs/1708.08325, 2017. URL http://arxiv.org/abs/1708.08325.

[22] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. CHI, April 2015.

[23] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, and Alex Kipman. Efficient human pose estimation from single depth images. *Decision Forests for Computer Vision and Medical Image Analysis*, pages 175–192, 2013. URL http://link.springer.com/chapter/10.1007/978-1-4471-4929-3{_}13{%}5Cnpapers3://publication/uuid/B0D3C6EC-3C89-4BF8-B1E4-CC0A8DD03567.

[24] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, and Andrew Blake. Efficient human pose estimation from single depth images. *Trans. PAMI*, 2012.

[25] Edgar Simó Serra. *Kinematic Model of the Hand using Computer Vision*. PhD thesis, Universitat Politécnica de Catalunya, 2011.

[26] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded Hand Pose Regression. pages 824–832, 2015.

[27] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: Data, methods, and challenges. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1868–1876, Dec 2015.

[28] Danhang Tang and Tae-kyun Kim. Latent Regression Forest : Structured Estimation of 3D Articulated Hand Posture. pages 3786–3793, 2014.

[29] I. Ulusoy and C.M. Bishop. Generative versus discriminative methods for object recognition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 258–265 vol. 2, June 2005.

[30] F.P.J. van der Hulst, S. Schatzle, C. Preusche, and A. Schiele. A functional anatomy based kinematic human hand model with simple size adaptation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 5123–5129, May 2012.

[31] Chengde Wan, Angela Yao, and Luc Van Gool. Hand pose estimation from local surface normals. 9907:554–569, 10 2016.

[32] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3456–3462, Dec 2013.

[33] Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 33–42, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

Índice alfabético